

# ACExamen20140624.pdf



**PruebaAlien**



**Arquitectura de Computadores**



**2º Grado en Ingeniería Informática**



**Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación**  
**Universidad de Granada**

A black and white photograph of a young woman with long dark hair, looking upwards and to the right with a hopeful expression.

**WUOLAH**  
**Alicia será astronauta.**

**#NoTeApuntesAWuolah**



ugr Universidad  
de Granada

ETSIIT  
Escuela Técnica Superior  
de Ingenierías Informáticas  
y de Telecomunicación



2º curso / 2º cuatr.

Grupos A,B,C,D del  
Grado Ing. Inform.

Doble Grado Ing.  
Inform. y Mate.

## Arquitectura de Computadores (AC)

Examen de teoría. 24 de junio de 2014

Puntuación: 4 puntos

Duración: 2 horas

Identificación: DNI

**Cuestión 1.**(0.5 puntos) **(a)** Explique cuáles son las diferencias del modelo de consistencia débil y el modelo de consistencia de liberación. **(b)** Indique en qué son exactamente iguales ambos modelos.

**Cuestión 2.**(0.5 puntos) Diferencias entre un core VLIW y un core superescalar.

**Ejercicio 1.** (0.75 puntos) Una empresa dedicada al desarrollo de microprocesadores ha recibido dos propuestas de proyectos viables de sus equipos de ingenieros, pero sólo tiene presupuesto para invertir en uno de ellos:

- El primero tiene un coste de un millón de euros y se basa en la mejora de la microarquitectura para reducir los CPI de las instrucciones a la mitad.
- El segundo tiene un coste de seiscientos mil euros y consiste en la aplicación de nuevas técnicas de optimización de código para conseguir que el compilador que genera código ejecutable para su familia de microprocesadores reduzca en los programas un 25% del número de instrucciones de tipo LOAD, un 25% de las de tipo STORE y otro 25% las BRANCH.

¿Qué proyecto debería seleccionar? ¿Por qué? Razone la respuesta apoyándola con los cálculos que estime oportunos.

**Nota:**La familia de procesadores de esta empresa tiene una arquitectura de tipo LOAD/STORE en las que las operaciones sólo utilizan como operandos registros de la CPU. Para un conjunto de programas representativos de su actividad se tiene que el 30% de las instrucciones son operaciones con números enteros (2 CPI), el 20% son operaciones con números en coma flotante (8 CPI), el 20% LOADs (4 CPI), el 10% STOREs (4 CPI) y el 20% BRANCHs (4 CPI).

**Ejercicio 2.** (0.75 puntos) Se quiere encontrar la mejor configuración de diseño para la pala de un aerogenerador. Hay un total de 240 configuraciones posibles. La simulación de cada una de las configuraciones implica realizar  $10^{14}$  operaciones en coma flotante (todas las posibles configuraciones requieren el mismo número).

- Calcule una estimación del tiempo que se necesitaría para simular todas las configuraciones posibles en un procesador superescalar a 2 GHz capaz de terminar dos operaciones en coma flotante por ciclo. Haga los cálculos teniendo en cuenta el número de operaciones como flotante indicados.
- Estime cuántos procesadores similares al del apartado (a) tendrían que trabajar en paralelo simulando configuraciones para conseguir reducir ese tiempo a unas 2 horas suponiendo que sólo trabaja un procesador (para asignar trabajos, acumular resultados, etc.) durante 20 minutos del tiempo total (independientemente del número de procesadores que trabajen), el resto del tiempo estarían trabajando todos los procesadores. Suponga que el trabajo se reparte por igual entre los procesadores disponibles.
- Estime cuál sería la máxima ganancia en prestaciones que se podría conseguir teniendo en cuenta todos los datos aportados hasta ahora y que se puede disponer de cualquier número de procesadores.

**Ejercicio 3. (0.75 puntos)** (a) En el contexto de sincronización, ¿qué es una barrera? (b) Suponga que está desarrollando una implementación openMP para un compilador de C/C++. Escriba el código que usaría para implementar la directiva `barrier`. Tenga en cuenta que esta directiva la pueden usar los threads varias veces durante la ejecución. Utilice las funciones de OpenMP que necesite de la siguiente tabla (no se puede usar directivas para implementar una directiva):

Descripción	Función de la biblioteca OpenMP
Cerrar el cerrojo, es un <code>lock(k)</code>	<code>omp_set_lock(&amp;k)</code>
Abrir el cerrojo, es un <code>unlock(k)</code>	<code>omp_unset_lock(&amp;k)</code>
Devuelve el número de threads en la región paralela	<code>omp_get_num_threads()</code>

(c) Indique qué variables deben ser privadas, cuáles deben ser compartidas, y cuáles pueden ser compartidas o privadas indistintamente. No puede debe dejar ninguna variable sin clasificar.

**Ejercicio 4. (0.75 puntos)** En un procesador todas las instrucciones pueden predicarse. Para establecer los valores de los predicados se utilizan instrucciones de comparación (`cmp`) con el formato (p) p1, p2 `cmp.cnd x,y` donde `cnd` es la condición que se comprueba entre `x` e `y` (`lt`, `ge`, `eq`, `ne`,....). Si la condición es verdadera `p1=1` y `p2=0`, y si es falsa, `p1=0` y `p2=1`. La instrucción sólo se ejecuta si el predicado `p=1` (habrá sido establecido por otra instrucción de comparación).

En estas condiciones, utilice instrucciones con predicado para escribir sin ninguna instrucción de salto el siguiente código (A, B y C son variables):

```
if (A>B) then { B=B+1;} else {if (A>C) then C=C+1; else A=A+1;}
```