

3. SEGMENTACIÓN ENCAUZADA

3.1. ¿Por qué los cauces aritméticos son típicos de los computadores vectoriales? ¡2!

3.2. ¿Cuál es el principal inconveniente de la memoria entrelazada simple? ¿Cuál es el principal inconveniente de la memoria entrelazada, en general, frente a la memoria caché? ¡3!

3.3. En un cauce síncrono de cuatro etapas hay un 20% de probabilidades de que exista un riesgo que detenga el cauce durante dos ciclos de reloj y un 5% de probabilidades de un riesgo que detenga el cauce durante un ciclo de reloj. El 75% restante de probabilidades corresponde al caso en que no haya riesgos. A) ¿Cuál es la mejora del rendimiento respecto a un sistema equivalente no encauzado cuyo ciclo de reloj sea tres veces mayor, si no tenemos en cuenta los riesgos? B) ¿Cuál es esa mejora si tenemos en cuenta los riesgos? ¡4!

3.4. Supongamos un cauce síncrono de k segmentos. Sea p la probabilidad de encontrar una instrucción de salto y q la probabilidad de que al ejecutar una instrucción de salto se rompa la secuencia normal. A) Si cada vez que esto ocurre hubiera que esperar a que la instrucción de salto salga del cauce para que entre la nueva instrucción, ¿cuál sería el número medio de instrucciones que saldrían del cauce por ciclo de reloj? B) Si en cada ruptura de secuencia, se puede cargar la nueva instrucción el 80% de las veces cuando la instrucción de salto termina su segunda etapa y el 20%, cuando la instrucción de salto termina su tercera etapa, ¿cuál será ahora la productividad? C) ¿Cuánto ha mejorado si $pq=0,1$ y $k=4$? ¿Y si $k=5$? ¡4!

3.5. Representa el diagrama etapas-tiempo para la secuencia de instrucciones:

S1: $I \equiv \text{Reg}[3] \leftarrow \text{Mem}[5]$
 $J \equiv \text{Reg}[2] \leftarrow \text{NOT}(\text{Mem}[\text{Reg}[3]])$
 $K \equiv \text{Reg}[3] \leftarrow \text{AND}(\text{Reg}[3], \text{Reg}[2])$
 $L \equiv \text{Mem}[\text{Reg}[3]] \leftarrow \text{Reg}[1]$

y el diagrama instrucciones-tiempo para la secuencia

S2: $I \equiv \text{Reg}[4] \leftarrow \text{Mem}[5]$
 $J \equiv \text{Reg}[2] \leftarrow \text{NOT}(\text{Mem}[\text{Reg}[4]])$
 $K \equiv \text{Mem}[\text{Reg}[4]] \leftarrow \text{Reg}[2]$

En ambos casos, supóngase existencia de cortocircuito, cachés separadas sin fallos y detección dinámica de las colisiones. ¡6!

3.6. La siguiente secuencia de instrucciones se ejecuta en nuestro procesador con cortocircuito y cachés separadas sin fallos:

```
I1 ≡  Reg[2] ← ADD (Reg[2], Mem[3])
I2 ≡  Mem[5] ← Reg[2]
I3 ≡  CMP (Reg[2], Reg[3])
I4 ≡  if EQUAL then goto IS
I5 ≡  ...
...
IS ≡  ...
```

Supongamos que se cumple la condición de salto. Dibuja el diagrama instrucciones-tiempo cuando se predice salto efectivo y el diagrama etapas tiempo cuando se predice salto no efectivo. Los diagramas terminarán cuando IS termine FPC. ¡6!