

Arquitectura e Ingeniería de Computadores

Examen final (26/06/2007). Teoría.

Nota.- Cada cuatrimestre se califica de 0 a 10 puntos. La nota final de la asignatura se obtiene tal como se indica en la ficha docente de la asignatura.

Preguntas correspondientes al Primer Cuatrimestre

1. a) Explica brevemente cómo funciona el hardware de la “unidad de cortocircuito” que se introduce en la etapa EX del DLX básico para detectar y resolver las dependencias LDE entre instrucciones aritméticas. (1,25 puntos)

b) Completa la figura de la hoja adjunta con todas las conexiones necesarias para implementar el cortocircuito. Numera las conexiones que has dibujado y explica el significado de cada una de ellas. (1, 25 puntos)

2. La figura representa la tasa de fallos de una cache al ejecutar una misma carga de trabajo sobre un mismo procesador, pero variando el tamaño de la cache y el grado de asociatividad. El tamaño de bloque se mantiene constante y la política de reemplazamiento es LRU. Responde razonadamente a las siguientes preguntas:

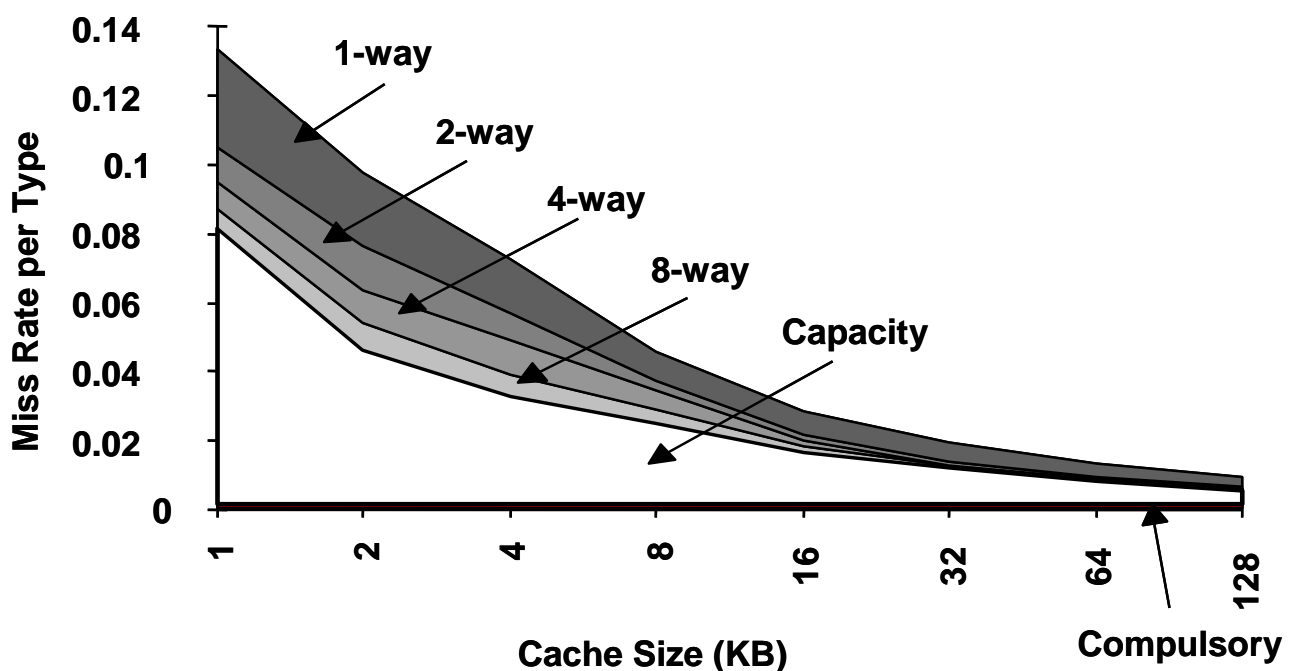
a) Explica qué son los fallos iniciales (compulsory). (0,5 puntos)

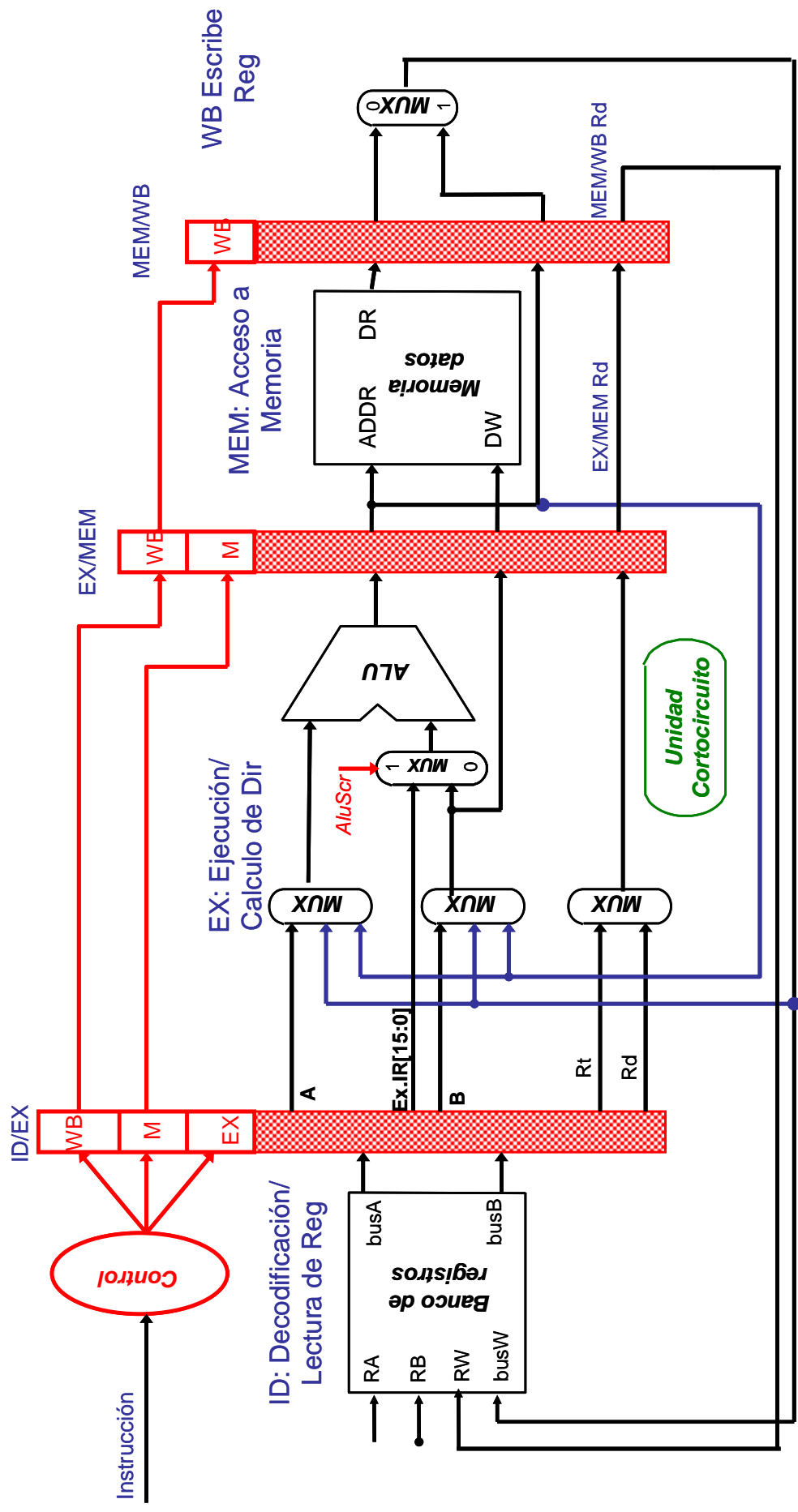
b) Explica por qué motivo la tasa de fallos iniciales se mantiene constante, independientemente del tamaño de la cache. (0,5 puntos)

c) Para una cache de 4 KB directa, indica el valor aproximado de la tasa de fallos de conflicto. ¿Cómo lo has obtenido? (0, 5 puntos)

d) ¿En qué consiste la regla empírica 2:1? Indica con un ejemplo cómo se cumple esta regla en la gráfica. (0,5 puntos)

e) Explica razonadamente si el tiempo de acceso a la cache mejora o empeora al aumentar el grado de asociatividad. (0,5 puntos)





Arquitectura e Ingeniería de Computadores. Examen Parcial (Problemas). 26/06/2007

1) En un DLX con segmentación ejecutamos el siguiente fragmento de código:

```
loop: ADDI r5,r6,#1
      LD F4,0(r5)
      MULF F6,F2,F4
      ADDF F8,F4,F2
      DIVF F2,F8,F6
      LD F0,0(r2)
      MULF F4,F0,F4
      SD 0(r5),F4
      DIVF F4,F2,F6
      ADDI r5,r5,#1
      ADDF F4,F0,F8
      SUBI r6,r6,#1
      BNEZ r6,loop
      ADDI r2,r2,#1
```

Dibujar el diagrama instrucción-tiempo e indicar los cortocircuitos realizados. Indicar claramente las paradas y sus causas. Calcular los CPI (ciclos por instrucción) en el régimen estacionario. Se supone que:

- * Un dato se puede escribir en un registro y leer su valor en el mismo ciclo.
- * Se dispone de lógica de cortocircuito (forwarding).
- * Los saltos se resuelven en la etapa de decodificación.
- * La detección de todo tipo de riesgos y generación de paradas se realiza en la etapa de decodificación.
- * En el caso de los riesgos WAW se produce una parada hasta que la instrucción ya lanzada entre en la etapa de memoria.
- * El registro r6 contiene inicialmente el valor "1000".
- * Existe predicción estática de salto no tomado.
- * Dos instrucciones no pueden acceder simultáneamente a la etapa de acceso a memoria ni tampoco a la de escritura en el banco de registros.
- * Se dispone de las siguientes unidades funcionales:

UF	Cantidad	Latencia	Segmentación
FP ADD	1	2	No
FP MUL	1	4	No
FP DIV	1	5	No
Int ALU	1	1	No

PUNTACIÓN: 3 PUNTOS

2) Disponemos de un procesador dotado de planificación dinámica de instrucciones según el método de Tomasulo. Se dispone de las unidades funcionales que se muestran en la tabla siguiente:

UF	Cantidad	Latencia	Segmentación
FP ADD/SUB	2	3/3	No
FP MUL/DIV	2	5/8	No

Suponer que en un mismo ciclo no se puede escribir un dato en el bus común e iniciar la ejecución de una operación que estuviera en espera de ese dato, y que existe una estación de reserva por cada unidad funcional.

Para la secuencia de instrucciones de la tabla siguiente, indica en qué ciclo (o ciclos) de reloj se realiza cada una de sus fases. Indica qué tipos de riesgos se producen.

Instrucción	Issue Emisión	Ejecución	Escribe resultado
ADDD F2,F2,F4			
SUBD F0,F4,F2			
MULTD F4,F2,F0			
ADDD F6, F0,F0			
DIVD F4,F4,F6			
MULTD F2,F6,F6			
SUBD F0,F2,F6			
MULTD F2,F2,F0			

PUNTACIÓN: 2 PUNTOS

SOLUCIÓN

1)

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29		
ADDI r5,r6,#1	IF	ID	EX	M	WB																										
LD F4,0(r5)		IF	ID	EX	M	WB																									
MULD F6,F2,F4			IF	ID	ID	M1	M2	M3	M4	M	WB																				
ADDD F8,F4,F2				IF	ID	A1	A2	M	WB																						
DIVD F2,F8,F6					IF	ID	ID	ID	D1	D2	D3	D4	D5	M	WB																
LD F0,0(r2)						IF	IF	IF	ID	EX	M	WB																			
MULD F4,F0,F4									IF	ID	ID	M1	M2	M3	M4	M	WB														
SD 0(r5),F4										IF	IF	ID	ID	ID	ID	EX	M	WB													
DIVD F4,F2,F6												IF	IF	IF	IF	IF	ID	D1	D2	D3	D4	D5	M	WB							
ADDI r5,r5,#1																	IF	ID	EX	M	WB										
ADDD F4,F0,F8																		IF	ID	ID	ID	ID	A1	A2	M	WB					
SUBI r6,r6,#1																			IF	IF	IF	IF	ID	ID	EX	M	WB				
BEQZ r6,loop																						IF	IF	ID							
ADDI r2,r2,#1																								IF							
ADDI r5,r6,#1																										IF	ID				

XX: Parada por riesgo LDE

XX: Parada por riesgo estructural (siguiente etapa ocupada)

XX: Riesgo EDE

XX: Riesgo estructural: dos instrucciones intentan acceder simultáneamente a memoria

CPI = 25/14 = 1.79

** : Asumiendo forwarding combinacional, de no existir un ciclo más de penalización

SOLUCIÓN

2)

Instrucción	Issue Emisión	Ejecución	Escribe resultado
ADDD F2,F2,F4	1	2-4	5
SUBD F0,F4,F2	2	6-8	9
MULTD F4,F2,F0	3	10-14	15
ADDD F6, F0,F0	6	10-12	13
DIVD F4,F4,F6	7	16-23	24
MULTD F2,F6,F6	16	17-21	22
SUBD F0,F2,F6	17	23-25	26
MULTD F2,F2,F0	23	27-31	32