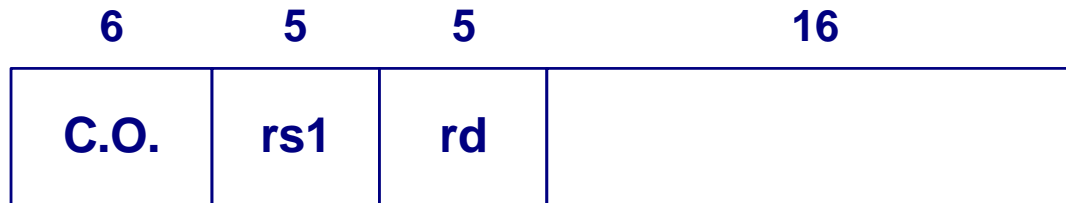
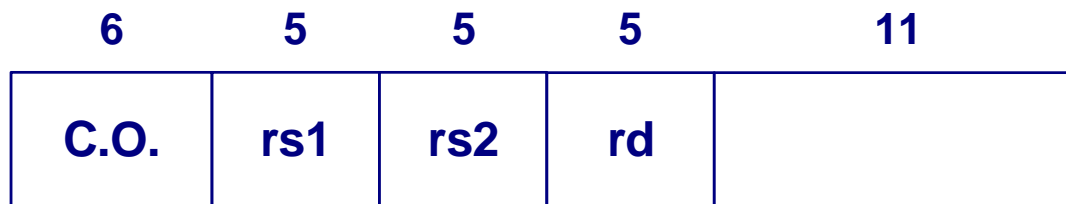




Formatos de Instrucción en el MIPS R3000



Carga /Almacenamiento
Saltos Condicionales



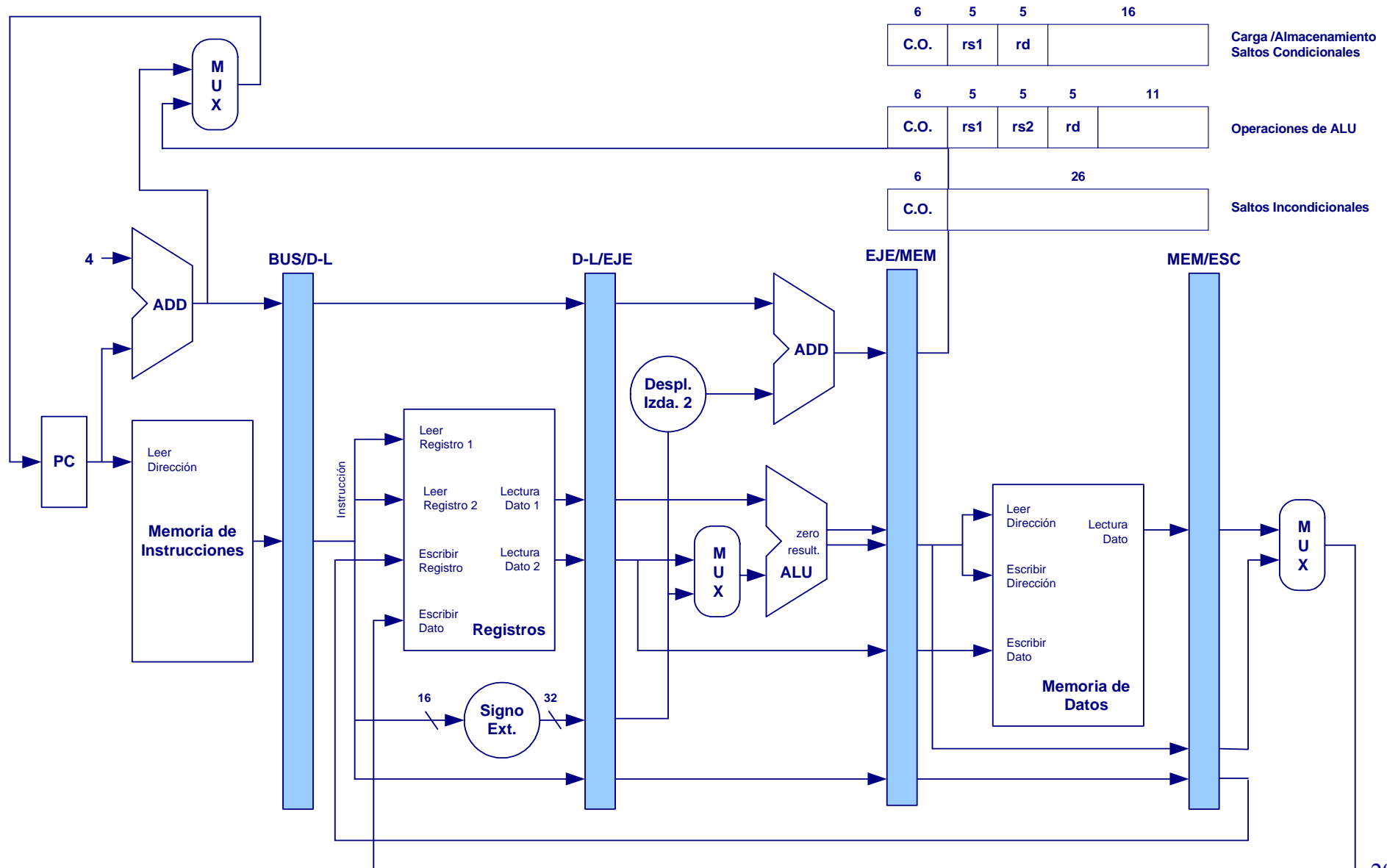
Operaciones de ALU



Saltos Incondicionales

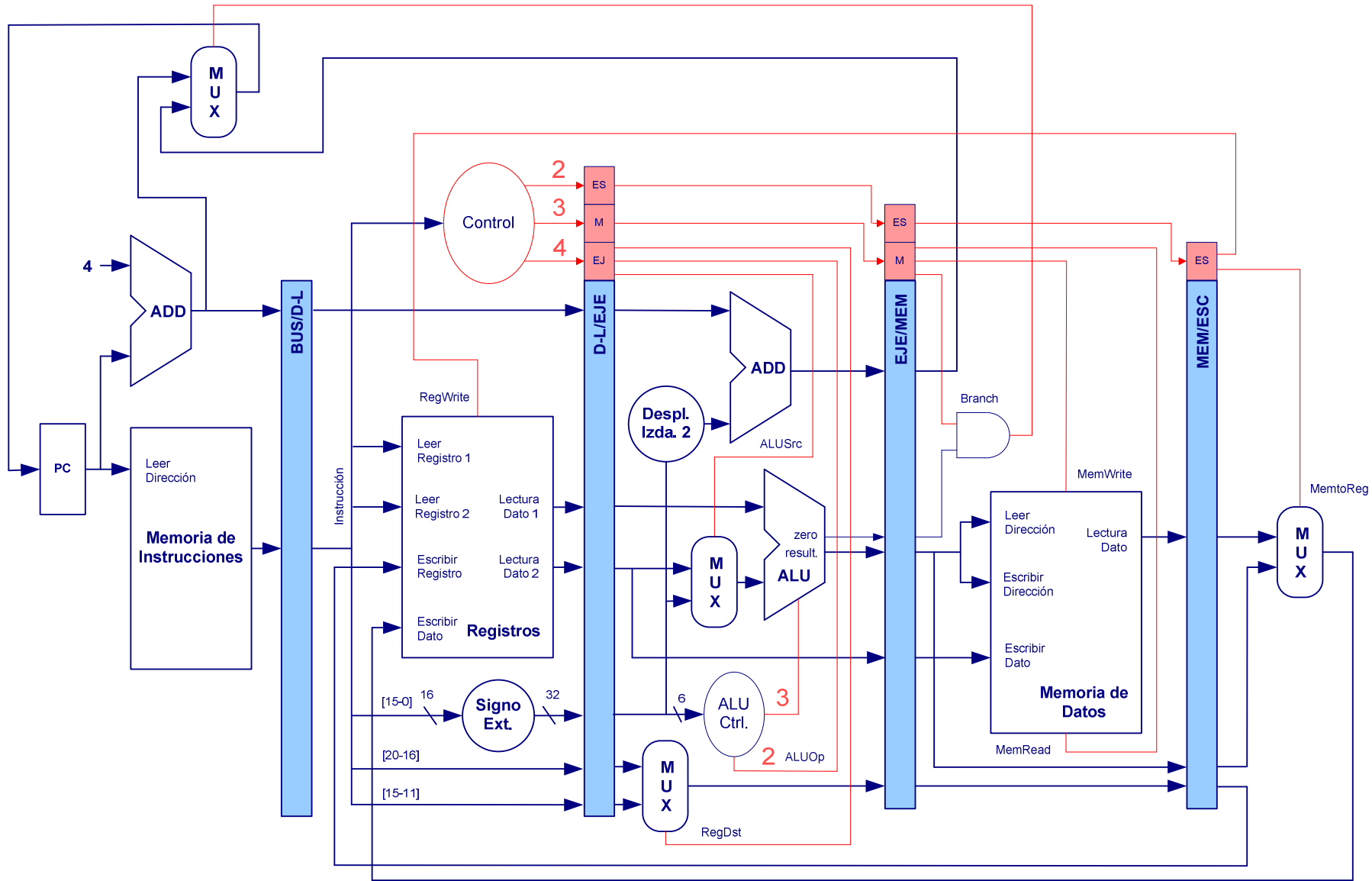


Implementación de la Segmentación





Control de la Segmentación





Riesgos por Dependencias de Datos

- Problema de la dependencia de datos
- Tipos de dependencias de datos
- Dependencias en el MIPS R3000
- Métodos de resolución
 - ✓ Resolución estática (en tiempo de compilación)
 - ✓ Resolución dinámica (mediante hardware adicional)



Problema de la Dependencia de Datos

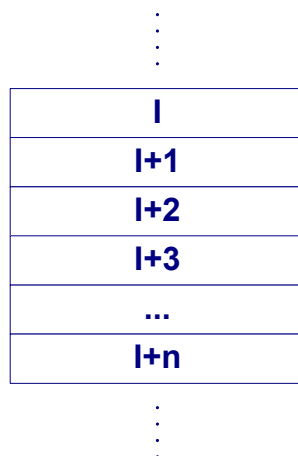
★ Las Lecturas/Escrituras sobre una misma variable (registro) deben efectuarse en el orden que especifica el programa secuencial

➔ Dominio de una Instrucción (**D[I]**):

Conjunto de variables (registros) sobre las que la instrucción realiza lecturas

➔ Rango de una Instrucción (**R[I]**):

Conjunto de variables (registros) sobre las que la instrucción realiza escrituras

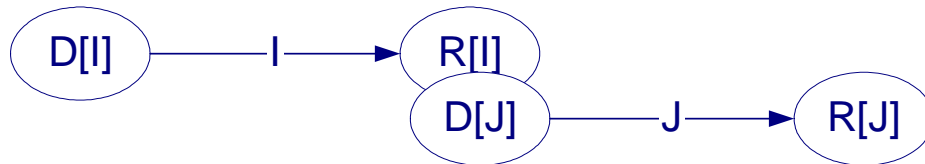


J =



Tipos de Dependencias de Datos

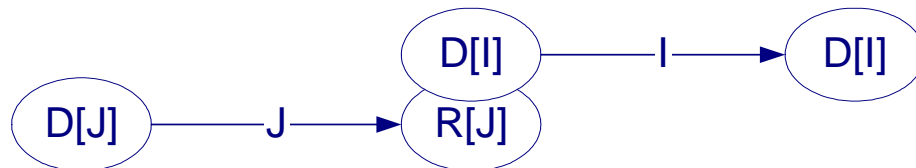
★ **Lectura despues de Escritura (Verdadera)**



Ejemplo:

R1 = ...
... = R1

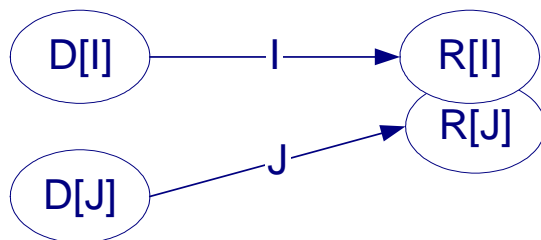
★ **Escritura despues de Lectura (Antidependencia)**



Ejemplo:

... = R1
R1 = ...

★ **Escritura despues de Escritura (de salida)**

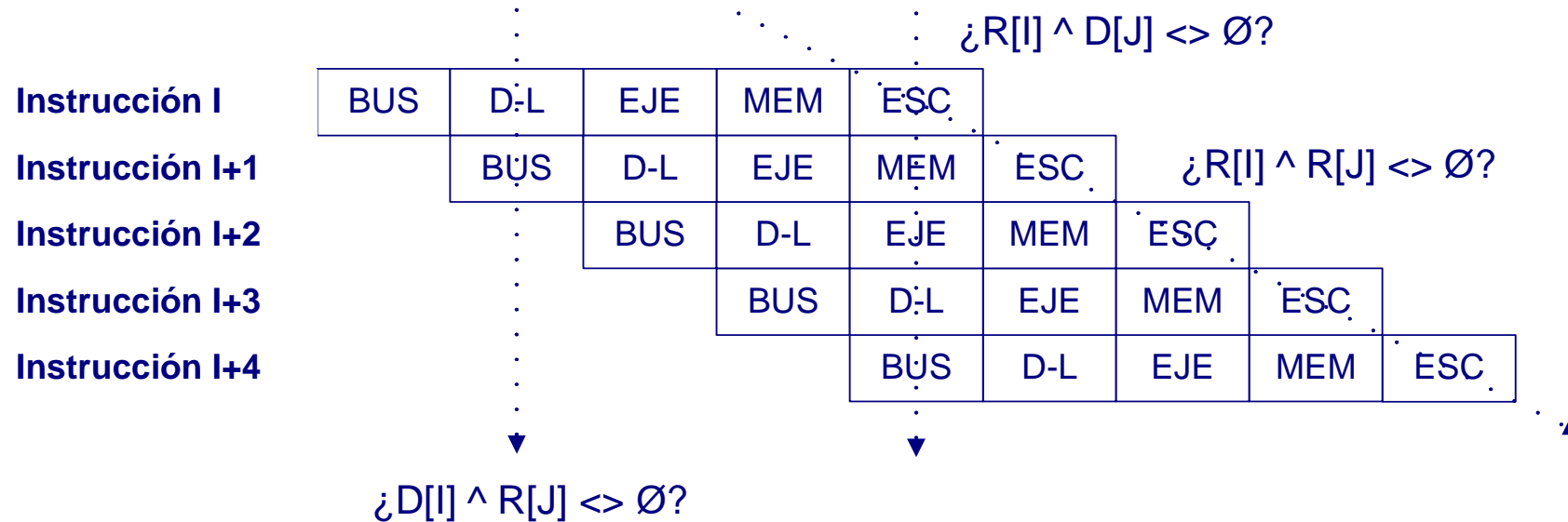


Ejemplo:

R1 = ...
R1 = ...



Dependencias en el MIPS R3000



- ★ **Escritura despues de Lectura** → ¿Puede alguna instrucción I+n escribir antes de que I lea? →
- ★ **Escritura despues de Escritura** → ¿Puede alguna instrucción I+n escribir antes de que I escriba? →
- ★ **Lectura despues de Escritura** → ¿Puede alguna instrucción I+n leer antes de que I escriba? →



Ejemplo de Dependencia LDE

Formato de Instrucciones MIPS:

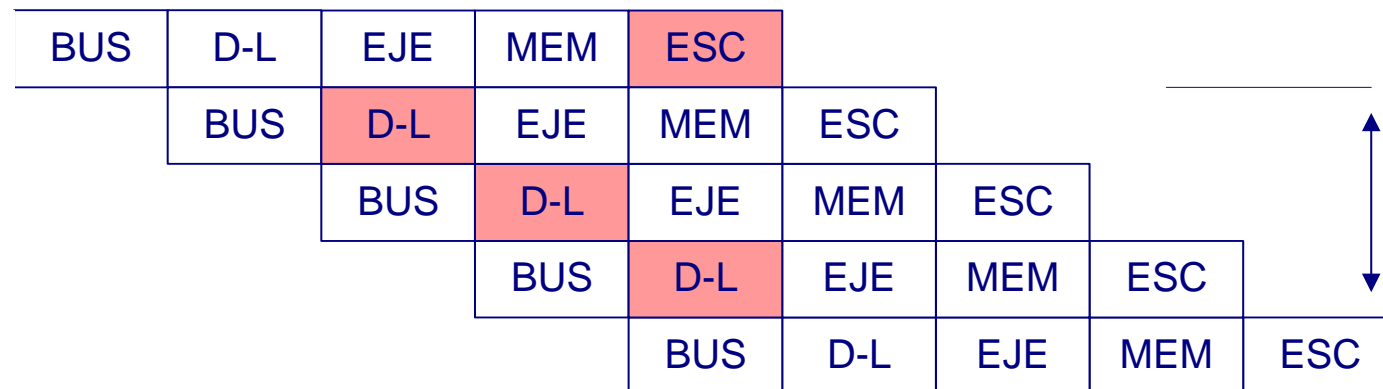
ADD R1, R2, R3

SUB R4, R1, R5

AND R6, R1, R7

OR R8, R1, R9

XOR R10, R1, R11



Número máximo de instrucciones afectadas =



Resolución Estática (I)

★ Inserción de Instrucciones NOP

ADD R1, R2, R3

NOP

NOP

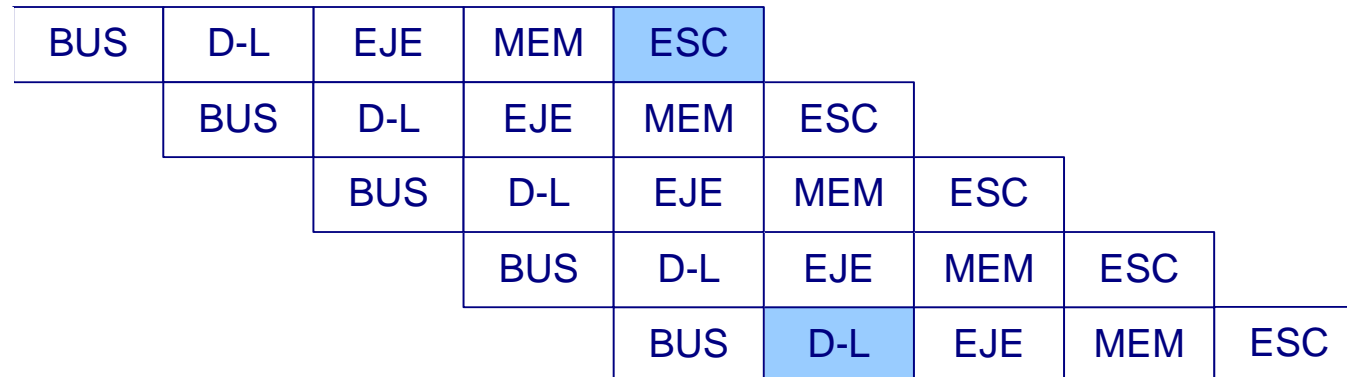
NOP

OR R8, R1, R9

SUB R4, R1, R5

AND R6, R1, R7

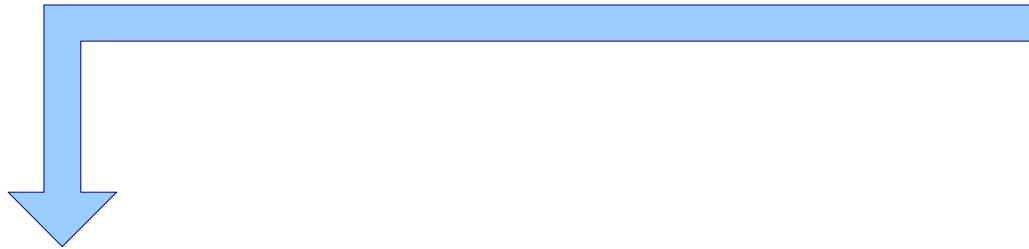
XOR R10, R1, R11





Resolución Estática (II)

★ Reordenación de Código



ADD R1, R2, R3

SUB R4, R1, R5

AND R6, R1, R7

OR R8, R1, R9

XOR R10, R1, R11

AND R12, R12, R13

SUB R14, R15, R14

ADD R16, R17, R18

ADD R1, R2, R3

AND R12, R12, R13

SUB R14, R15, R14

ADD R16, R17, R18

SUB R4, R1, R5

AND R6, R1, R7

OR R8, R1, R9

XOR R10, R1, R11

BUS	D-L	EJE	MEM	ESC				
	BUS	D-L	EJE	MEM	ESC			
		BUS	D-L	EJE	MEM	ESC		
			BUS	D-L	EJE	MEM	ESC	
				BUS	D-L	EJE	MEM	ESC





Resolución mediante Hardware (I)

★ Detención del cauce

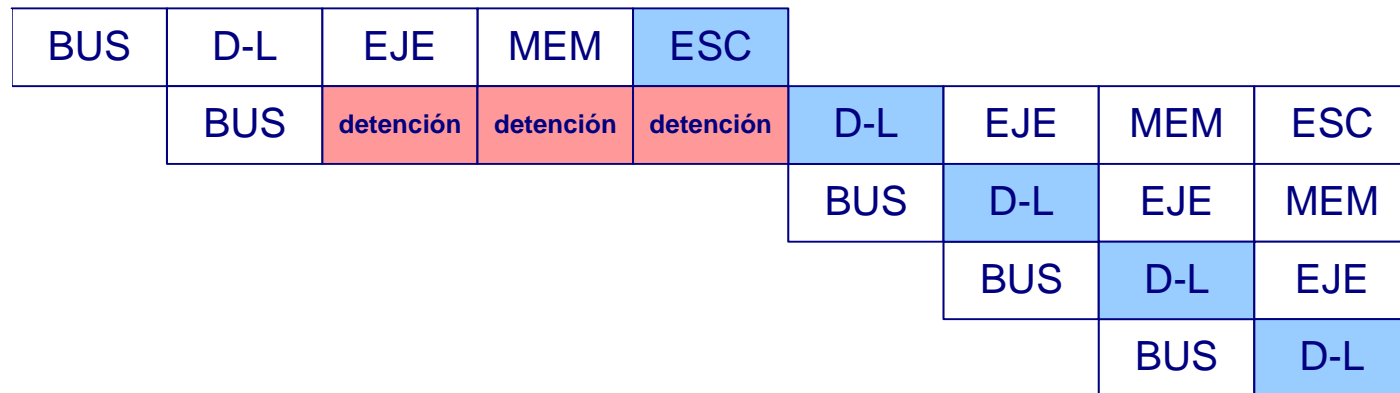
ADD R1, R2, R3

SUB R4, R1, R5

AND R6, R1, R7

OR R8, R1, R9

XOR R10, R1, R11

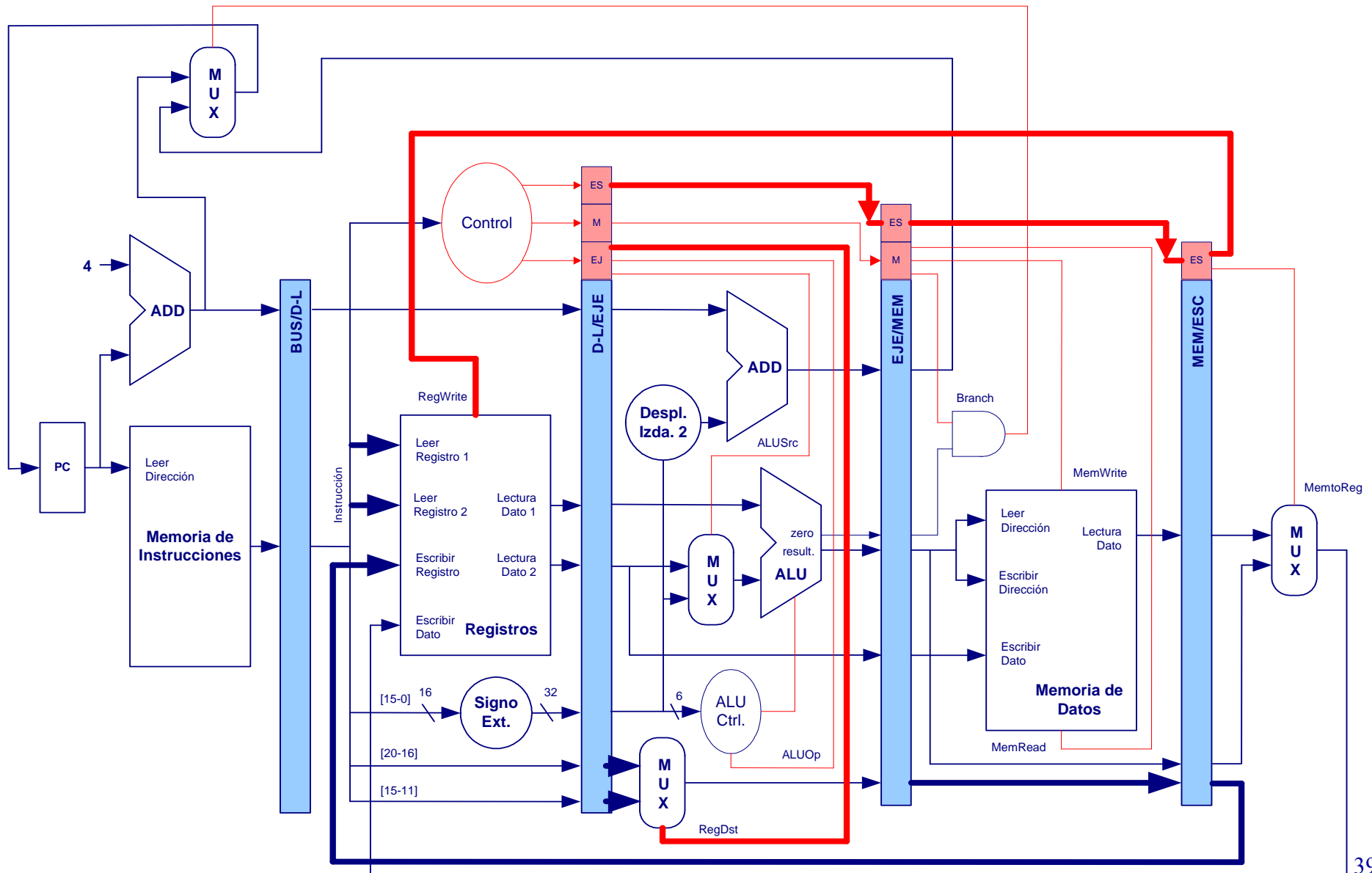


Alternativa Hardware mas sencilla:



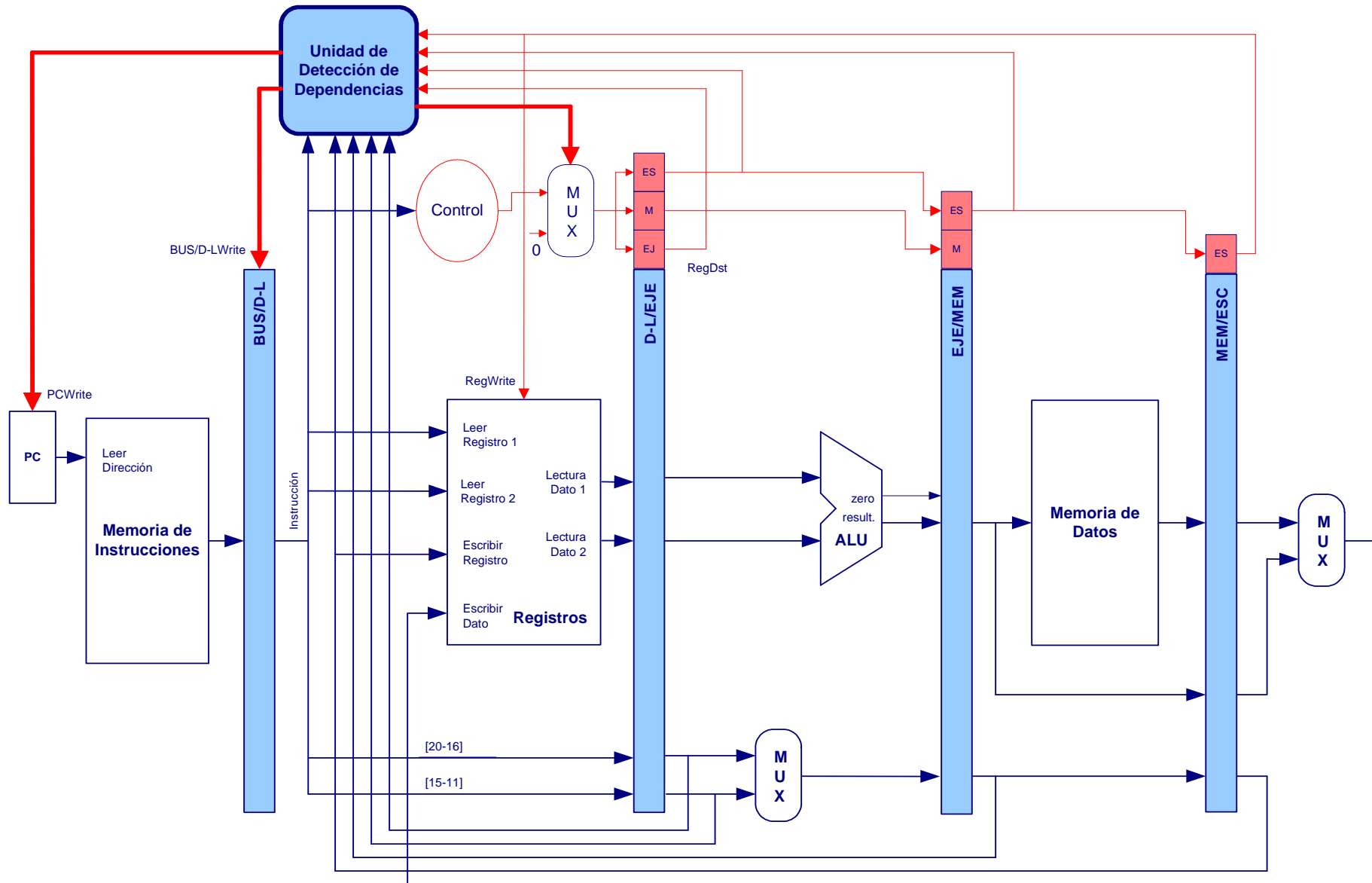


Detección de Dependencias





Control de Dependencias





Control de Dependencias

I1	BUS ₁	D-L ₁	EJE ₁	MEM ₁	ESC ₁				
		BUS ₂	D-L ₂	-	-	-			
			BUS ₂	D-L ₂	-	-	-		
				BUS ₂	D-L ₂	-	-	-	
I2					BUS ₂	D-L ₂	EJE ₂	MEM ₂	ESC ₂



Resolución mediante Hardware (II)

★ División del ciclo de Escritura/Lectura

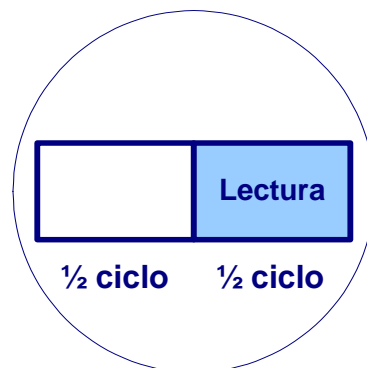
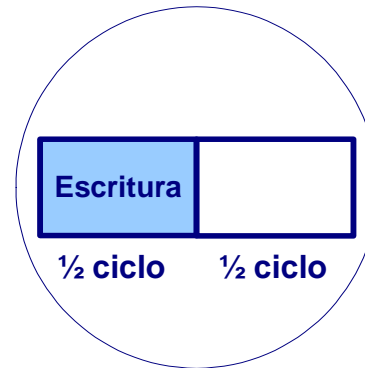
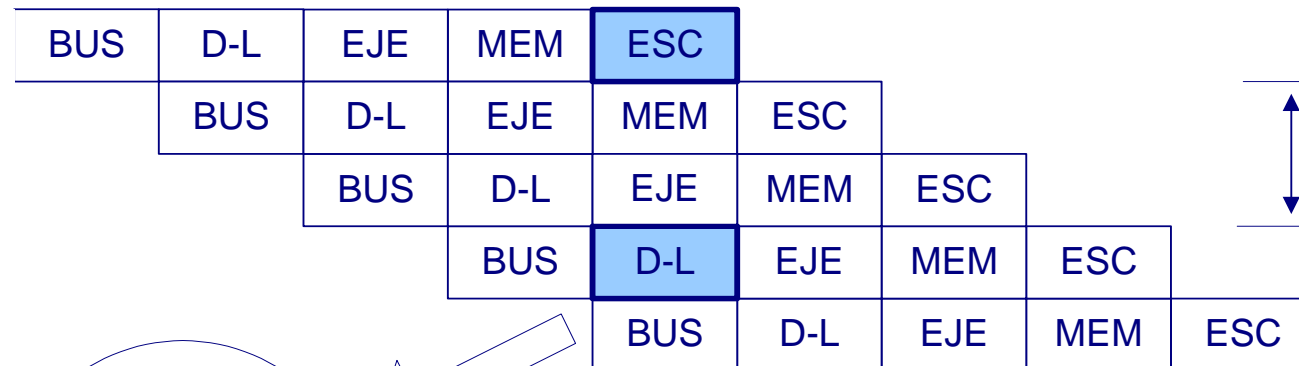
ADD R1, R2, R3

?

?

OR R8, R1, R9

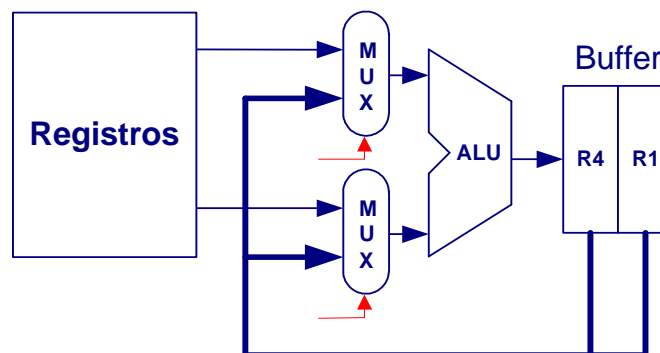
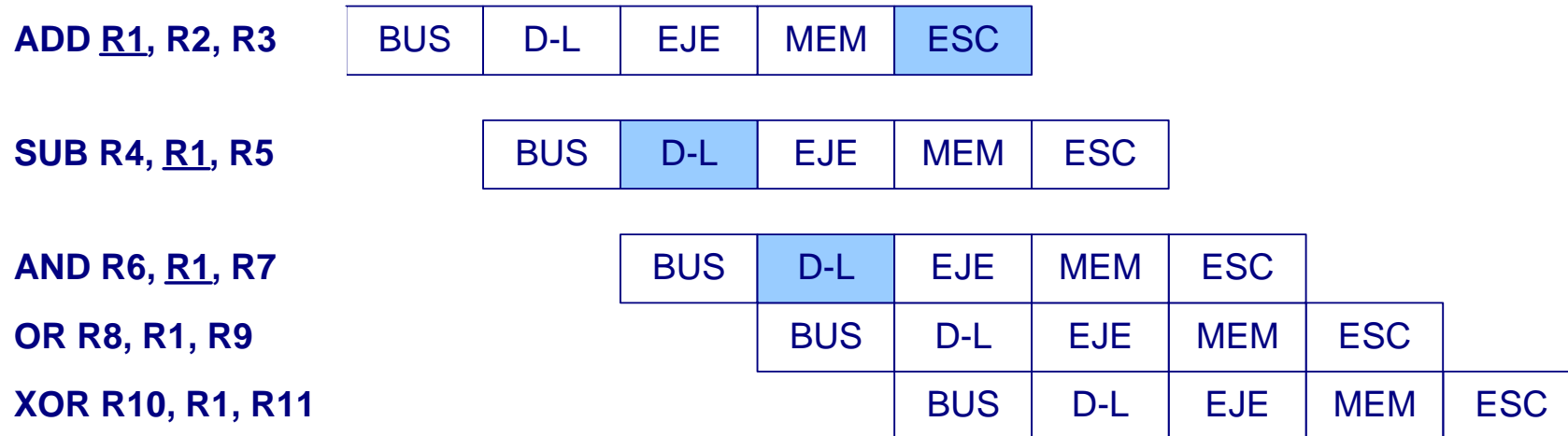
XOR R10, R1, R11





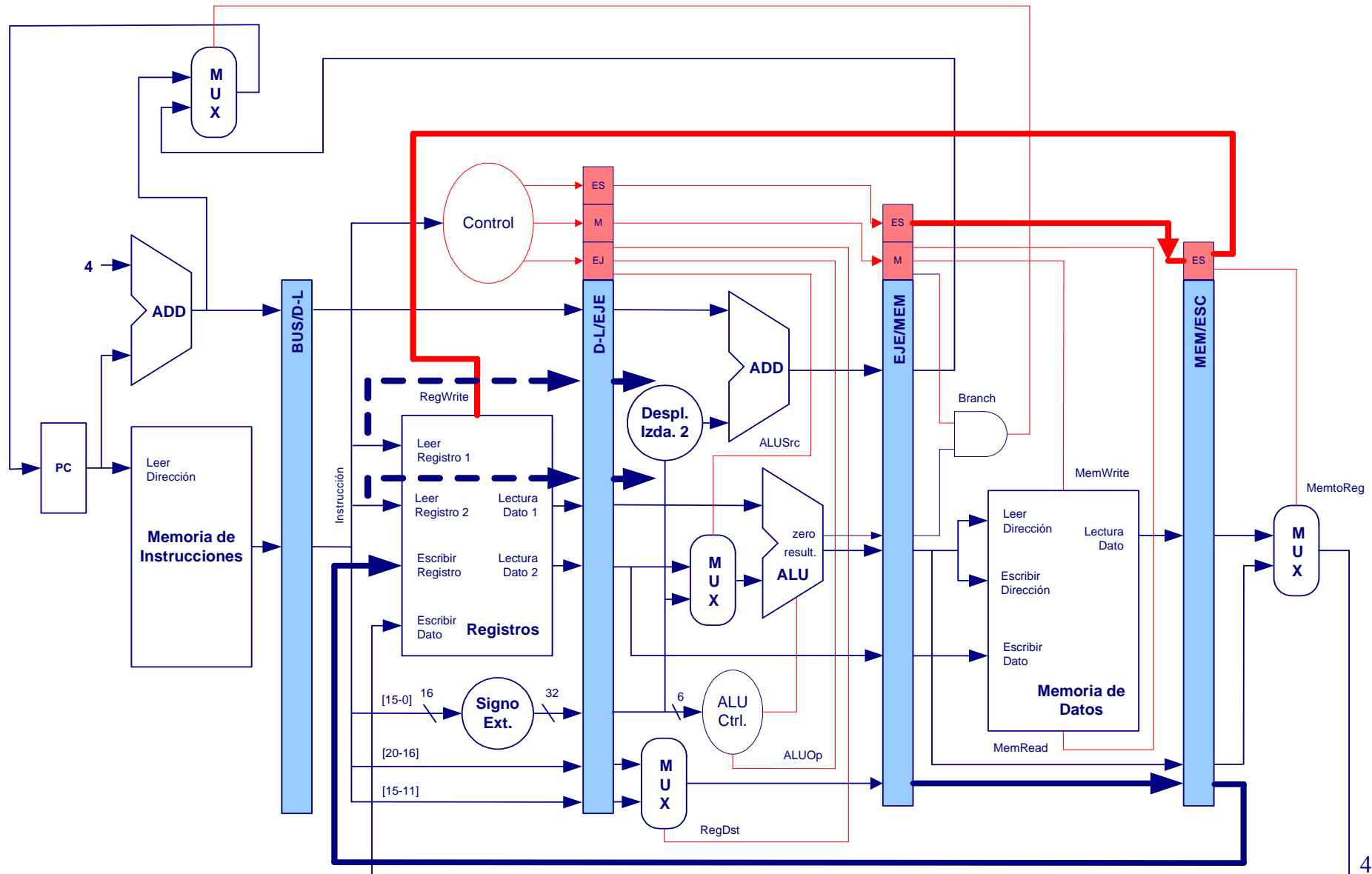
Resolución mediante Hardware (III)

★ Técnica de Adelantamiento o Desvío (*Forwarding, Bypassing*)



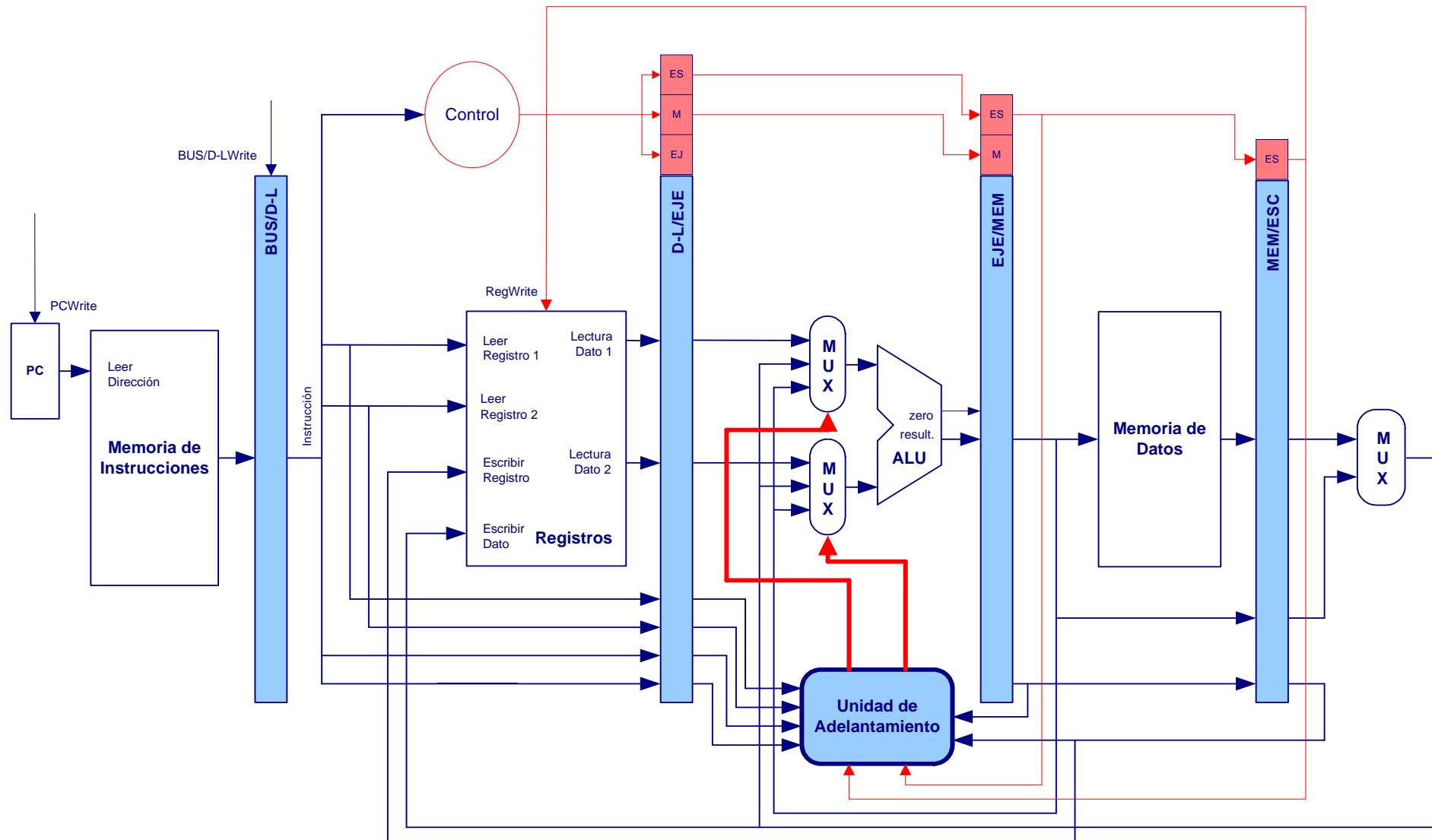


Viabilidad del Adelantamiento





Control del Adelantamiento





Instrucciones de Carga (I)

Efecto de una instrucción de carga en la segmentación:

LW R1, 32(R6)

BUS	D-L	EJE	MEM	ESC
-----	-----	-----	-----	-----

ADD R4, R1, R7

BUS	D-L	detención	EJE	MEM	ESC
-----	-----	-----------	-----	-----	-----

SUB R5, R1, R8

BUS	detención	D-L	EJE	MEM	ESC
-----	-----------	-----	-----	-----	-----

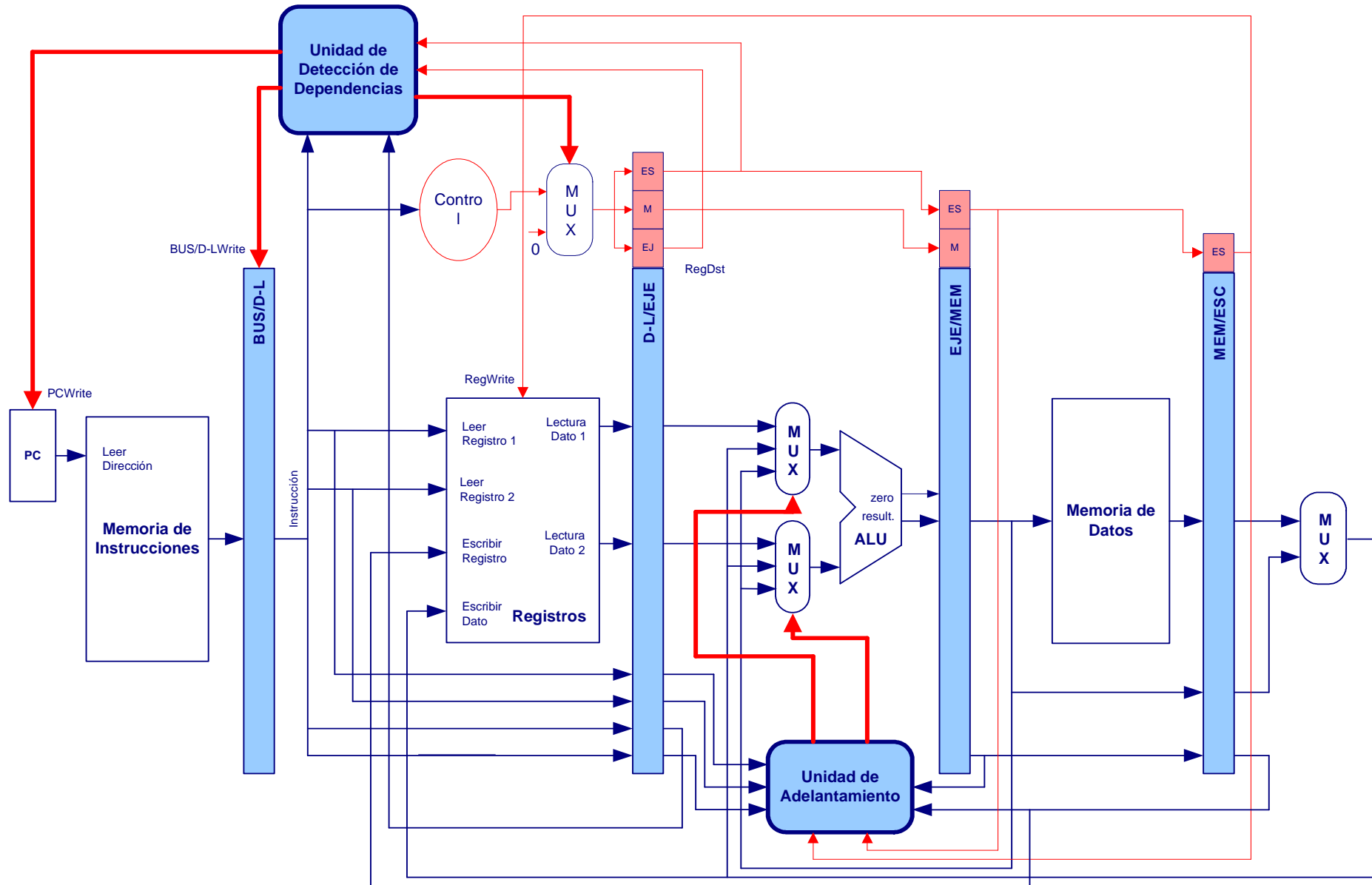
AND R6, R1, R7

detención	BUS	D-L	EJE	MEM	ESC
-----------	-----	-----	-----	-----	-----





Control de Dependencias/Adelantamiento





Instrucciones de Carga (II)

Secuencia de código para la operación $A=B+C$:

LW R1, B



LW R2, C



ADD R3, R1, R2



SW A, R3





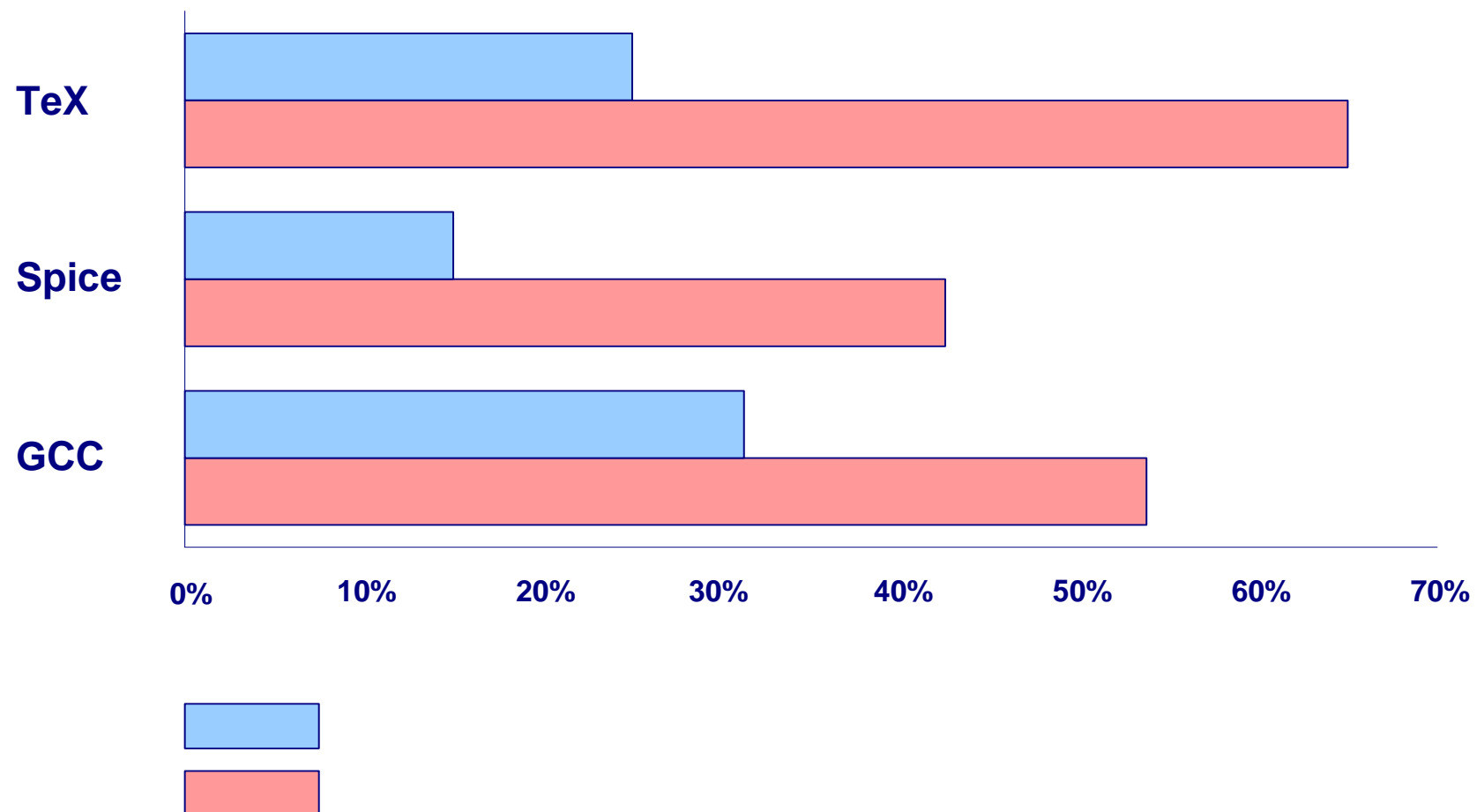
Instrucciones de Carga (II)

- Planificación de la Segmentación:
✓
- Carga retardada:
✓
- Retardo de carga / Hueco de retardo de carga:
✓



Instrucciones de Carga (III)

Porcentaje de cargas que causan detención con la segmentación:





Instrucciones de Carga (IV)

No Dependencia

LW R1, 45(R2)
ADD R5, R6, R7
SUB R8, R6, R7
OR R9, R6, R7

No hay problema, porque no existe dependencia de R1 en las tres siguientes instrucciones

Dependencia que requiere detención

LW R1, 45(R2)
ADD R5, R1, R7
SUB R8, R6, R7
OR R9, R6, R7

El Hardware detecta el uso de R1 en la instrucción ADD y detiene tanto a ella como a las siguientes antes de que comience la etapa EJE de ADD

Dependencia superada por adelantamiento

LW R1, 45(R2)
ADD R5, R6, R7
SUB R8, R1, R7
OR R9, R6, R7

El Hardware detecta el uso de R1 en la instrucción SUB y adelanta el resultado de la carga a la ALU en el instante en el que comienza la etapa EJE de SUB

Dependencia con accesos ordenados

LW R1, 45(R2)
ADD R5, R6, R7
SUB R8, R6, R7
OR R9, R1, R7

No se requiere ninguna acción, porque la lectura de R1 por parte de la instrucción OR tiene lugar en la segunda mitad del ciclo en la etapa D-L, mientras que la escritura de dato cargado en R1 tiene lugar en la primera mitad del ciclo

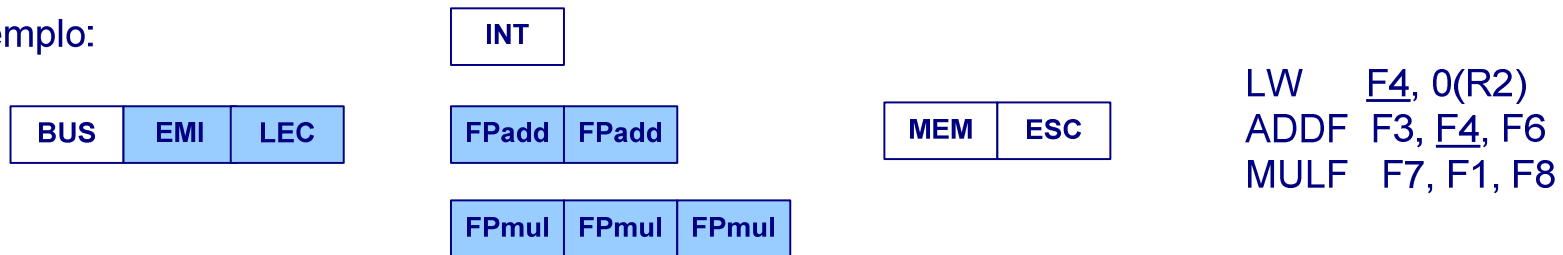


Resolución mediante Hardware (IV)

★ Ejecución fuera de orden con Marcador (Scoreboarding)

- solo se detienen las instrucciones que dependen, y no las posteriores
- la etapa D-L se divide en Emisión (EMI) y Lectura de Registros (LEC)
- necesidad de múltiples unidades funcionales de ejecución

Ejemplo:



- las instrucciones pueden emitirse y/o completarse fuera de orden
- aparecen problemas de dependencias EDL y EDE
- el marcador lleva cuenta del estado de todas las instrucciones, unidades funcionales y registros, y controla la emisión y detención de instrucciones

