

Arquitectura de Computadores

1. Introducción
2. La CPU
3. Lenguaje Máquina
4. La Memoria
5. Sistemas de Entrada/Salida
6. Segmentación (*Pipeline*)
 - MIPS 64
7. Memoria Caché
8. Arquitecturas RISC

- Bus de datos de 64 bits
- Registros de 64 bits
 - 32 registros de propósito general (R0..R31)
 - 32 registros de coma flotante (F0..F32)
- Instrucciones de longitud fija (32 bits)
- Código de operación de longitud fija (6 bits)
- Arquitectura Load/Store
- Memoria separada para instrucciones y datos (Harvard)
- Cauce segmentado en 5 etapas de 1 ciclo cada una



Etapa IF

- Extracción de la instrucción
- $PC = PC + 4$

Etapa ID

- Decodificación de la instrucción
- Lectura de los registros
- Extensión de signo (si es necesario)
- Actualización del PC en caso de salto

Etapas EX

- Cálculos en la ALU
- Cálculo de dirección de operandos de memoria

Etapas MEM

- Acceso a memoria (Load/Store)

Etapas WB

- Escritura de los registros



Problema: Se accede a la memoria a la vez
Solución: Memoria separada (Harvard)



Problema: Se escriben y leen los registros a la vez
Solución: Escritura en el 1º subciclo y lectura en el 2º

RAR (Read After Read)

No presenta problemas

DADD R1,R2,R3

AND R4,R5,R3



RAW (Read After Write)

¡Problemas!

DADD R1,R2,R3

AND R4,R5,R1



WAR (Write After Read)

Se da cuando hay ejecución fuera de orden

DADD R1,R2,R3

AND R3,R4,R5

WAW (Write After Write)

Se da cuando hay ejecución fuera de orden u operaciones multiciclo

DIV.D F1,F2,F3

ADD.D F1,F4,F5



¿Dónde se detectan
los riesgos?

En la fase ID

Solución de
riesgos RAW

- Software

- Inserción de instrucciones NOP

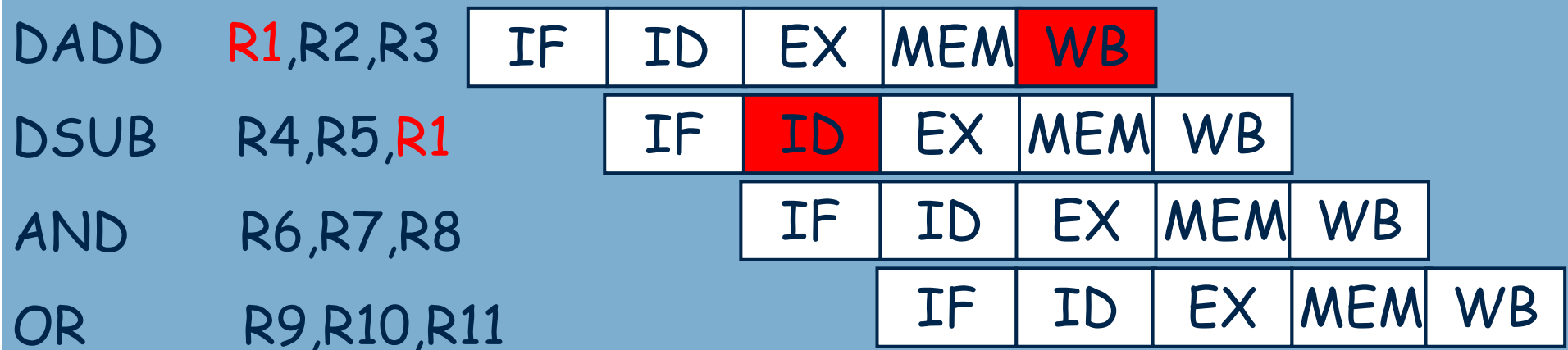
- Reordenación de código

- Hardware

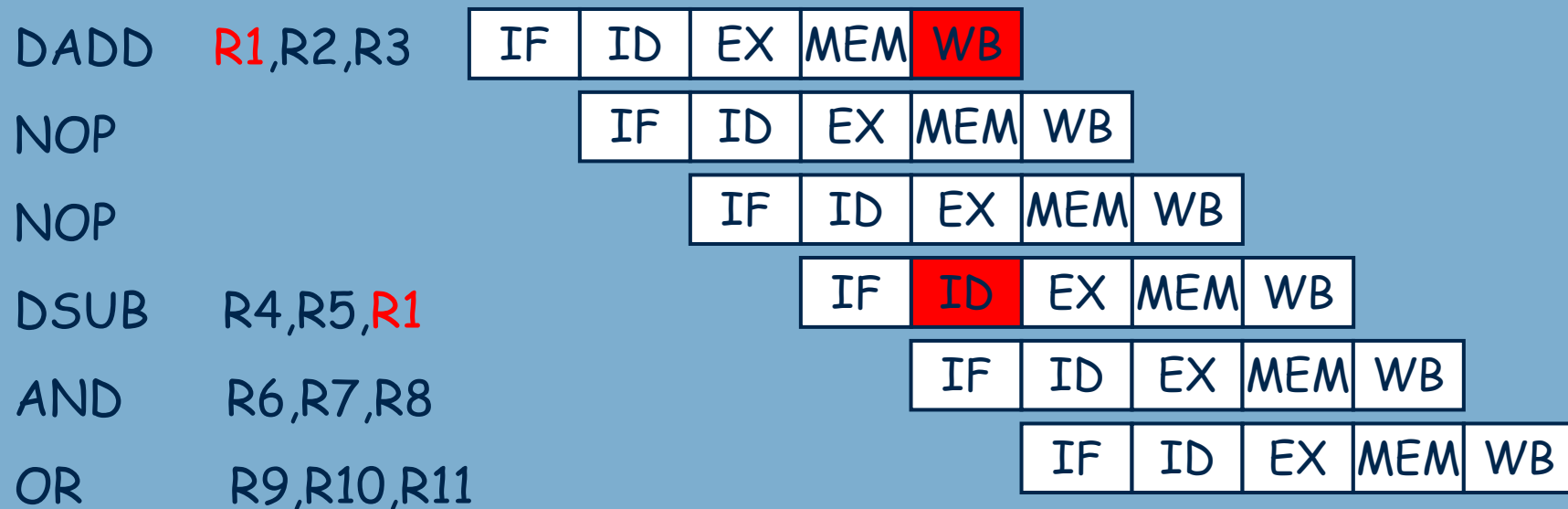
- Detención del cauce

- Anticipación

El problema ...

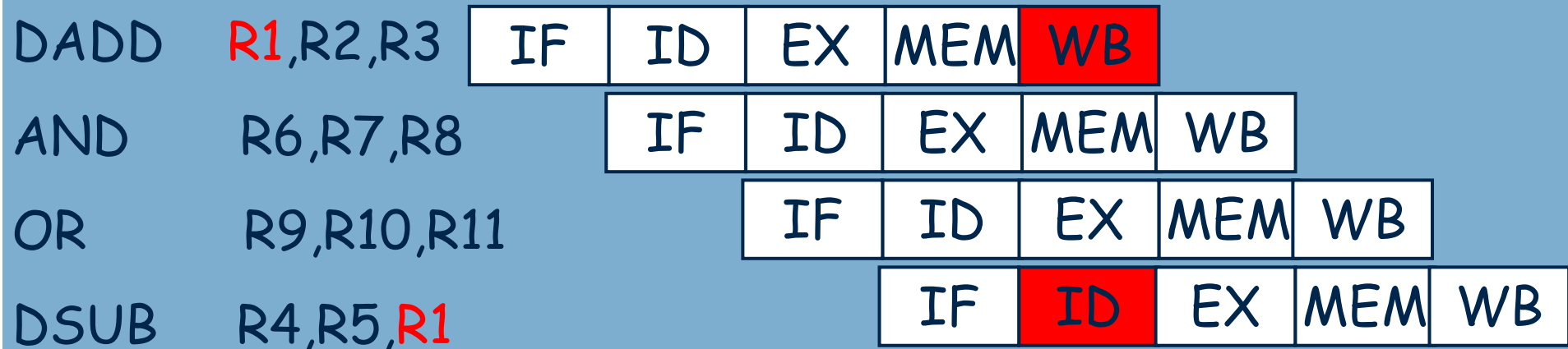


Inserción de instrucciones NOP (Solución software)



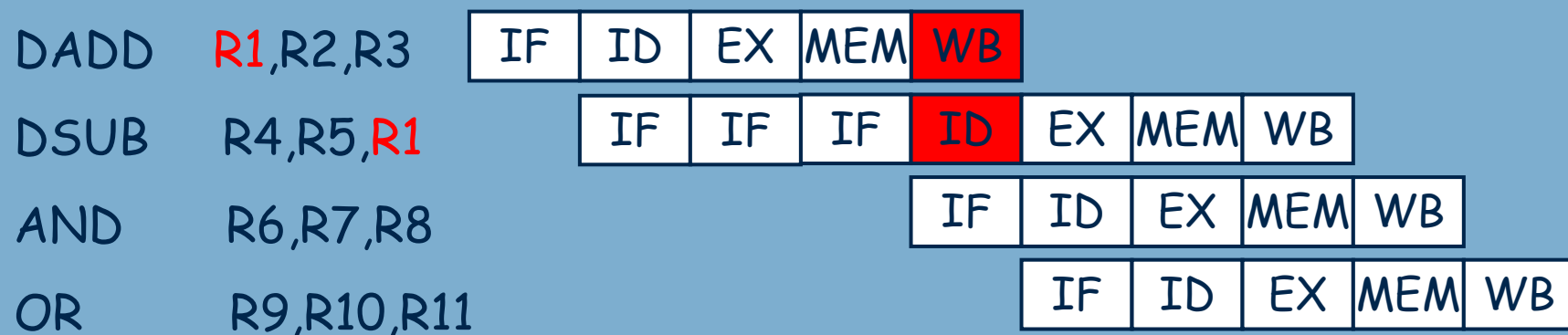
Se alarga el tiempo de ejecución

Reordenación del código (Solución software)



Se mantiene el tiempo de ejecución

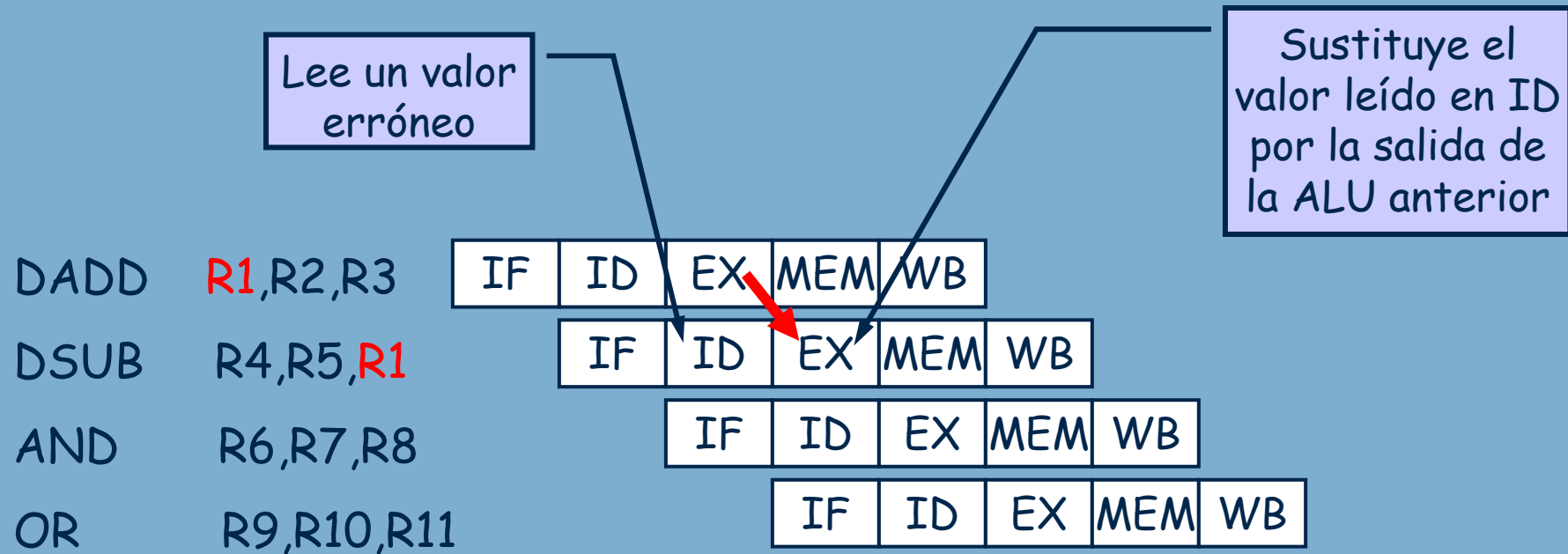
Detención del cauce (Solución Hardware)



Se alarga el tiempo de ejecución igual que insertando NOP

Anticipación (forwarding) (Solución hardware)

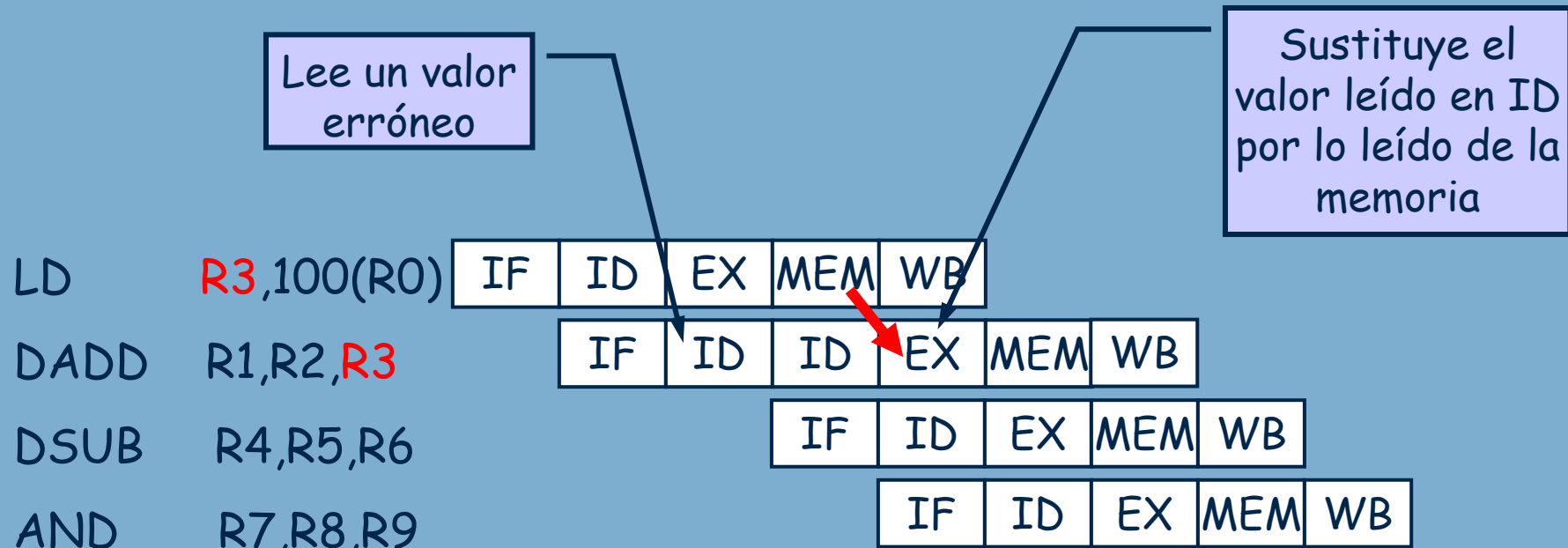
Caso 1:



Se mantiene el tiempo de ejecución

Anticipación (forwarding) (Solución hardware)

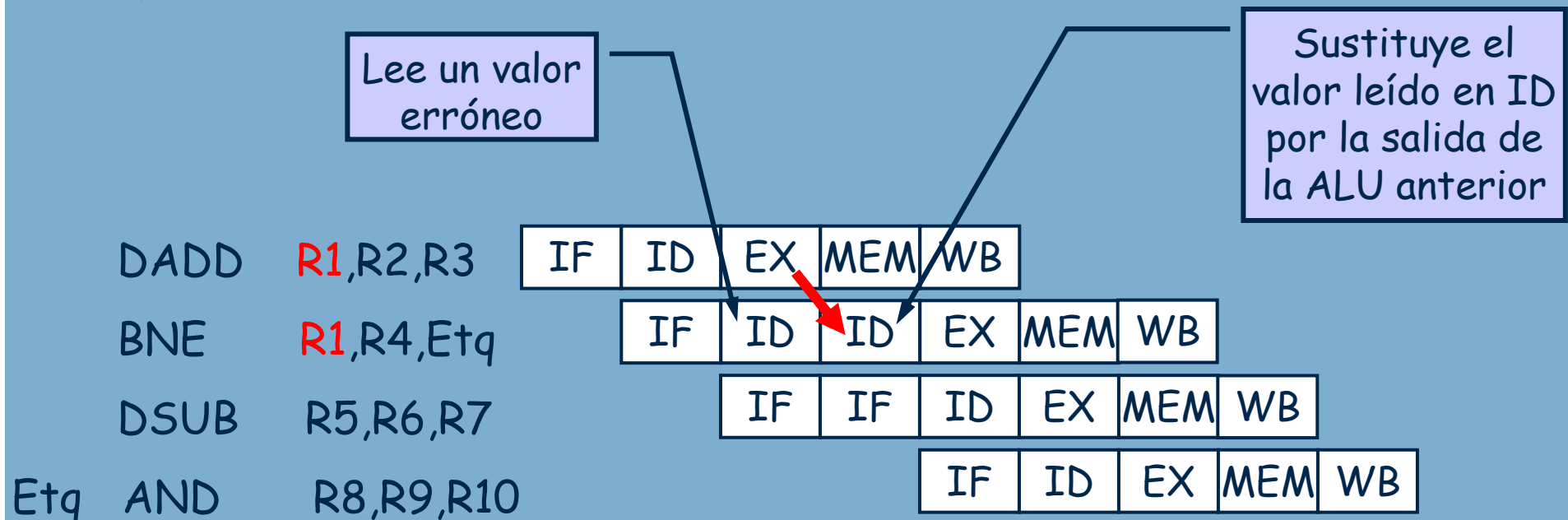
Caso 2:



Se incrementa 1 ciclo el tiempo de ejecución

Anticipación (forwarding) (Solución hardware)

Caso 3:

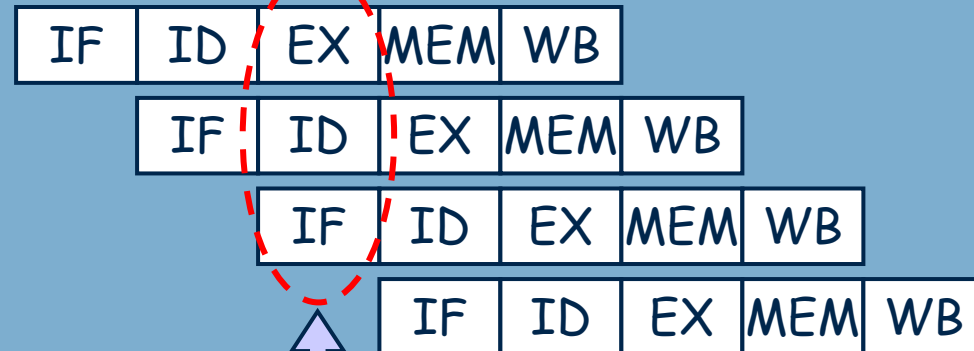


Si salta, se abortará la ejecución de "DSUB"

El problema ...

ADD R1,R2,R3
BEQ R4,R5,Etq
AND R6,R7,R8
Etq OR R9,R10,R11

Aquí no se sabe si hay
que ejecutar AND ...



... pero ya se ha cargado

¡AND siempre se ejecuta!

Solución de riesgos de control

- Software

- Inserción de instrucciones NOP

- Salto retardado

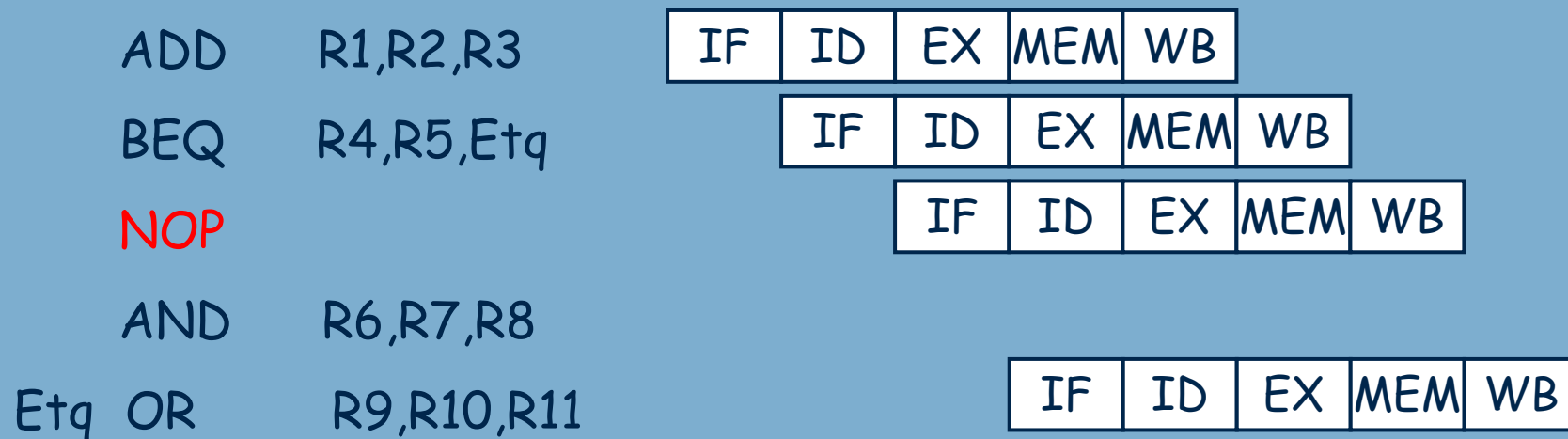
- Hardware

- Detención del cauce

- Predicción del salto

Inserción de instrucciones NOP (Solución software)

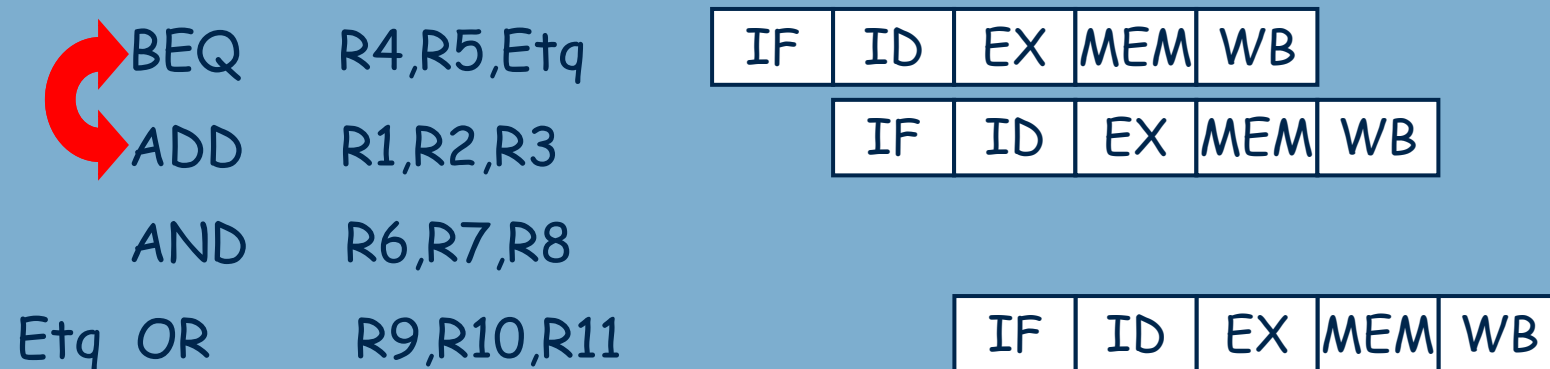
En caso de salto ...



... AND no se ejecuta

Salto retardado (Solución software)

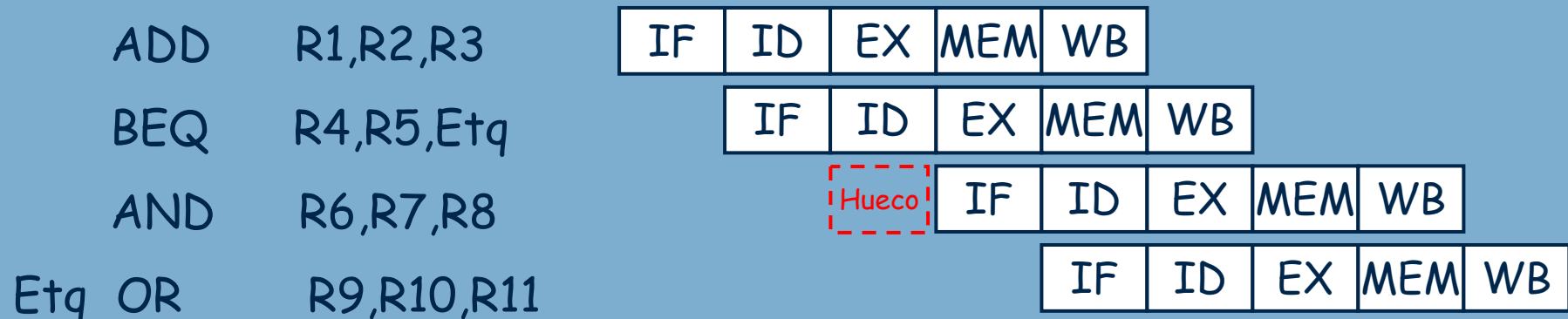
En caso de salto ...



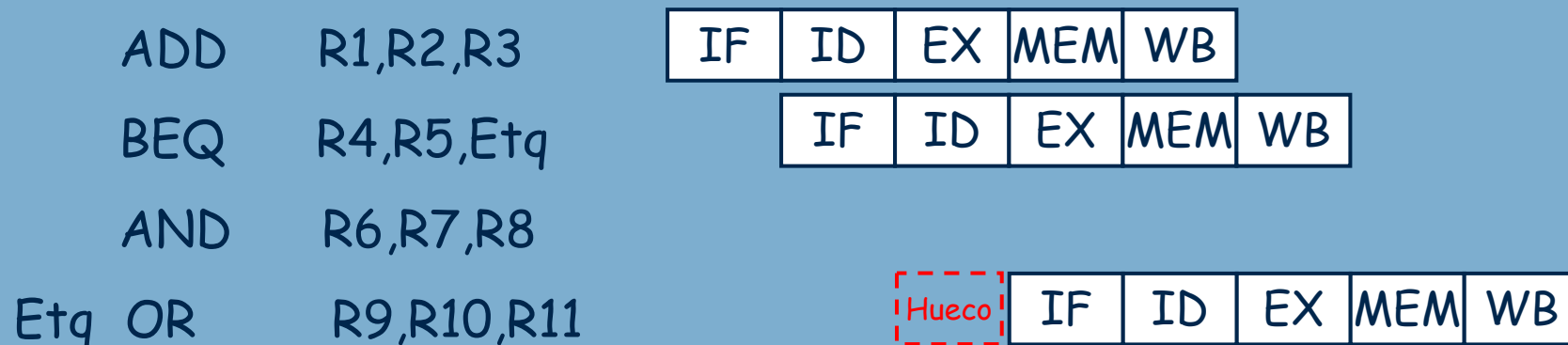
... AND no se ejecuta

Detención del cauce (Solución Hardware)

En caso de no salto ...



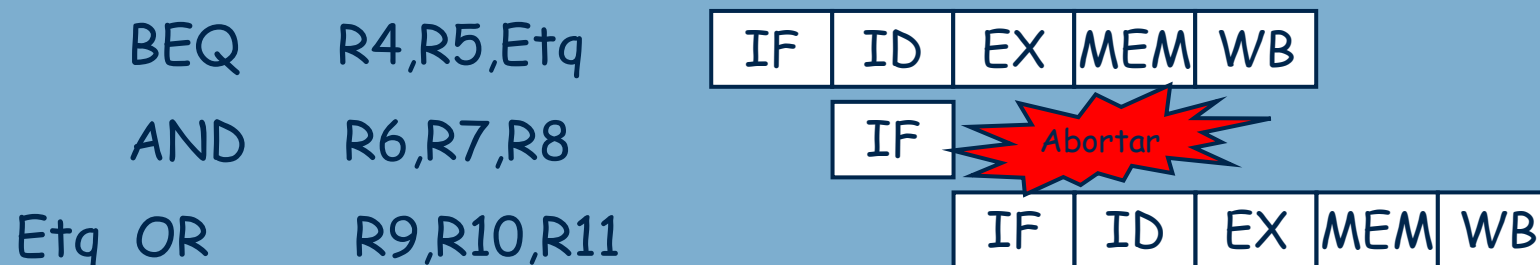
En caso de salto ...



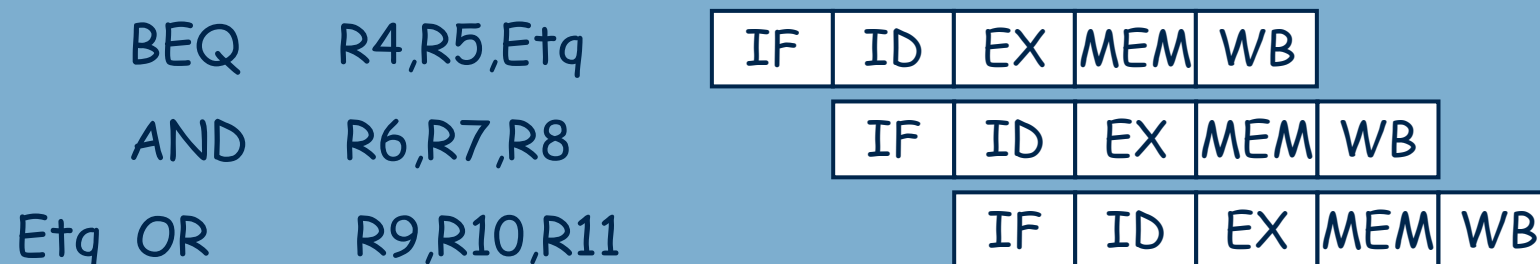
Predicción del salto (Solución Hardware)

Se predice **no** saltar

Si falla → penalización = 1 ciclo



Si acierta → penalización = 0



Predicción del salto (Solución Hardware)

Se predice **sí** saltar

No se utiliza en MIPS 64 ya que la penalización siempre es de 1 ciclo

BEQ R4,R5,Etq
AND R6,R7,R8
Etq OR R9,R10,R11



Hasta este instante no se conoce la dirección del salto

?

?