

Ingeniería de Computadores

Enero 2017

Nombre: _____

Grupo: _____

Normas de realización:

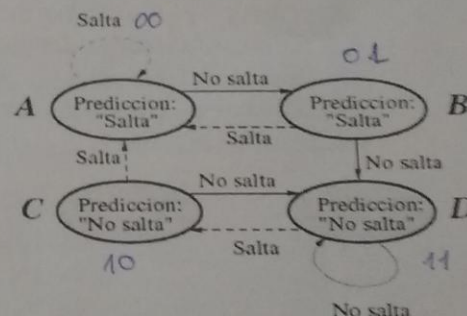
- Incluir el nombre en todas las hojas utilizadas
- Todas las respuestas han de ser correctamente detalladas y razonadas.
- Las respuestas deben estar escritas con bolígrafo negro o azul

Pregunta 1 (1,5 ptos). Contesta a las siguientes preguntas:

- (1 pto) Explica qué dos tipos de buffer de renombrado podemos encontrar y cuáles son las ventajas de cada tipo.
- (0,5 ptos) Explica cómo se realiza el encaminamiento en una red CCC (no hace falta dibujarla)

Pregunta 2 (2 ptos). Una universidad ha adquirido un supercomputador formado por 32768 nodos conectados mediante una red malla abierta 3D cuyos enlaces tienen una velocidad de 4Gbit/s. Para terminar de analizar el rendimiento del supercomputador se desea saber cuánto tardará un paquete formado por 24 bytes (incluyendo la cabecera) que se envía desde el nodo 1015 al nodo 22222. El tiempo de enrutamiento es de 27ns. Calcula los tiempos de envío tanto utilizando "store and forward" como "wormhole". Nota: la cabecera del paquete está formada por 4 bytes.

Pregunta 3 (2,5 ptos) Suponer un computador superescalar que dispone un buffer de reorden, que permite resolver los riesgos WAR y WAW, y una ventana de instrucciones con un número de entradas suficiente. El procesador es capaz de decodificar, emitir y completar 3 instrucciones por ciclo. Además, la emisión de las instrucciones puede ser desordenada y dispone de unidad de adelantamiento. Para las tareas de ejecución, se dispone de las siguientes unidades segmentadas: 2 FP mul/div (5c), 2 FP add (2c), 2 ALU int (1) y 2 load/store (3). Finalmente, se dispone de un predictor de saltos dinámico que utiliza BTB de 4 entradas y 2 bits de predicción. Cuando se añade una nueva entrada en el BTB, su primera predicción siempre sería de estado A (salto efectivo).



En el computador se ejecuta el siguiente fragmento de programa:

```
; r1 almacena la dirección de a
; r2 almacena la dirección de b
    addi r3,r1,#80 ; condicion de final
    addi r1,r1,#8 ; inicialización de los indices
    addi r2,r2,#8 ;
    addi r5,r1,#3;
    ld f0,coef ; cargar coeficiente
loop: ld f2,-8(r1) ; cargar a[i-1]
      ld f4,0(r1) ; cargar a[i]
      beqz r5, fin    r5 = 2
      muld f8,f2,f0 ; a[i-1]*coef
      divd f9,f2,f0 ;
      addd f4,f8,f4 ; a[i-1]*coef + a[i]
      sd 0(r2),f4 ; almacenar b[i]
      addi r1,r1,#8 ; incrementar indices
      addi r2,r2,#8
      subi r5,r5,#1
      slt r4,r1,r3
      bnez r4,loop
fin: subd f2,f1,f3
```

$r1 = 0$
 $r2 = 100$

$\rightarrow r5 = 0$ fin $r5 = 1$

set less than
si $! = 0$ loop

- a) (1,5 pts) Planificar las instrucciones utilizando una tabla como la siguiente hasta la primera iteración del bucle (sin realizar el salto). Suponer que inicialmente $r1=0$ y $r2=100$

inst	IF	ID/ISS	EX	ROB	WB	Comentario

- b) (0,5 pts) Realizar una traza de ejecución del código, mostrando el contenido de la BTB, (BTB inicialmente vacía) para todas las iteraciones del bucle.

Dir salto	Dir destino	Bits predicción

- c) (0,25 pts) ¿Existe alguna penalización en la ejecución del código? Si es así, indica con qué instrucción y cuándo
- d) (0,25 pts) Determinar el número de ciclos que tardaría en ejecutarse el código

Pregunta 4 (1 pto) Explica la diferencia entre predicción dinámica explícita y predicción dinámica implícita.

última vez

Historia

Pregunta 5 (1.75 puntos). En la paralelización de una aplicación orientada a una máquina paralela de memoria distribuida, un ingeniero ha descompuesto dicha aplicación en 12 tareas, que nombramos como $T_1, T_2 \dots T_{12}$, donde el subíndice indica el **orden de ejecución de cada tarea en la versión secuencial**. Seguidamente nos indica que los grupos de tareas T_2 a T_6 (ambas incluidas) y T_8 a T_{10} (ambas incluidas) **son independientes entre sí**.

Por último, nos informa del porcentaje de tiempo que toma cada tarea por separado en la versión secuencial (ver tabla adjunta). Suponga que disponemos de 2 multicomputadores con 2 y 4 nodos respectivamente (**todos los nodos iguales**), conectados entre sí con una red cuya sobrecarga se puede modelar como $T_{\text{overhead}}(p) = 0.05 \cdot p$ (donde p es el número de procesadores).

T_1 : 15%	T_5 : 8%	T_9 : 9%
T_2 : 15%	T_6 : 10%	T_{10} : 8%
T_3 : 5%	T_7 : 5%	T_{11} : 4%
T_4 : 12%	T_8 : 7%	T_{12} : 2%

Se pide:

- a) Dibuje el grafo de precedencia entre tareas y calcule la fracción no paralelizable del problema (0.25 puntos)
- b) ¿Cuál es la **máxima** ganancia en velocidad que se puede obtener con cada uno de los multicomputadores? (1 punto)
- c) ¿En qué cluster es más eficiente la ejecución de nuestra aplicación con la paralelización propuesta? (0.5 puntos)

Nota: Indique claramente cuál es la asignación de tareas a los distintos nodos que maximiza la ganancia en velocidad. Explique clara y pormenorizadamente cada paso que dé en la resolución del problema.

Pregunta 6 (1.25 puntos):

Una cierta línea de caché de uno de los nodos de un multiprocesador equipado con un sistema de caché que implementa el protocolo MESI pasa al estado I. Indique **todo** los que podemos saber (mecanismo ha llevado a la línea a cambiar de estado, porqué, posible estado o estados previos, ...) y qué consecuencias tiene cambiar al estado I (¿en qué casos se ha de compartir y con quién el contenido de dicha línea?).