

## Arquitectura e Ingeniería de Computadores

Octubre 2009

Nombre:

Grupo:

### Normas de realización:

- Las preguntas han de ser respondidas en bloques de folios independientes, incluyendo el nombre en cada uno de los folios.
- Todas las respuestas han de ser correctamente detalladas y razonadas.
- No se corregirán las respuestas a lápiz.

### Pregunta 1. (1,5 ptos)

Supongamos un procesador superescalar en que se captan 4 instrucciones por ciclo, se decodifican 3 instrucciones por ciclo, se emiten 3 instrucciones por ciclo como máximo, y se retiran hasta 3 instrucciones por ciclo. La emisión y la ejecución son desordenadas, y las instrucciones, una vez decodificadas, se introducen en un buffer de reordenamiento (ROB) que permite la finalización ordenada del procesamiento de las instrucciones.

Se pide:

a) (1 pto) Indicar las dependencias entre instrucciones y como evolucionaría el buffer de reordenamiento hasta que se hayan retirado todas las instrucciones de la secuencia:

```
addd f1, f1, f4  
multd f3, f1, f2  
addd f6, f1, f4  
subd f4, f1, f6
```

b) (0,5 ptos) Suponiendo una frecuencia de 2Ghz, ¿Cuánto tarda en procesarse la secuencia de instrucciones?

Nota: la suma y la resta consumen 1 ciclo de reloj y la multiplicación 4 ciclos. Considerad que no hay limitaciones en la capacidad de los buffers, y en el número de unidades funcionales. Se supone que f1, f2 y f4 tienen valores válidos previos. Las etapas del cauce son: IF, ID/ISS, EX, ROB, WB.

### Pregunta 2. (2 ptos)

En un procesador VLIW cuyas instrucciones pueden codificar dos operaciones (dos campos o slots en cada instrucción VLIW), todas las operaciones deben 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 operación sólo se ejecuta si el predicado p=1 (habrá sido establecido por otra instrucción de comparación). Indicar cómo se escribiría la sentencia:

```

for i=1 to 2 do
    X[i+1] = Y[i]/2;
    if (X[i] < 0) then Y[i] = X[i+1]/2;
    else if (X[i] == 2) then X[i+1] = Y[i+2];
    else if (Y[i] == 1) then X[i] = Y[0];
    else X[i] = X[i+1]/Y[i];
finFor

```

en lenguaje ensamblador, sin ninguna operación de salto y con el mínimo número de instrucciones VLIW, teniendo en cuenta que las instrucciones de comparación sólo pueden aparecer en el primer campo o slot de la instrucción VLIW (el resto de las instrucciones pueden aparecer en cualquier campo). Todas las instrucciones tienen la misma latencia (1 ciclo).

*Utilizar como modo de direccionamiento Registro base + desplazamiento (R-I)*

Ejemplo: lw r1, inm16(reg)

### **Pregunta 3** (1,5 ptos)

Se dispone de una arquitectura vectorial que incorpora una memoria distribuida en 5 bloques con 3 palabras cada bloque. El acceso que se utiliza en la memoria es acceso simultáneo o tipo S. Teniendo en cuenta que el tipo de vectores con el que más se va a trabajar con vectores de 11 elementos, ¿qué tipo de entrelazado será el más óptimo para organizar la memoria, superior o inferior?

### **Pregunta 4** (2 ptos)

Sea una imagen de tamaño 4 Megapíxeles (2048x2048) codificada como 256 niveles de gris y almacenada en una matriz (denominada imagen[2048][2048]) a la que se desea extraer el histograma y la media de los valores de sus píxeles.

Se pide:

- (0,5 ptos) Desarrollar el algoritmo secuencial en C, identificar las tareas que lo componen así como su grafo de dependencias de tareas.
- (0,5 ptos) Paralelizar el algoritmo mediante el esquema oportuno suponiendo que se dispone de un cluster Beowulf formado por  $N = 64$  máquinas, y consiguiendo la máxima eficiencia posible.
- (1 pto) Obtener el tiempo de ejecución en paralelo frente al secuencial en función de las variables que consideres necesarias para la correcta especificación de los tiempos. Las variables seleccionadas deben de tener sentido con respecto al análisis que se está realizando. Por ejemplo,  $T_i$ =tiempo inicialización;  $T_c$ =tiempo comunicación,  $T_f$ =tiempo finalización, etc...

Nota: el histograma de una imagen indica qué número de píxeles tienen un determinado valor de gris. Es decir, cuántos píxeles tienen un nivel 0 de gris, cuántos tienen un nivel 1 de gris, etc.

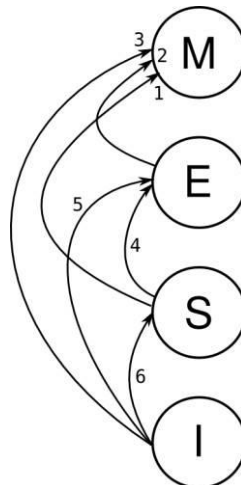
**Pregunta 5** (1,5 pts)

a) Explica cuál sería el camino para desplazarse en un hipercubo 5D desde el nodo 00001 hasta el nodo 10011 utilizando un algoritmo de encaminamiento por reducción de dimensiones. Explica **con detalle** el algoritmo que has utilizado para hacer el recorrido.

b) Explica en qué consiste el algoritmo West-First (pon también un ejemplo). ¿A qué familia de algoritmos pertenece? ¿Qué característica principal tienen estos algoritmos?

**Pregunta 6** (1,5 pts)

La figura siguiente muestra la transición de estados para un bloque en una caché con protocolo MESI. Las transiciones continuas son las debidas a acciones del procesador de la caché, y las transiciones discontinuas (que deberá indicar) a transiciones provocadas por paquetes del bus.



Se pide:

a) Defina y enumere en una tabla todas las transacciones posibles en este protocolo (con líneas continuas y discontinuas), e indique para cada una el evento que provoca la transición y el paquete que genera (formato: evento / paquete)

Ejemplos:

PrEsc/PtLecEx = Procesador Escribe y provoca un paquete de Petición de lectura exclusiva.

PtLec/RpBloque = paquete de lectura y generación de paquete respuesta bloque

PrLec/- = Procesador Lee y no se genera ningún paquete

b) Indique si las transiciones que hay marcada son todas correctas. En caso contrario, suprima las incorrectas. Añadir si falta alguna transición.

c) ¿Qué evento puede provocar que se vuelva al estado I?