Test Diseño del Procesador: Rendimiento

1. Si el procesador A tiene una frecuencia de 1GHz y el procesador B una frecuencia de 2GHz, entonces el rendimiento de A es menor que el de B. Indica si es verdadero o falso y razona tu respuesta.

   Falso, el número de instrucciones y ciclos por instrucción puede ser menor para el procesador A.

   $$T = N * CPI * T_C$$

2. Si el procesador A tiene una frecuencia de 1GHz y el procesador B una frecuencia de 2GHz, y ambos tienen el mismo repertorio de instrucciones, entonces el rendimiento de A es menor que el de B. Indica si es verdadero o falso y razona tu respuesta.

   Falso, los ciclos por instrucción pueden ser menos en el procesador A.

   $$T = N * CPI * T_C$$

3. Si el procesador A tiene una frecuencia de 1GHz y el procesador B una frecuencia de 2GHz, y ambos tienen la misma microarquitectura, entonces el rendimiento de A es menor que el de B. Indica si es verdadero o falso y razona tu respuesta.

   Verdadero.

   $$T = N * CPI * T_C$$

Test Diseño del Procesador: Operaciones multiciclo

4. Dado un procesador segmentado en 5 etapas como el visto en clase con las siguientes características:
   - Los datos se pueden leer y escribir en el banco de registros en el mismo ciclo de reloj
   - Incorpora la técnica de forwarding o cortocircuito para la anticipación de operandos
   - Dispone de un sumador/restador en punto flotante no segmentado con una latencia de 4 ciclos

   Se pide calcular el CPI correspondiente a la ejecución del siguiente código.

LD F0,0(R1)	IF	ID	E	M	WB											
LD F2,8(R1)		IF	ID	E	M	WB										
ADDD F4,F0,F2			IF	Dp	ID	A1	A2	A3	A4	M	WB					
SUBD F6,F0,F2				Fp	IF	Dp	Dp	Dp	ID	A1	A2	A3	A4	M	WB	
ADDI R1,R1,#8						Fp	Fp	Fp	IF	ID	E	M	WB			
SD F6,0(R2)										IF	Dp	Dp	ID	E	M	WB

CPI= 16 ciclos/6 instrucciones

5. Dado el procesador segmentado de la cuestión anterior, si su frecuencia de reloj es 1GHz, ¿cuál es su rendimientos en MFLOPS al ejecutar el código anterior?

   MFLOPS = 2 instrucciones en punto flotante/(16 ciclos/10^9ciclos/segundo) =

   = 2000/16 MFLOPS

6. Dado un procesador segmentado en 5 etapas como el visto en clase con las siguientes características:
   - Los datos se pueden leer y escribir en el banco de registros en el mismo ciclo de reloj
   - Incorpora la técnica de forwarding o cortocircuito para la anticipación de operandos
   - Dispone de un sumador/restador en punto flotante segmentado con una latencia de 4 ciclos

   ¿Cuál es la diferencia en número de ciclos con respecto al mismo código ejecutado en el procesador de la actividad anterior?

LD F0,0(R1)	IF	ID	E	M	WB											
LD F2,8(R1)		IF	ID	E	M	WB										
ADDD F4,F0,F2			IF	Dp	ID	A1	A2	A3	A4	M	WB					
SUBD F6,F0,F2				Fp	IF	ID	A1	A2	A3	A4	M	WB				
ADDI R1,R1,#8						IF	ID	E	M	WB						
SD F6,0(R2)							IF	Dp	Dp	ID	E	M	WB			

La diferencia en ciclos es 3.

7. Compara las 2 soluciones estudiadas en clase para los riesgos EDE en función del número de ciclos de ejecución y de la seguridad que ofrecen.

   - Primera solución: parar la segunda instrucción en la etapa de decodificación hasta que la primera termine su etapa de ejecución, es decir, la etapa de ejecución de la segunda instrucción comienza cuando termine la de la primera.

   - Segunda solución: inhibir la escritura de la primera instrucción, de modo que sólo se almacene el resultado de la segunda, que es el correcto.

   En primer lugar, la primera solución es más lenta, ya que obliga a parar el pipeline. Y aunque a primera vista parece que es más segura, la segunda opción no implica ningún riesgo, ya

que el único problema que podríamos tener es que alguna instrucción entre medias (o la segunda instrucción) necesitasen el valor generado por la primera. Realmente esto no es un problema si tenemos implementada la técnica de anticipación de operandos (fordwarding) ya que las instrucciones que necesitasen ese dato lo recibirían directamente de la etapa de ejecución de la instrucción (no sería necesario leerlo del registro en el que no se ha escrito). TENDRÍAMOS QUE MODIFICAR LA LÓGICA DEL FORWARDING PARA DETECTAR QUE NO NECESITAMOS EL DATO ALMACENADO EN EL REGISTRO PERO SÍ EL DATO GENERADO POR LA PRIMERA INSTRUCCIÓN.

¿Qué sucedería si ocurre una excepción entre las dos instrucciones con dependencia EDE? El banco de registros se encontraría en un estado de inconsistencia en el momento del salto a la subrutina que atiende la excepción. Esto podría ser un problema si la subrutina no detecta que los datos del banco de registros no son correctos. En consecuencia, SI SE PRODUCEN EXCEPCIONES LA SEGUNDA OPCIÓN PUEDE NO SER PRECISA.

Test Diseño del Procesador: Gestión de excepciones

8. Para la gestión de excepciones, ¿qué modificaciones hacen falta en la ruta de datos para que la CPU pueda ejecutar la rutina de tratamiento de la excepción?

   Añadimos una nueva entrada en el multiplexor PCmux con la dirección de la rutina de tratamiento de excepciones “4000040”, de este modo cuando se produce una excepción saltamos a la rutina de tratamiento de excepciones.

9. ¿Cómo detecta la rutina de tratamiento de excepción qué rutina de servicio en particular debe ejecutarse para resolver cada excepción?

   Cuando ocurre una excepción su causa se almacena en el registro Cause de la ruta de datos, y la rutina de tratamiento de excepciones lee el valor almacenado en este registro.

10. ¿Qué implicaciones en cuestión de seguridad crees que puede tener el hecho de que no se atiendan las excepciones hasta el final de la ejecución de la instrucción que la produjo?

    La implicación más importante es que incluso habiendo algún error en la operación en curso que ha causado la excepción, la instrucción continúa ejecutándose. Incluso si NO se le permitiera cambiar los valores de los registros ni de la memoria, podría realizar algunos cambios en el estado del sistema (por ejemplo traer datos a la memoria cache).

Problemas Diseño del Procesador Multiciclo

Problema 1: Dado un procesador segmentado en 5 etapas con las siguientes características:

- Los datos se pueden leer y escribir en el banco de registros en el mismo ciclo de reloj
- No incorpora la técnica de forwarding o cortocircuito para la anticipación de operandos
- Los saltos se resuelven en la etapa de decodificación y la siguiente instrucción empieza a ejecutarse y se descarta en caso de tomarse el salto

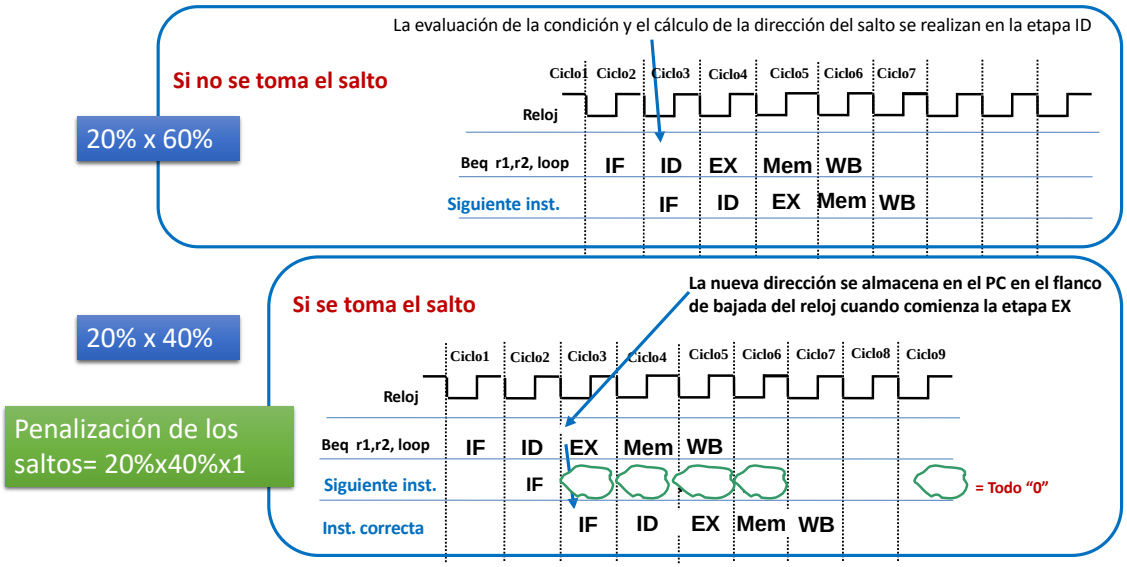
Se ejecuta en él una aplicación con las siguientes características:

- El 20% de las instrucciones son saltos condicionales, y se ellos se toman el 40%. No existen dependencias de datos con las instrucciones de salto.
- El 18% de las veces, la instrucción Ii+1 tiene una dependencia LDE con la instrucción Ii (30% son instrucciones load)
- El 6% de las veces, la instrucción Ii+2 tiene una dependencia LDE con la instrucción Ii (30% de ellas son instrucciones load). En estos casos no existe dependencias de datos entre las instrucciones Ii+1 y Ii+2

Se pide calcular:

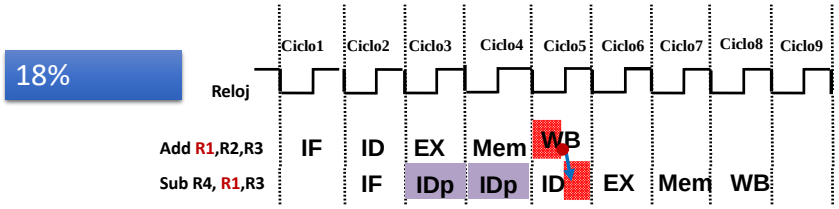
a) CPI
b) CPI tras añadir la técnica de forwarding o cortocircuito
c) Speedup del procesador a) frente al b)

20% de las instrucciones son saltos condicionales, de los cuales se toman el 40%. Las instrucciones de salto no tienen dependencias de datos.

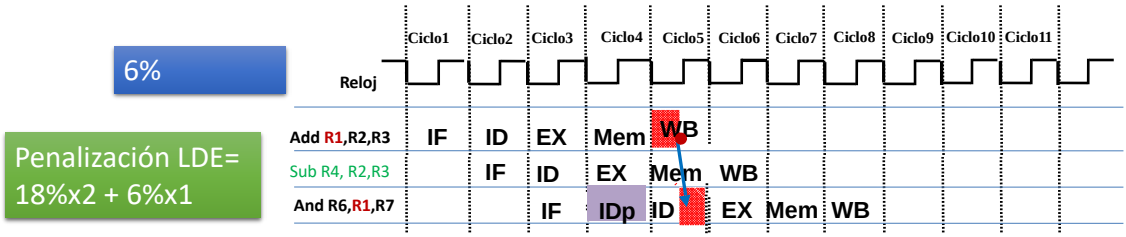


LDE. Sin anticipación de operandos (forwarding).

18% de las veces la instrucción I_{i+1} tiene una dependencia LDE con la instrucción I_i (30% de éstas son instrucciones de load).

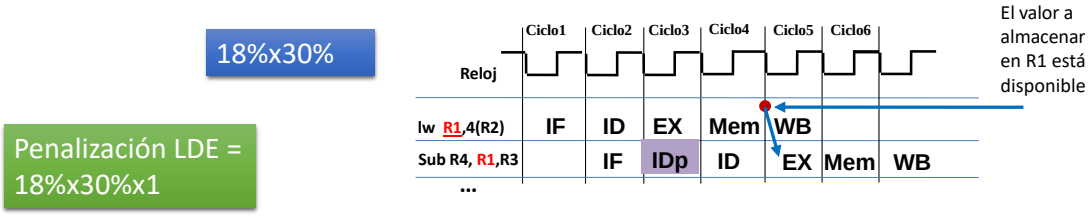


6% de las veces la instrucción I_{i+2} tiene una dependencia LDE con la instrucción I_i (30% de éstas son instrucciones de load), y en estos casos nunca hay dependencias entre las instrucciones I_{i+1} y I_{i+2}.

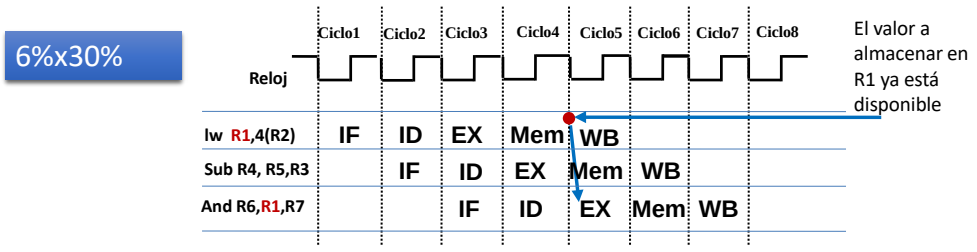


Si añadimos forwarding solo las instrucciones load producen paradas en el pipeline.

18% de las veces la instrucción I_{i+1} tiene una dependencia LDE con la instrucción I_i (30% de éstas son instrucciones de load).



6% de las veces la instrucción I_{i+2} tiene una dependencia LDE con la instrucción I_i (30% de éstas son instrucciones de load), y en estos casos nunca hay dependencias entre las instrucciones I_{i+1} y I_{i+2}.



a) CPI = 1 + Penalización saltos + Penalización LDE
Penalización saltos = 20% x 40% x 1
Penalización LDE = 18% x 2 + 6% x 1
CPI = 1 + 0,08 + 0,42 = 1,5

b) CPI con forwarding
Penalización LDE = 18% x 30% x1
CPI = 1 + 0,08 + 0,054 = 1,134

c) SPEEDUP de a) frente al b)
Speedup = CPIa / CPIb = 1,32

Problema 2. Dado un procesador segmentado en 5 etapas con las siguientes características:

- Los datos se pueden leer y escribir en el banco de registros en el mismo ciclo de reloj
- La detección de riesgos y paradas del pipeline se realizan en la etapa de decodificación
- Incorpora la técnica de forwarding o cortocircuito para la anticipación de operandos
- Los saltos se resuelven en la etapa de decodificación y se resuelven mediante paradas
- Los riesgos estructurales referentes a la etapa de acceso a memoria se resuelven mediante paradas en la último ciclo de la etapa de ejecución
- Los riesgos EDE se resuelven mediante la inhibición de la escritura de la primera instrucción

Se ejecuta el siguiente programa:

```
LD      F10,0(R1)
MULD    F4,F0,F10
LD      F12,0(R2)
ADDD    F2,F12,F4
LD      F4,8(R1)
MULD    F12,F4,F12
LD      F12,16(R1)
```

a) Representa el diagrama de instrucción/tiempo señalando los cortocircuitos
b) Calcula el CPI

a)

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
LD F10,0(R1)	IF	ID	EX	MEM	WB															
MULD F4,F0,F10		IF	IDp	ID	X1	X2	X3	X4	X5	MEM	WB									
LD F12, 0(R2)				IF	ID	EX	MEM	WB												
ADDD F2,F12,F4					IF	IDp	IDp	IDp	ID	A1	A2	MEM	WB							
LD F4, 8(R1)									IF	ID	EXP	EX	MEM	WB						
MULD F12,F4,F12										IF	IDp	IDp	ID	X1	X2	X3	X4	X5	MEM	WB
LD F12, 16(R1)													IF	ID	EX	MEM	WB			

b) CPI = 20/7