

## 2. SEGMENTACIÓN ENCAUZADA: EJERCICIOS Y CUESTIONES

**2.1.** En una CPU encauzada la dirección de salto es conocida en la tercera etapa y la condición de salto una etapa después. Completa la siguiente tabla con las correspondientes penalizaciones, en ciclos, del salto/bifurcación.

| Tratamiento de saltos           | Penalización salto (incondicional) | Penalización bifurcación no efectiva (condicional) | Penalización bifurcación efectiva (condicional) |
|---------------------------------|------------------------------------|----------------------------------------------------|-------------------------------------------------|
| Parar instrucciones posteriores |                                    |                                                    |                                                 |
| Predecir salto no efectivo      |                                    |                                                    |                                                 |
| Predecir salto efectivo         |                                    |                                                    |                                                 |

**2.2.** Supongamos que los saltos incondicionales son un 5% de todas las instrucciones mientras que los condicionales (bifurcaciones) son un 20%. Además los condicionales provocan salto un 60% de las veces. La dirección de salto se conoce al final del segundo ciclo en los saltos incondicionales, en tanto que las bifurcaciones se resuelven al final del tercer ciclo. Suponiendo que únicamente la primera etapa del cauce puede estar activa desde que un salto/bifurcación entra en el cauce hasta que se resuelve, calcula cuánto aumenta el CPI en relación a una máquina sin riesgos de control. (Ignorar otros posibles riesgos).

**2.3.** Considera el siguiente fragmento de código:

```
loop: LW    R1, 0(R2)
      ADDI  R1, R1, #1
      SW    R1, 0(R2)
      ADDI  R2, R2, #4
      SUB   R4, R3, R2
      BNZ   R4, loop
```

Asume que el valor inicial de R3 es R2+396 y que, mientras no se diga lo contrario, todos los accesos a caché son aciertos.

- A. Supón que no hay cortocircuito y que las instrucciones de bifurcación no son retardadas y actualizan el PC en MEM. Además, cada vez que se detecta una bifurcación se detienen las instrucciones posteriores hasta que se conoce la instrucción por la que seguir. Dibuja un diagrama instrucciones-tiempo correspondiente a la ejecución en DLX del fragmento de código considerado. Dibuja sólo la parte necesaria para conocer cuantos ciclos requiere la ejecución completa del bucle.

- B. Responde a la misma cuestión (diagrama y número de ciclos) suponiendo que hay cortocircuito y se predice salto no efectivo.
- C. Con los mismos supuestos del apartado anterior excepto que los saltos son retardados con un ciclo de retardo, planifica las instrucciones del bucle (sólo se permite cambiar instrucciones de sitio y modificar operandos en las instrucciones). Una vez optimizado el código dibuja un diagrama instrucciones-tiempo y calcula el número total de ciclos que supone la ejecución del bucle.

**2.4.** Considera el siguiente fragmento de código:

```

loop: LD      F0, 0(R2)
      LD      F4, 0(R3)
      MULTD   F0, F0, F4
      ADDD    F2, F0, F2
      ADDI    R2, R2, #8
      ADDI    R3, R3, #8
      SUB     R5, R4, R2
      BNZ     R5, loop

```

Asume que el valor inicial de R4 es R2+792 y que, en caso de que varias instrucciones lleguen a WB al mismo tiempo, tiene prioridad la primera instrucción en el orden secuencial.

- A. Muestra el diagrama instrucciones-tiempo correspondiente a la ejecución del código considerado en DLX FP sin cortocircuito. Asumir que las bifurcaciones se tratan deteniendo las instrucciones posteriores hasta conocer el destino. ¿Cuántos ciclos supone la ejecución completa del bucle?
- B. Responde a la misma cuestión suponiendo que hay cortocircuito y se predice salto no efectivo.
- C. Con bifurcaciones retardadas un ciclo y cortocircuito planifica las instrucciones, dibuja un diagrama instrucciones-tiempo y calcula el número total de ciclos que supone la ejecución del bucle.

**2.5.** Utiliza la información de las transparencias 14 y 24 sobre DLX FP (DLX multiciclo) para calcular la relación entre el número medio de paradas por operación de punto flotante y la latencia de la correspondiente operación (completa la tabla siguiente) ¿Qué consecuencia general se extrae de estos cálculos?

|                          | Dependencias RAW |          |        | Riesgo estructural |
|--------------------------|------------------|----------|--------|--------------------|
|                          | +/-/convert      | Multiply | Divide | Divide             |
| Nº medio paradas         |                  |          |        |                    |
| Latencia                 |                  |          |        |                    |
| % paradas sobre latencia |                  |          |        |                    |

**2.6.** En R4000 una instrucción de suma en punto flotante ha finalizado la etapa RF. ¿Con cuántos ciclos de antelación ha tenido que ser emitida una instrucción de multiplicación para que la instrucción de suma tenga que ser detenida (no emitida) para evitar colisionar con ella? Ídem en el caso de una instrucción de división emitida con antelación. Utiliza la información, sobre el R4000, de la transparencia 33.

**2.7.** Las interrupciones de entrada/salida pueden ser tratadas de un modo más sencillo que las internas con retorno. ¿Cómo?

**2.8.** Típicamente una rutina de tratamiento de una interrupción interna necesita conocer los valores de entrada de la instrucción interruptora. En casos así es prácticamente imprescindible que la interrupción sea precisa. ¿Por qué?