

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 (2 ptos). Contesta a las siguientes preguntas:

- (1 pto) Dibuja una red mariposa de 8 entradas y 8 salidas con conmutadores 2x2
- (1 pto) En qué consiste la predicción dinámica implícita. Pon un ejemplo.

Pregunta 2 (2 ptos). Un banco ha adquirido un supercomputador formado por 32768 nodos conectados mediante una red toro 3D cuyos enlaces tienen una velocidad de 2Gbit/s. Para terminar de analizar el rendimiento del supercomputador se desea saber cuánto tardará un paquete formado por 56 bytes (incluyendo la cabecera) que se envía desde el nodo 3056 al nodo 12018. El tiempo de enrutamiento es de 9ns. Calcula los tiempos de envío tanto utilizando "store and forward" como "wormhole". Nota: la cabecera del paquete está formada por 2 bytes.

Pregunta 3 (1,5 ptos). Conteste a las siguientes preguntas:

- (0,75 ptos) Un sistema multiprocesador compuesto por N nodos monoprocesador con caché utiliza el protocolo MESI para mantener la coherencia de sus cachés. Una determinada línea de caché de uno de sus procesadores está inicialmente en estado M. Indique para las siguientes órdenes en el bus que observa la controladora de caché para dicha línea de caché: cambio de estado (indicando qué significa cada estado y en qué estado está la memoria principal con respecto a la/s línea/s de caché) y el flujo de información (qué memorias o cachés se actualizan y quién recibe o da la información)
 - BusRdX
 - BusRd
- (0,75 ptos) ¿Puede una línea de caché del anterior computador pasar del estado S al estado E mediante una orden de procesador (PrWr o PrRd)? Explique razonada y pormenorizadamente su respuesta.

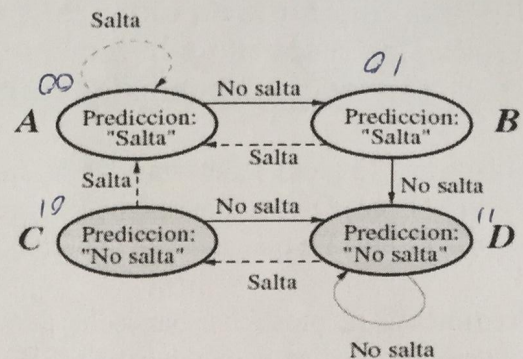
Pregunta 4 (1,5 ptos). Un estudiante de IC ha paralelizado mediante MPI una cierta aplicación. Después del proceso de paralelización ha visto que el 10% de la aplicación no se puede paralelizar. El restante 90% está definido por un gran ciclo for que sí es paralelizable para cualquier número de nodos. Se pide (explique razonadamente cada apartado):

- (0,5 ptos) ¿A partir de qué número de procesadores obtenemos ganancias estrictamente mayores que 4? Suponga que el tiempo de sobrecarga es nulo.
- (1 pto) Demuestre que si suponemos un tiempo de sobrecarga linealmente dependiente del número de nodos con los que paraleliza (p), el tiempo de cómputo paralelo no siempre mejora con p, sino que necesariamente debe llegar a degradarse a partir de un cierto número de nodos (p').

Pregunta 5 (1,5 puntos). Conteste a las siguientes preguntas:

- (0,75 puntos) Explica en qué consiste el encadenamiento de operaciones en las máquinas vectoriales
- (0,75 puntos) ¿Para qué se utilizan las estructuras buffer de renombrado y buffer de reorden de los procesadores superescalares?

Pregunta 6 (1.5 puntos). Suponer un computador superescalar que dispone un buffer de reorden, que permite resolver los riesgos WAR y WAW, y una cola y ventana de instrucciones con un número de entradas suficiente. El procesador es capaz de captar 3 instrucciones por ciclo, decodificar, emitir y completar 2 instrucciones por ciclo. La ejecución implementa adelantamiento. Además, la emisión y finalización de las instrucciones puede ser desordenada. 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 3 entradas y 2 bits de predicción. Cuando se añade una nueva entrada en el BTB, su primera predicción sería de estado A (salto efectivo) si el salto es hacia atrás y de estado D (saldo no efectivo) si el salto es hacia adelante.



En el computador se ejecuta el siguiente fragmento de programa:

```

0x01      addi r3,r0,#4 ; r3=4
0x02      add r4,r0,r3 ; r4=r3
0x03      loop: subi r3,r3,#1 ; r3=r3-1
0x04      beqz r3,end; si r3=0 saltar end
0x05      ld f1,-8(r1) ; cargar a[i-1]
0x06      ld f2,0(r1) ; cargar a[i]
0x07      ld f3,0(r2); cargar b[i]
0x08      muld f4,f1,f2 ; a[i-1]*a[i]
0x09      muld f4,f4,f3 ; a[i-1]* a[i] * b[i]
0x0A      sd 0(r1),f4 ; almacenar a[i]
0x0B      addi r1,r1,#8; r1=r1+8
0x0C      addi r2,r2,#8; r2=r2+8
0x0D      bnez r4,loop -
0x0E      end: addd f4, f4, f4

```

Handwritten notes:
 r_1, r_2, r_3, r_4

3

2

1

- a) (1 punto) 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 puntos) 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