



Tema 4: Segmentación *-pipelining-* 2ª parte

Arquitectura de Computadores
Grado en Ingeniería Informática

Mercedes Rodríguez García

Índice

1. Segmentación *-pipelining-*
2. Consideraciones en el diseño del repertorio de instrucciones
3. Camino de datos de un procesador segmentado
4. Representaciones gráficas de la segmentación
5. Riesgos del pipeline
 - 5.1. Riesgos estructurales
 - 5.2. Riesgos de datos
 - 5.2.1. Métodos para resolver los riesgos de datos
 - 5.2.2. Tipos de riesgos de datos
 - 5.3. Riesgos de control
 - 5.3.1. Métodos para resolver los riesgos de control
6. ILP avanzado
 - 6.1. Estrategias para incrementar ILP
 - 6.2. Ejecución múltiple con planificación estática
 - 6.3. Ejecución múltiple con planificación dinámica
7. Casos reales

Bibliografía

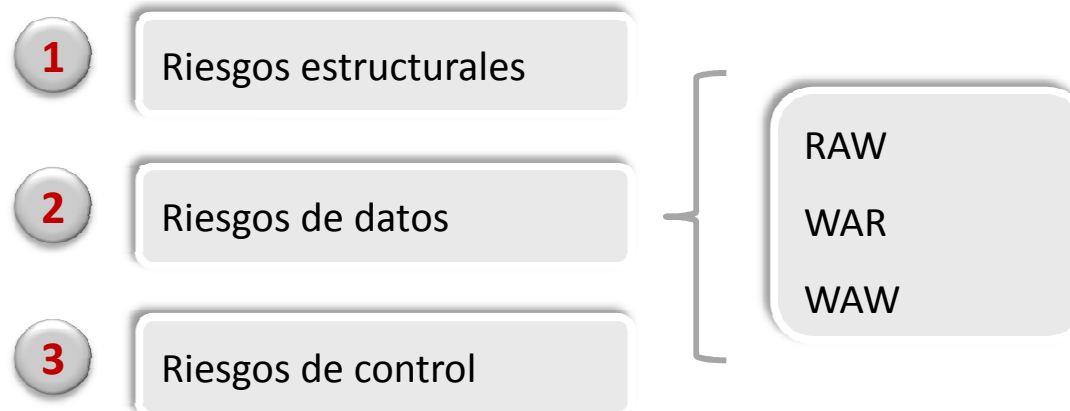
Estructura y diseño de computadores: Capítulo 4.
Patterson y Hennessy. Editorial Reverte, 2011.

5. Riesgos del pipeline

1. Segmentación *-pipelining-*
2. Consideraciones en el diseño del repertorio de instrucciones
3. Camino de datos de un procesador segmentado
4. Representaciones gráficas de la segmentación
5. Riesgos del pipeline
 - 5.1. Riesgos estructurales
 - 5.2. Riesgos de datos
 - 5.2.1. Métodos para resolver los riesgos de datos
 - 5.2.2. Tipos de riesgos de datos
 - 5.3. Riesgos de control
 - 5.3.1. Métodos para resolver los riesgos de control
6. ILP avanzado
 - 6.1. Estrategias para incrementar ILP
 - 6.2. Ejecución múltiple con planificación estática
 - 6.3. Ejecución múltiple con planificación dinámica
7. Casos reales

5. Riesgos del pipeline

Riesgo -Hazard- situación que obliga a interrumpir el flujo continuo de instrucciones en el pipeline. Según la causa, se distinguen tres tipos de riesgos:



5.1. Riesgos estructurales

Se producen cuando dos instrucciones del pipeline necesitan utilizar a la vez **la misma unidad funcional**.



¿Qué ocurriría si NO tuviéramos una memoria de datos e instrucciones separada?

5.2. Riesgos de datos RAW

En el pipeline, una instrucción depende del resultado de una instrucción anterior, pero éste no está disponible todavía.

RAW: Read After Write

Estado del pipeline en un instante dado:

1º) Write.- En el pipeline hay una instrucción que realiza una **escritura** en el registro X (registro destino), pero todavía no ha llegado a la etapa WB (etapa en la que se produce la escritura). Esta instrucción puede provocar un riesgo y la llamaremos **instrucción H**.

2º) Read.- En el pipeline, una instrucción posterior a la instrucción H, realiza la **lectura** de ese registro X (registro fuente), pero el registro todavía no tiene el valor correcto. A esta instrucción la llamaremos **instrucción dependiente**.



¿Qué instrucciones pueden provocar un riesgo RAW?

5.2. Riesgos de datos RAW

Ejemplo

Sea la siguiente secuencia de instrucciones:

```
LW $1, 0($10)  
LW $2, 4($10)  
SUB $3, $1, $2  
ADD $4, $3, $3
```



5.2. Riesgos de datos RAW

Problema 1

Descubre las dependencias en esta secuencia de instrucciones. Los valores de los registros son \$1 = 10, \$2 = 10, \$3 = 30, \$5 = 20

¿Qué valor debería tener el registro \$2 en la instrucción SUB?

	Acción que realiza la instrucción
add \$2, \$1, \$3	
sub \$12, \$2, \$5	



Pero, ¿Qué valor es el que se lee del registro \$2 cuando la instrucción SUB está en la etapa ID?

5.2. Riesgos de datos RAW

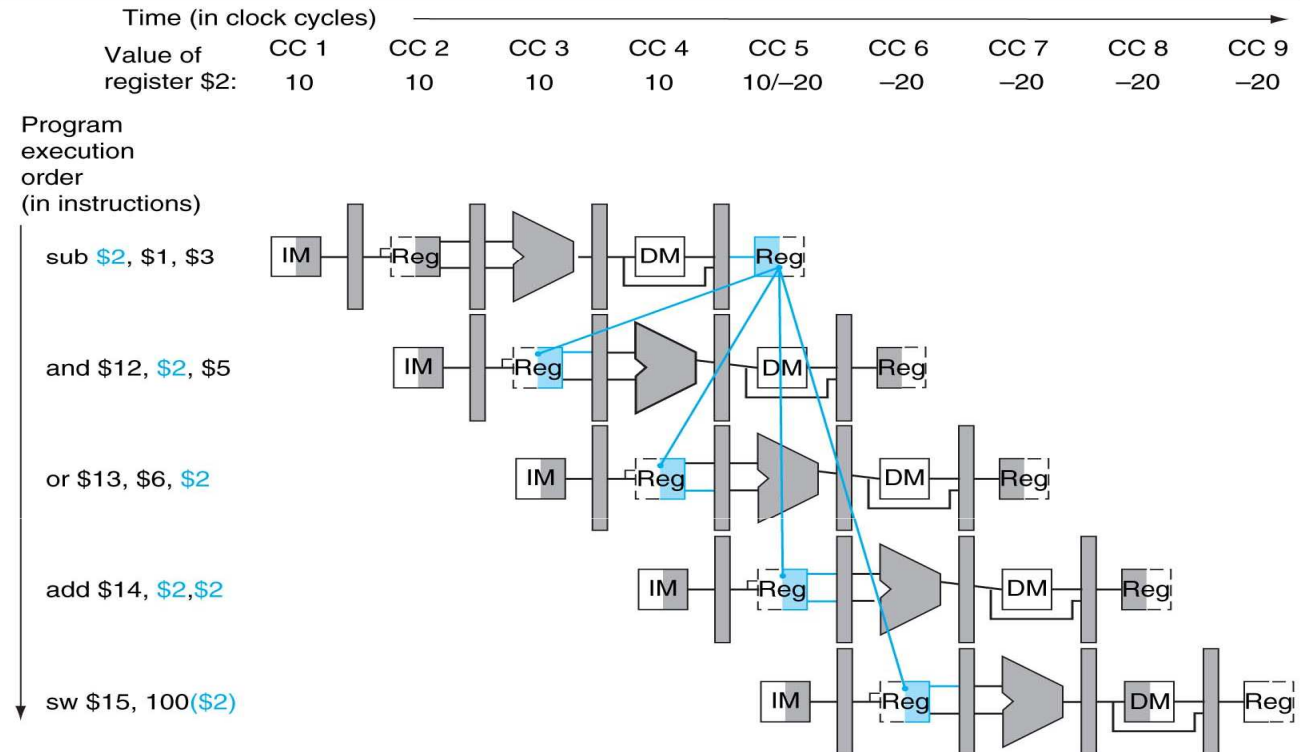
Problema 2

Descubre las dependencias en esta secuencia de instrucciones. Los valores de los registros son \$1 = 20, \$2 = 10, \$3 = 40

	Acción que realiza la instrucción
Sub \$2, \$1, \$3	
And \$12, \$2, \$5	
Or \$13, \$6, \$2	
Add \$14, \$2, \$2	
Sw \$15, 100(\$2)	

5.2. Riesgos de datos RAW

Problema 2



-Figura obtenida de [PATT11]-

- 1.- ¿Cuándo se escribe el resultado de la resta en \$2?
- 2.- ¿Qué instrucciones obtendrían un valor incorrecto de \$2?
- 3.- ¿Cuándo se escribe el resultado de la resta en EX/MEM?
- 4.- Para las instrucciones del punto 2, ¿qué ocurre si conseguimos anticipar el resultado de la resta a la ALU?
- 5.- ¿Cómo se podría implementar la anticipación?

5.2.1. Métodos para resolver riesgos de datos RAW

1

BLOQUEO del pipeline –*pipeline stall*- o burbuja

GESTIONADO POR EL PROCESADOR

La instrucción que provoca el riesgo (instrucción H) sigue avanzando en el pipeline. Las instrucciones que siguen a la instrucción H se detienen en el pipeline hasta que se resuelva el riesgo, es decir, hasta que se realice la escritura en el registro destino.

2

ANTICIPACIÓN del resultado –*forwarding*-

GESTIONADO POR EL PROCESADOR

La instrucción que provoca el riesgo (instrucción H) proporciona el resultado a las instrucciones dependientes antes de que éste se almacene en el registro destino. ¿Cómo? mediante los registros de segmentación EX/MEM o MEM/WB, según el caso. Incluso con el uso de esta técnica, a veces hay que bloquear el pipeline.

3

REORDENACIÓN de código

GESTIONADO POR EL COMPILADOR

El compilador reorganiza el código insertando instrucciones independientes entre las que tienen dependencias para distanciarlas y conseguir un retardo. Si no existen instrucciones que puedan cambiar su orden sin modificar el resultado del programa, el compilador podrá insertar instrucciones NOP (esta instrucción no realiza nada).

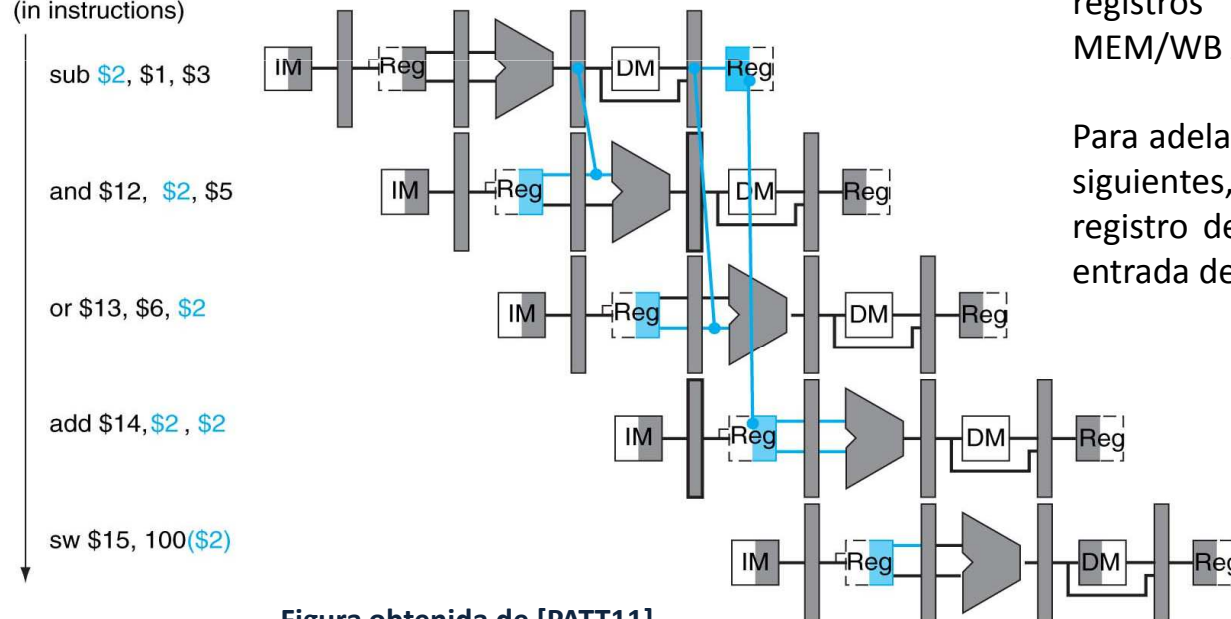
5.2.1. Métodos para resolver riesgos de datos RAW

Ejemplo

¿Cómo quedaría el pipeline del problema 2 si implementamos la anticipación del resultado?

	Time (in clock cycles) →								
	CC 1	CC 2	CC 3	CC 4	CC 5	CC 6	CC 7	CC 8	CC 9
Value of register \$2:	10	10	10	10	10/-20	-20	-20	-20	-20
Value of EX/MEM:	X	X	X	-20	X	X	X	X	X
Value of MEM/WB:	X	X	X	X	-20	X	X	X	X

Program execution order (in instructions)

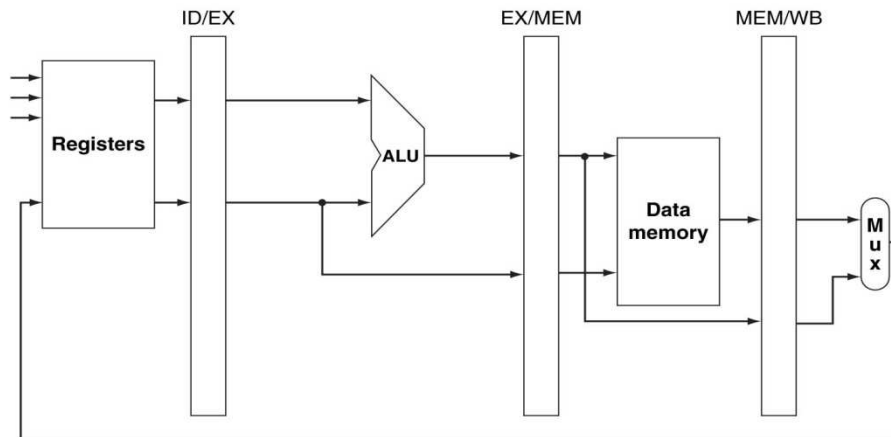


-Figura obtenida de [PATT11]-

El resultado de la resta está disponible en los registros de segmentación EX/MEM y MEM/WB **ANTES** que en el registro destino.

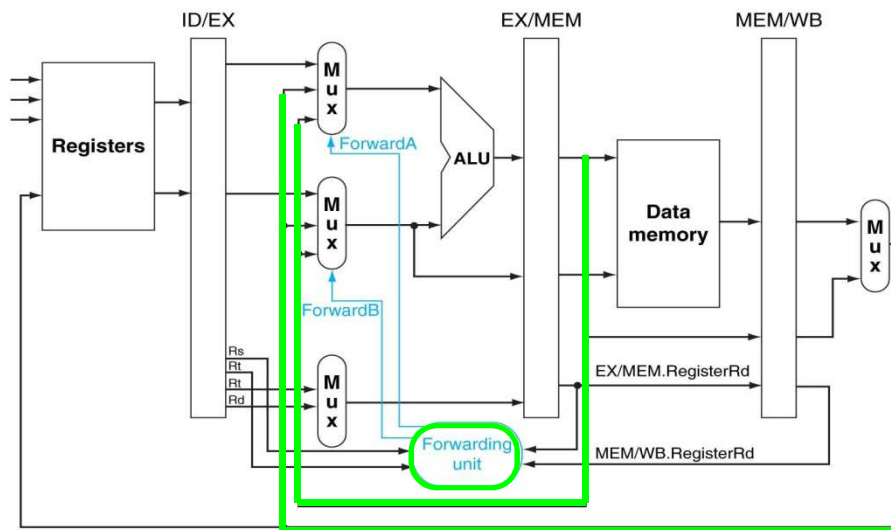
Para adelantar el resultado a las instrucciones siguientes, sólo hay que enviarlo desde el registro de segmentación apropiado hasta la entrada de la ALU.

5.2.1. Métodos para resolver riesgos de datos RAW



a. No forwarding

Camino de datos sin anticipación



b. With forwarding

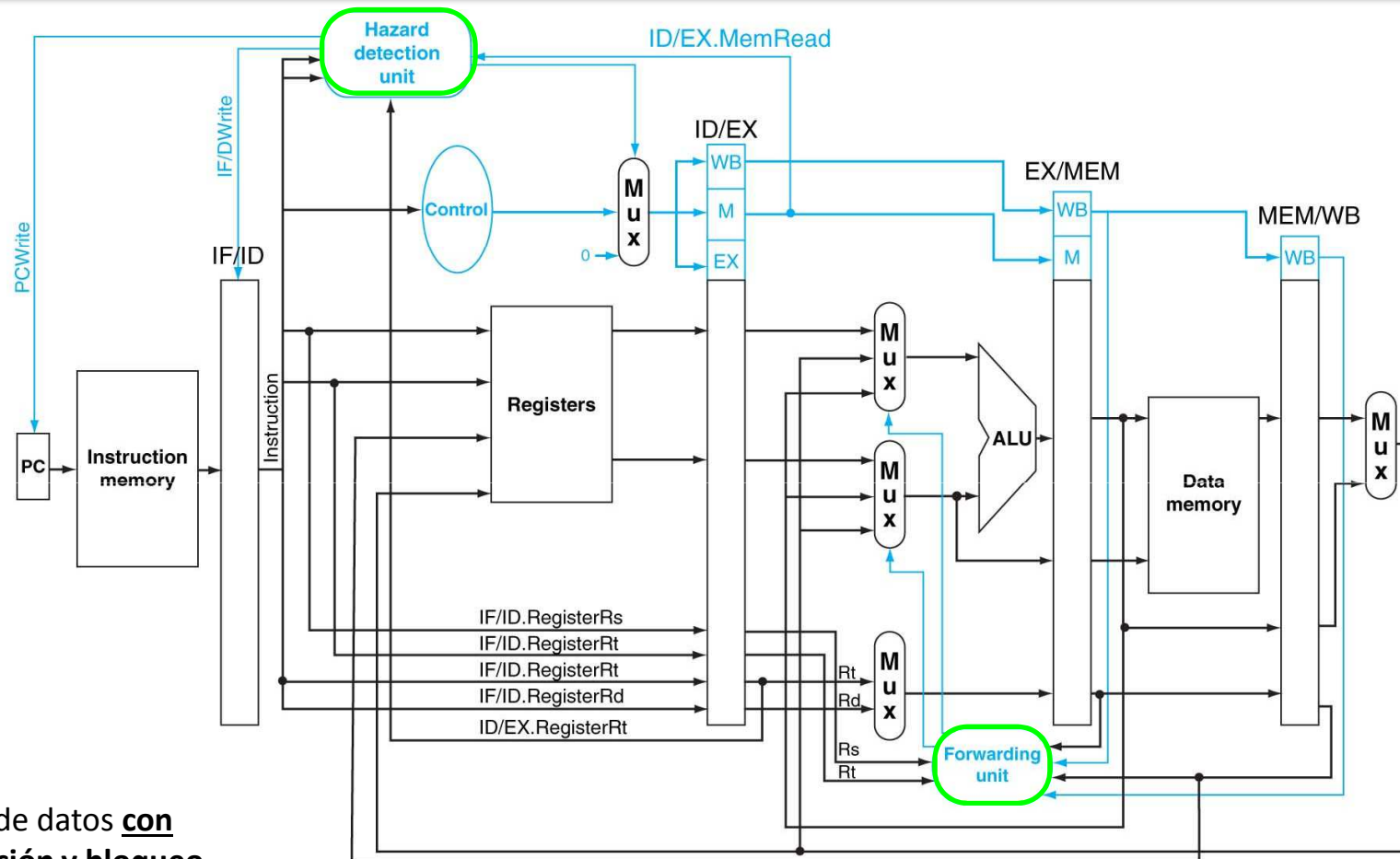
Camino de datos con anticipación

Para adelantar el dato hay que implementar una unidad de control adicional llamada **unidad de anticipación** -forwarding unit- y una serie de multiplexores en la entrada de la ALU.

La unidad de anticipación es capaz de detectar si hay que anticipar algún dato estudiando los identificadores de los registros en una y otra instrucción.

-Figuras obtenidas de [PATT11]-

5.2.1. Métodos para resolver riesgos de datos RAW



Camino de datos con anticipación y bloqueo

-Figura obtenida de [PATT11]-

Para implementar un pipeline que sea capaz de realizar bloqueos hay que añadir al procesador una unidad de control especial llamada **unidad de detección de riesgos** –*hazard detection unit*– que detecte si hay que insertar burbujas cuando la instrucción dependiente llegue a la etapa ID.

5.2.1. Métodos para resolver riesgos de datos RAW

Problema 3

Descubre las dependencias en esta secuencia de instrucciones. Valores: \$2 = 10 y el contenido de la posición de memoria \$1+20 es 99.

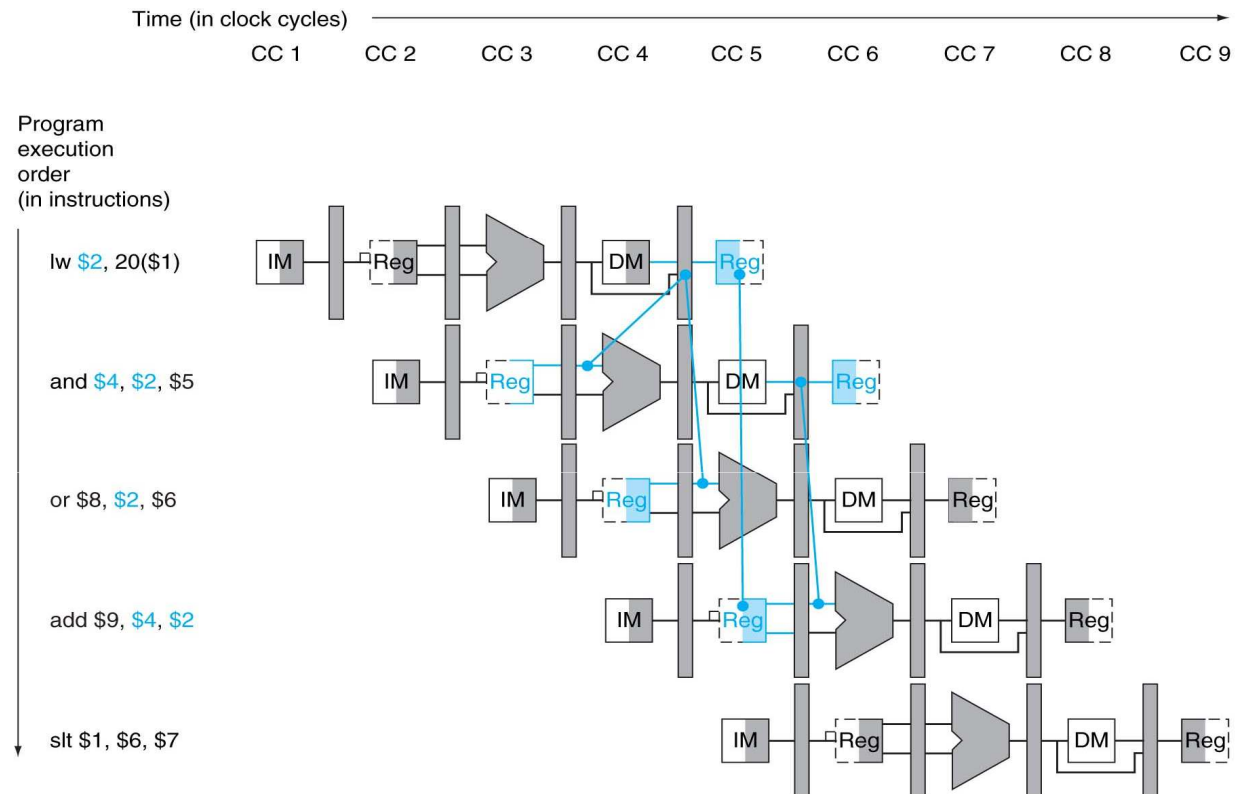
Señala todas las dependencias.

¿Qué valor debería tener el registro \$2 en la instrucción AND?

	Acción que realiza la instrucción
Lw \$2, 20(\$1)	
And \$4, \$2, \$5	
Or \$8, \$2, \$6	
Add \$9, \$4, \$2	
Slt \$1, \$6, \$7	

5.2.1. Métodos para resolver riesgos de datos RAW

Problema 3



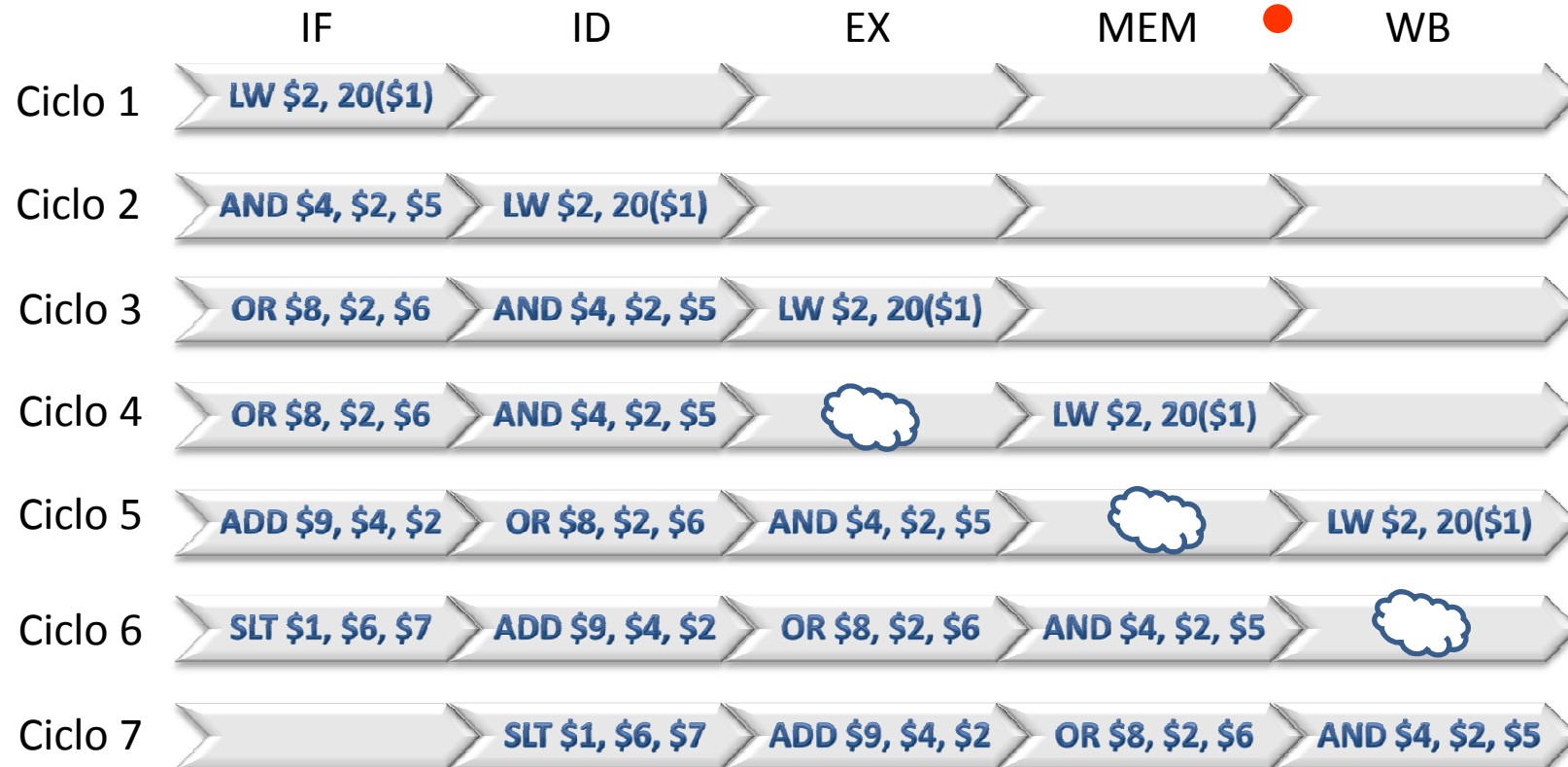
-Figura obtenida de [PATT11]-

- 1.- ¿Cuándo se almacena en \$2 el dato obtenido de memoria?
- 2.- ¿Qué instrucciones obtendrían el valor correcto de \$2? Identifica las dependencias que suponen un riesgo.
- 3.- ¿En qué registro de segmentación se almacena por primera vez el dato obtenido de memoria?
- 4.- ¿Con la anticipación se solucionan todos los riesgos identificados? ¿Hay que bloquear el pipeline?

5.2.1. Métodos para resolver riesgos de datos RAW

Problema 3

En este punto, el dato obtenido de memoria ya se ha almacenado en el registro MEM/WB

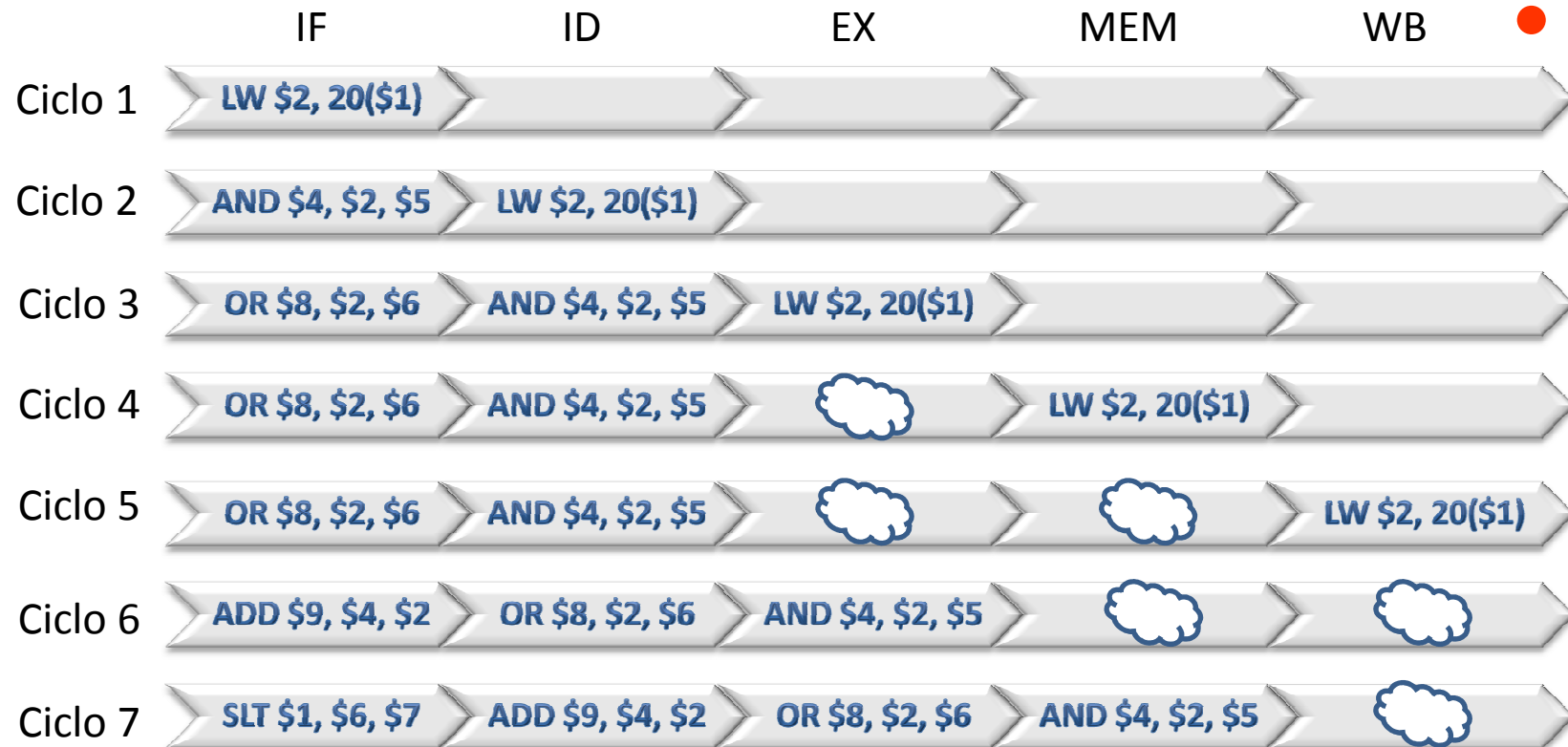


-Con anticipación-

5.2.1. Métodos para resolver riesgos de datos RAW

Problema 3

En este punto, el dato obtenido de memoria ya se ha almacenado en \$2



-Sin anticipación-

5.2.1. Métodos para resolver riesgos de datos RAW

Sobre las burbujas ...

Cuando se produce un bloqueo y se inserta **una burbuja** en el pipeline, las instrucciones que vienen detrás de la burbuja se detienen exactamente **un ciclo de reloj**.

Como una burbuja de aire en una tubería de agua, una burbuja retrasa todo lo que viene detrás, y avanza por el pipeline hasta salir por su extremo final.



¿Habéis encontrado alguna norma que identifique cuándo hay que insertar una burbuja ?