



Documento anónimo

borrador_soluciones_teoria_junio2016.pdf

Exámenes Resueltos (teoría y Prácticas)



2º Arquitectura de Computadores



Grado en Ingeniería Informática



Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación
UGR - Universidad de Granada



CUNEF POSTGRADO

La formación que necesitas para tu **futuro profesional**

⇒ **FINANZAS**
⇒ **DERECHO**

⇒ **DATA**
SCIENCE

www.cunef.edu

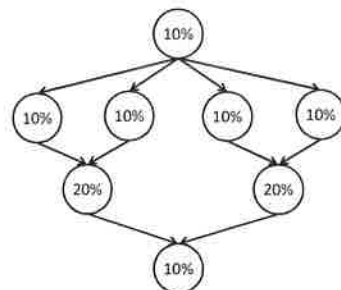
Grado en Ingeniería Informática y Grado en Ing. Informática y Matemáticas

Arquitectura de Computadores (AC)

Examen de 13 de Junio de 2016

Estudiante:

Ejercicio 1. (1 punto) En la Figura se muestra el grafo de dependencias entre tareas para una aplicación. En cada nodo del grafo la fracción del tiempo de ejecución secuencial que tarda cada tarea. El tiempo de ejecución secuencial de la aplicación es de 200 s. y las tareas no se pueden dividir en tareas de menor granularidad. Si dispone de un multiprocesador UMA con cuatro procesadores, conectados a través de un bus compartido, y la comunicación entre tareas, a través de la memoria compartida, tarda 10 segundos (si las tareas que se comunican están asignadas a procesadores



diferentes): (a) ¿Qué ganancia de velocidad obtiene si utiliza los cuatro procesadores del computador? ¿y si utiliza dos procesadores? (b) ¿Qué eficiencia obtiene si utiliza los cuatro procesadores del computador? ¿y si utiliza dos procesadores?

Ejercicio 2. (1 punto) Se dispone de un multiprocesador CC-NUMA con 4 procesadores o nodos (N1-N4) y una memoria de 32 GBytes. El multiprocesador implementa para mantener la coherencia de cache el protocolo MSI basado en directorios distribuidos sin difusión. Cada procesador dispone de una cache de datos de último nivel de 4MBytes con marcos de bloque (también llamados líneas) de 32 bytes. En el multiprocesador se están ejecutando en paralelo dos threads que acceden a los elementos de dos vectores X[] e Y[] de 8 elementos de 64 bits cada uno. Los vectores se encuentran almacenados a partir de una dirección de memoria múltiplo de 32: primero están almacenados los componentes de X[], justo a continuación, los elementos de Y[]. Conteste a las siguientes preguntas:

- ¿Qué estados puede tener un bloque en cache y en memoria? Defina los estados.
- ¿Cuál es el tamaño del subdirectorio de un nodo?
- ¿Cuántos bloques de memoria ocupan los vectores X[] e Y[]?
- Suponiendo que inicialmente los bloques que contienen ambos vectores no están en ninguna cache, ¿Cuál será el contenido de las entradas del directorio para cada uno de estos bloques?
- Indique los estados de los bloques en las caches y los cambios en los contenidos del directorio ante la siguiente secuencia de eventos (considere que inicialmente los bloques que contienen ambos vectores no están en ninguna cache): 1) Lectura generada por el procesador 1 a X[0]. 2) Escritura generada por el procesador 2 a X[1]. 3) Lectura generada por el procesador 1 a X[2]. 4) Escritura generada por el procesador 2 a Y[2]

NOTA: Suponga que bloques distintos se almacenan en la cache de cada procesador en marcos de bloque (líneas) diferentes.

Ejercicio 3. (1 punto) En un multiprocesador con un modelo de consistencia que garantiza los órdenes $W \rightarrow W$, $R \rightarrow W$, $R \rightarrow R$, se ha ejecutado el siguiente código (X e Y son variables (memoria), reg1 y reg2 son registros del núcleo, la variable Y está a 0 en memoria):

Accesos a memoria	Núcleo P1	Accesos a memoria	Núcleo P2
(1.1)W(X)	X=1;	(2.1)W(Y)	Y=2
(1.2)R(X)	reg1=X;	(2.2)W(X)	X=2
(1.3)R(Y)	reg2=Y;		

Contestar a las siguientes cuestiones:

- 0.8 (a) ¿Cuáles son los órdenes globales de los accesos a memoria que podrían ver P1 y P2 teniendo en cuenta que garantiza W->W, R->W, R->R? Escribir todas las combinaciones posibles. Indicar qué se obtiene en reg1 y reg2 para cada uno de los órdenes indicados y cuál sería el contenido de X e Y para cada uno de ellos.
- 0.2 (b) ¿Qué recursos incluyen los núcleos de procesamiento para relajar el orden W->R?

Ejercicio 4. (1 punto) La siguiente secuencia de instrucciones

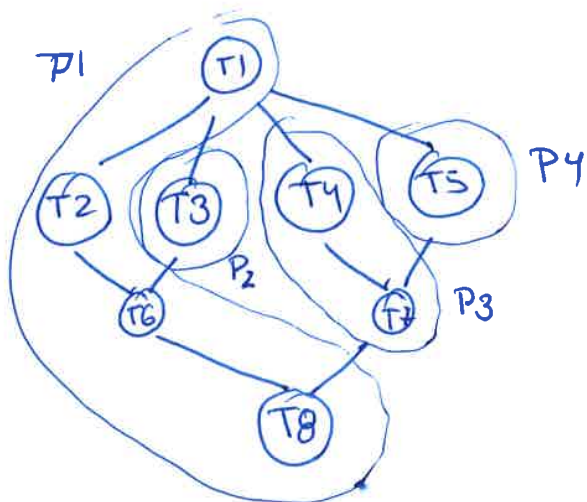
[illegible]

se ha ejecutado en un procesador superescalar capaz de captar cuatro instrucciones por ciclo y de decodificar tres instrucciones por ciclo. El procesador dispone de una estación de reserva común para todas las unidades funcionales con emisión desordenada de hasta tres instrucciones por ciclo, y de un ROB (Buffer de reorden) del que se pueden retirar tres instrucciones por ciclo.

- 0.8 (a) Indique la evolución temporal de las instrucciones a través de las etapas del procesador
- 0.1 (b) ¿Cuál es el valor promedio de CPI (ciclos por instrucción) en este código?
- 0.1 (c) ¿Para qué se usa ROB en un núcleo de procesamiento? ¿En qué etapas del cauce segmentado de un núcleo se accede al ROB?

NOTA: Suponga que el procesador tiene una unidad de carga de memoria, una unidad de almacenamiento, dos unidades de suma/resta y una de multiplicación. La multiplicación consume 3 ciclos, la suma/resta 1 ciclo, la carga de memoria 2 ciclos, y el almacenamiento 1 ciclo.

1.

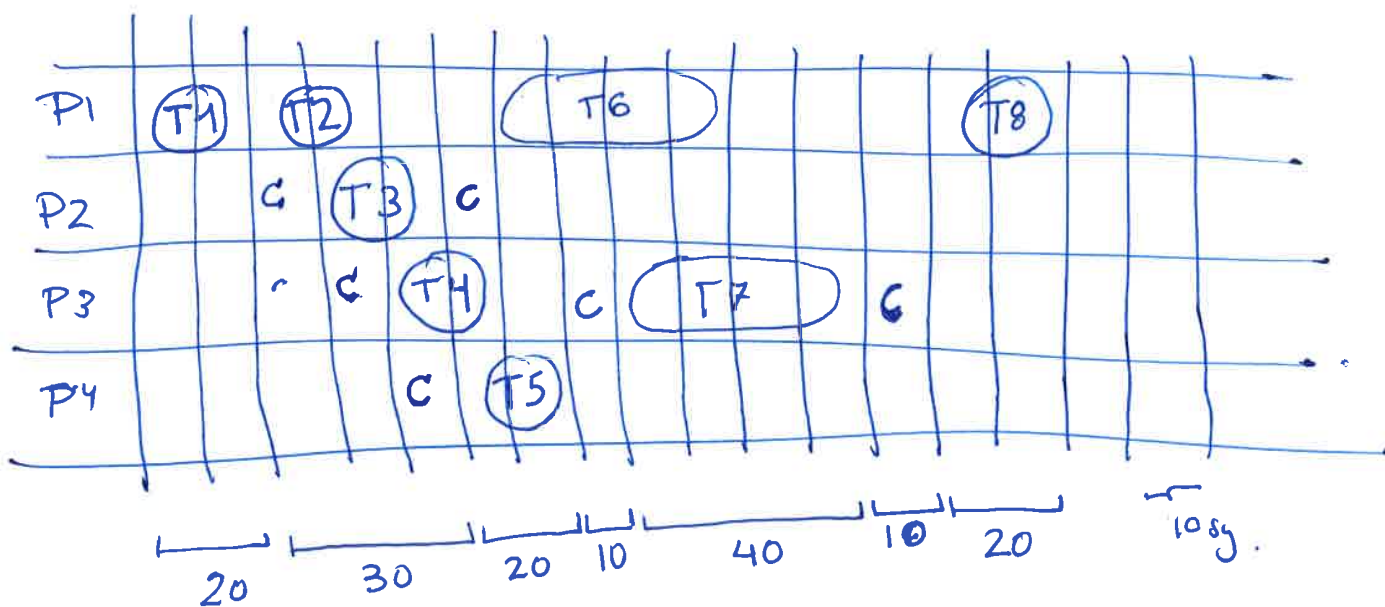


10% \rightarrow 20 sg.

20% \rightarrow 40 sg

T1-T5, T8 \Rightarrow 20 sg.

T6-T7 = 40 sg

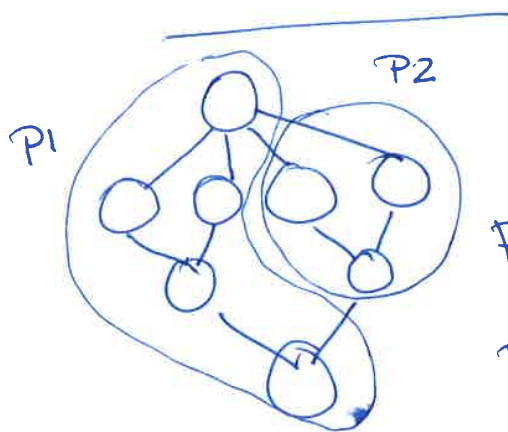


También válido:

$$20 + 20 + 40 + 20 + (30) + (20) + (10) = 160$$

$$S = \frac{200}{150} = \frac{20}{15}$$

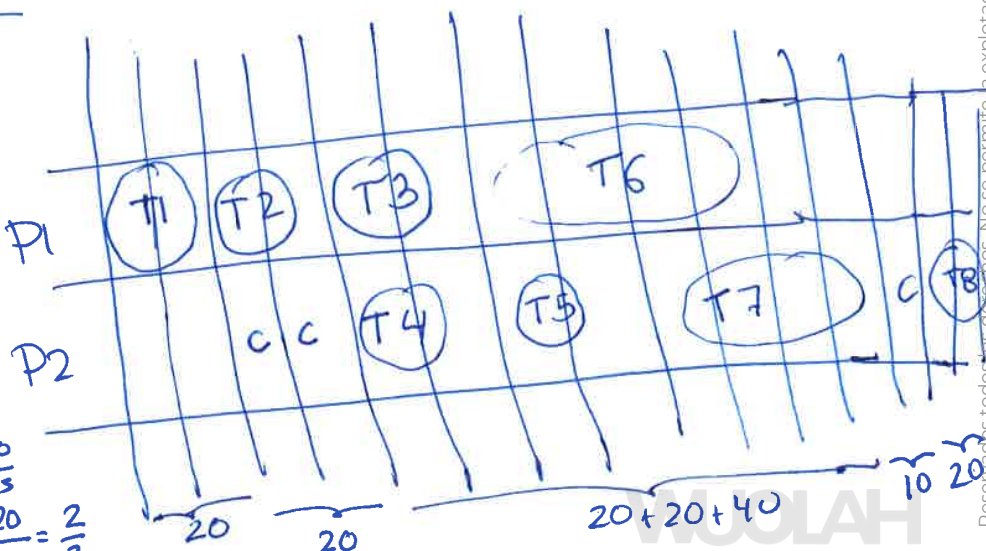
$$E = \frac{20}{15 \cdot 4} = \frac{20}{60} = \frac{1}{3}$$



T2 = 150 sg

$$S = \frac{200}{150} = \frac{20}{15}$$

$$E = \frac{20}{15 \cdot 2} = \frac{20}{30} = \frac{2}{3}$$



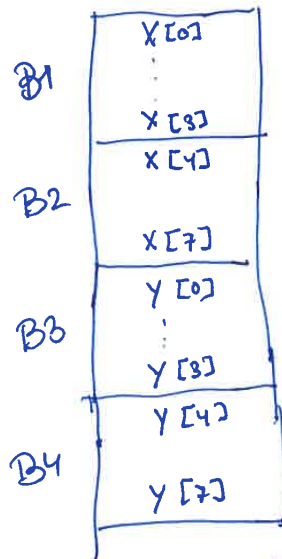
¡Define tu sueño y alcánzalo!

2-

4 nodos

N1 - N4

$$\frac{32 \text{ GBytes}}{4} = 8 \text{ GBytes.}$$



$$\left. \begin{array}{l} 32 \text{ Bytes/linea} \\ 8 \text{ Bytes/dato} \end{array} \right\} \rightarrow 4 \text{ datos/linea}$$

a) M S I
V / I

b) 4+1 bits/entrada.

$$\frac{8 \times 2^{30}}{2^5} = 8 \times 2^{25} = 2^{28} \text{ entradas}$$

$$5 \times 2^{28} \text{ bits}$$

c) 2 cada uno.

$$C_j(B_i) = I$$

0	0	0	0	0	V
0	0	0	0	0	V
0	0	0	0	0	V
0	0	0	0	0	V

d) $R_1(B1)$

$W_2(B1)$

$R_1(B1)$

$W_2(B3)$



$C3(B3) = M$

0	0	0	0	0	V
0	0	0	0	0	V
0	0	0	0	0	V
0	0	0	0	0	V

$C1(B1) = S$

1	0	0	0	0	V
0	0	0	0	0	V
0	0	0	0	0	V
0	0	0	0	0	V

$C1(B1) = I$

$C2(B1) = M$

0	1	0	0	0	V
0	0	0	0	0	V
0	0	0	0	0	V
0	0	0	0	0	V

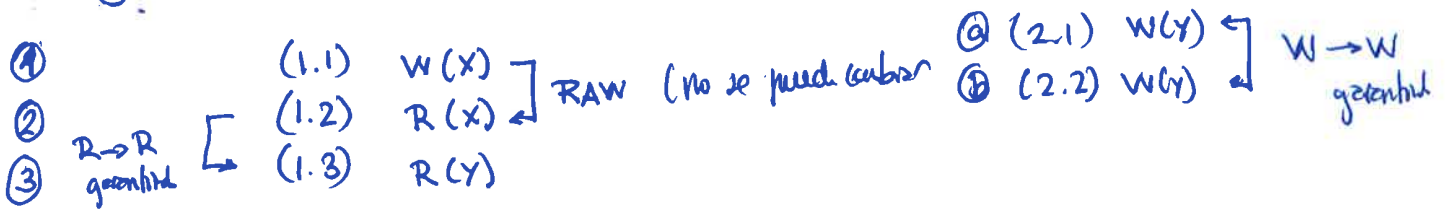
$C1(B1) = S$

$C2(B1) = S$

1	1	0	0	0	V
0	0	0	0	0	V
0	0	0	0	0	V
0	0	0	0	0	V

* Inicialmente, se cachea todo
* del cambio de que se indica.

3.-



No se producen cambios entre ellos.

② ① ③	R1 = 1 R2 = 2
② ① ③	R1 = 2 R2 = 2
② ① ③	R1 = 1 R2 = 2
② ① ③ ⑥	R1 = 1 R2 = 2
① ② ③	R1 = 2 R2 = 2
① ② ③	R1 = 1 R2 = 2
① ② ③ ⑥	R1 = 1 R2 = 2
① ② ③	R1 = 1 R2 = 2
① ② ③ ⑥	R1 = 1 R2 = 2
① ② ③ ⑥	R1 = 1 R2 = 2
① ② ③ ⑥	R1 = 1 R2 = 0

⑥ → Instrucciones específicas en el repertorio máquina que hacen que no se puedan producir adelantos en los accesos a memoria (todas las lecturas o escrituras anteriores se deben completar antes y las instrucciones se completan antes de las escrituras/lecturas posteriores).

$$T = T_L I + (n-1) C R$$

$12 = 6 + 6 \text{ CPS} \rightarrow$

$$\frac{12-6}{6} = CPI = \underline{\underline{1}}$$

Ejercicio 4. (1 punto) La siguiente secuencia de instrucciones

Instrucción	Significado	1	2	3	4	5	6	7	8	9	10	11	12
lw r1,0(r2)	r1 ← m(r2)	IF	ID	EX	EX	ROB	WB						
add r5,r4,r1	r5 ← r4+r1	IF	ID			EX	ROB	WB					
lw r4,0(r3)	r4 ← m(r3)	IF	ID			EX	EX	ROB	WB				
mult r6,r5,r4	r6 ← r5×r4	IF		ID				EX	EX	EX	ROB	WB	
add r4,r5,r1	r4 ← r5+r1		IF	ID			EX	ROB				WB	
sw 0(r3),r4	m(r3) ← r4		IF	ID				EX	ROB			WB	
sw 0(r2),r6	m(r2) ← r6		IF		ID						EX	ROB	WB

12
7

se ha ejecutado en un procesador superescalar capaz de captar cuatro instrucciones por ciclo y de decodificar tres instrucciones por ciclo. El procesador dispone de una estación de reserva común para todas las unidades funcionales con emisión desordenada de hasta tres instrucciones por ciclo, y de un ROB (Buffer de reorden) del que se pueden retirar tres instrucciones por ciclo.

(a) Indique la evolución temporal de las instrucciones a través de las etapas del procesador

(b) ¿Cuál es el valor promedio de CPI (ciclos por instrucción) en este código?

(c) ¿Para qué se usa ROB en un núcleo de procesamiento? ¿En qué etapas del cauce segmentado de un núcleo se accede al ROB? *Finalizar orden de y reemplazando // Se utiliza tras la decodificación, la ejecución y en la etapa de finalización*

NOTA: Suponga que el procesador tiene una unidad de carga de memoria, una unidad de almacenamiento, dos unidades de suma/resta y una de multiplicación. La multiplicación consume 3 ciclos, la suma/resta 1 ciclo, la carga de memoria 2 ciclos, y el almacenamiento 1 ciclo.

Ejercicio 5. (1 punto) En un procesador con repertorio LOAD/STORE todas las instrucciones pueden predicarse. Para establecer los valores de los predicados se utilizan instrucciones de comparación con el formato **p1,p2 cmp.cnd ra,rb**, donde **cnd** es la condición que se comprueba entre los registros **ra** y **rb** (**lt, ge, eq, ne,...**). Si la condición es verdadera **p1=1** y **p2=0**, y si es falsa, **p1=0** y **p2=1**. Una instrucción precedida por un predicado, (**p**) **instruc**, sólo se ejecuta si el predicado que la precede es igual a 1, **p=1** (su valor habrá sido establecido por otra instrucción de comparación). (a) Escriba sin ninguna instrucción de salto el siguiente código, donde los arrays **X[i]** y **Y[j]** (**j=1,2**) están constituidos por elementos de 32 bits en posiciones consecutivas a partir de las direcciones contenidas en los registros **r1** (para la dirección donde empieza **X[j]**) y **r2** (para la dirección donde empieza **Y[j]**) :

```
for (i=1; 2; ++i) {
```

```
if (X[i]>Y[i]) then { X[i]=X[i]-Y[i];} else {if (Y[i]>X[i]) then X[i]=Y[i]-X[i]; else X[i]=2*X[i]}
```

}

(b) Con los predicados se eliminan los saltos ¿por qué resulta de interés eliminar los saltos? ¿Qué otro hardware puede incluir un núcleo de procesamiento relacionado con el procesamiento de los saltos?

Ejercicio 6. (1 punto) En un multiprocesador con un modelo de consistencia que garantiza los órdenes W->W, R->W, R->R, se ha ejecutado el siguiente código (X e Y son variables (memoria), reg1 y reg2 son registros del núcleo, la variable Y está a 0 en memoria):

Accesos a memoria	Núcleo P1	Accesos a memoria	Núcleo P2
(1.1)W(X)	X=1;	(2.1)W(Y)	Y=2
(1.2)R(X)	reg1=X;	(2.2)W(X)	X=2
(1.3)R(Y)	reg2=Y;		

Contestar a las siguientes cuestiones:

(a) ¿Cuáles son los órdenes globales de los accesos a memoria que podrían ver P1 y P2 teniendo en cuenta que garantiza W->W, R->W, R->R? Escribir todas las combinaciones posibles. Indicar qué se obtiene en reg1 y reg2 para cada uno de los órdenes indicados y cuál sería el contenido de X e Y para cada uno de ellos.

(b) ¿Qué recursos incluyen los núcleos de procesamiento para relajar el orden W→R?