Organización de Computadoras 2011

Pipeline

José Luis Hamkalo
Depto. de Electrónica

Organización de la Clase

- Introducción al procesamiento pipeline.
- Arquitectura MIPS 32 con implementación pipeline.
- Riesgos pipeline.
  - Estructurales
  - Datos
  - Control

José Luis Hamkalo
Depto. de Electrónica

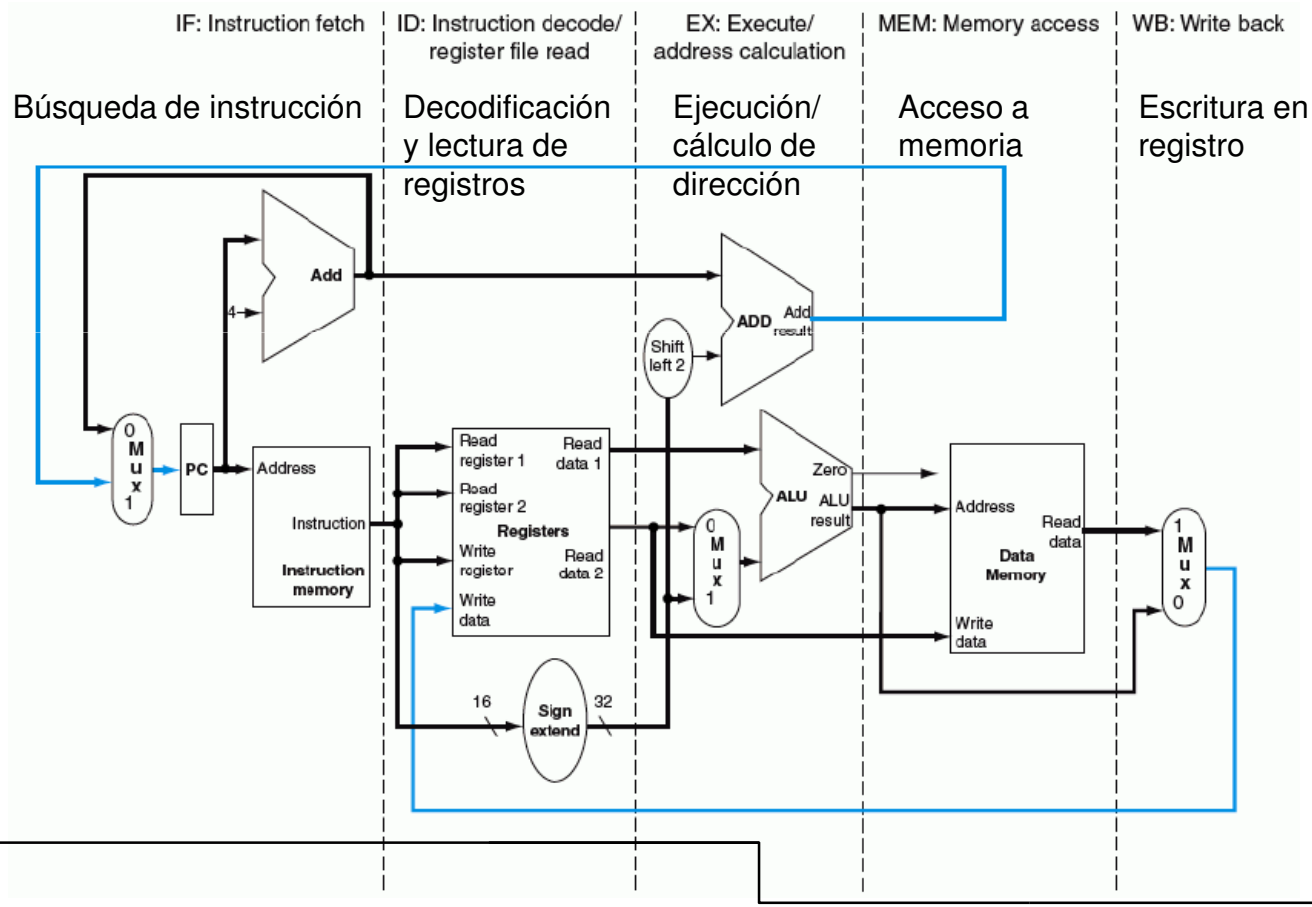
Procesamiento Pipeline. La Idea de una Línea de Montaje.

- Ejemplo de fabricación de un automóvil.
- Se requieren cuatro pasos:
  - Construcción de la carrocería.
  - Protección anticorrosiva y pintura.
  - Ensamblado mecánico: caja-motor.
  - Colocación de interiores y terminación.

Procesamiento Pipeline. La Idea de una Línea de Montaje (cont).

- Hipótesis: cada paso insume una hora.
- Esquema de producción tradicional:
  - Un auto a la vez.
  - Se produce un auto cada cuatro horas.
- Esquema de línea de montaje:
  - Se fabrican cuatro autos a la vez.
  - En distintos grados de avance.
  - Se inicia y termina un auto por hora.

Camino de Datos MIPS 32 Simplificado. Implementación Monociclo.



Reloj

José Luis Hamkalo
Depto. de Electrónica

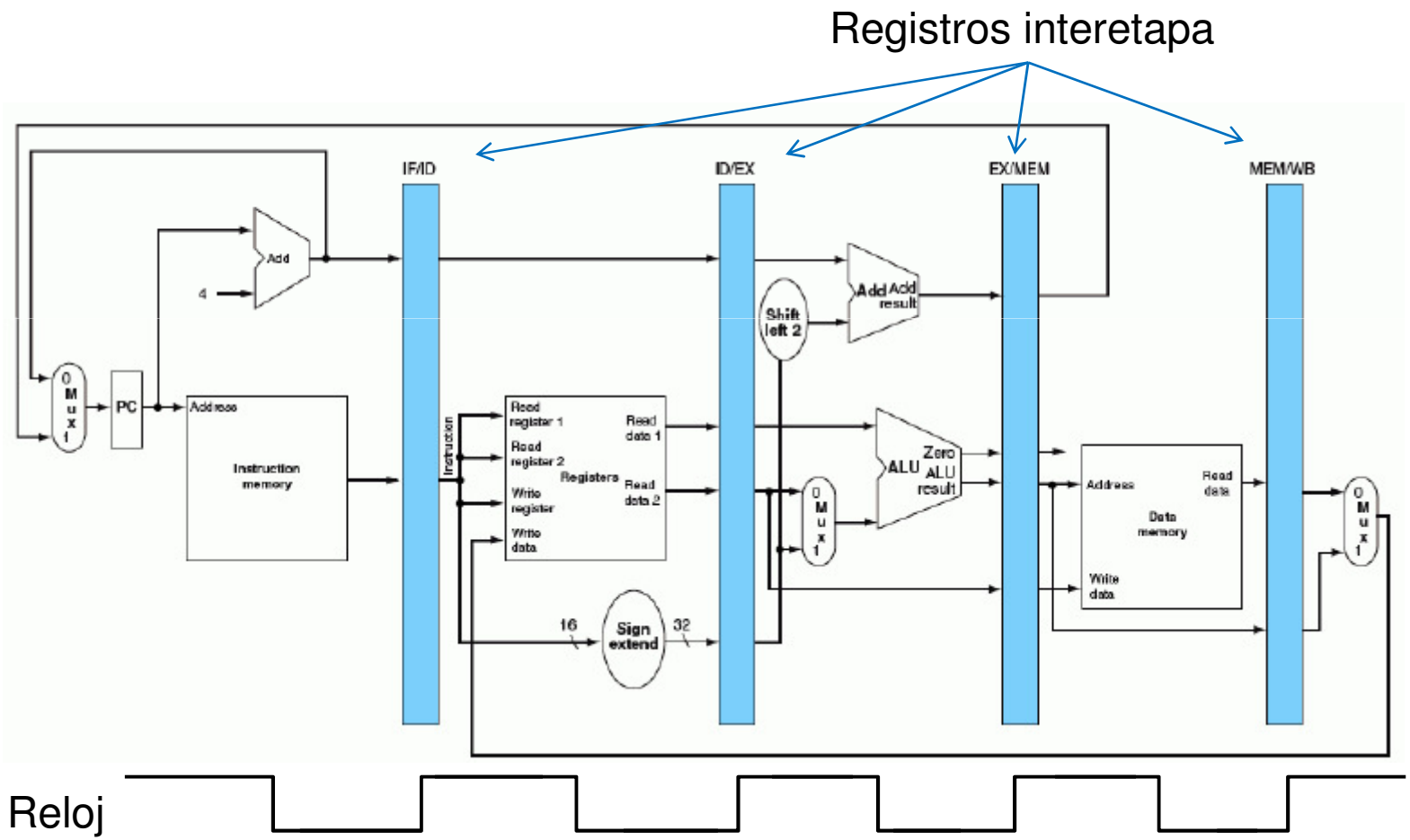
Dividir el Trabajo en Etapas Separadas por Registros.

- Busqueda de Instrucciones (IF)
- Decodificación y búsqueda de operandos (ID)
- Ejecución y cálculo de dirección (EX)
- Acceso a memoria (MEM)
- Write back (WB)

Procesamiento pipeline: procesar como en una línea de montaje.

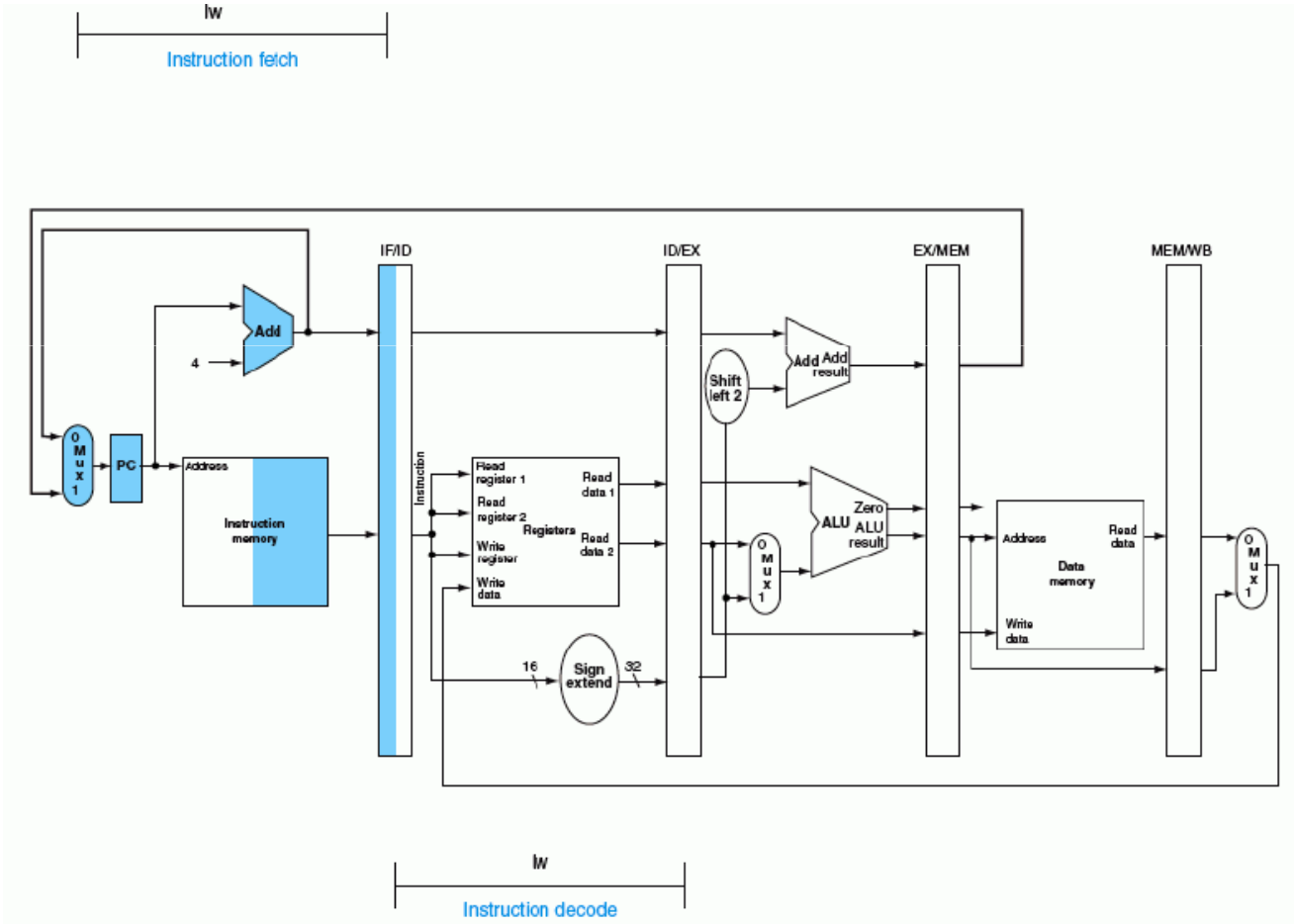
José Luis Hamkalo
Depto. de Electrónica

Versión Pipeline del Camino de Datos MIPS



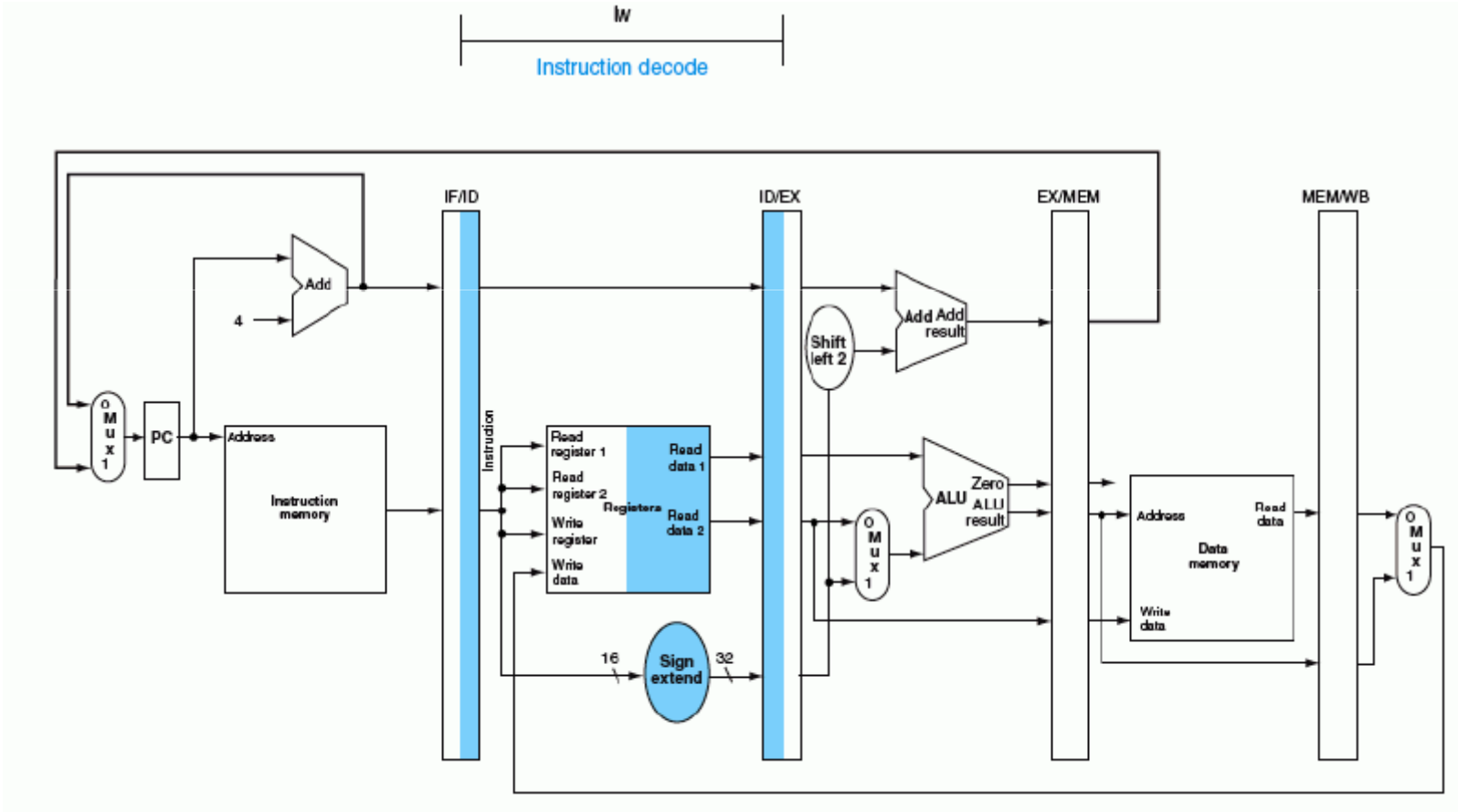
José Luis Hamkalo
Depto. de Electrónica

Etapa IF de una instrucción lw



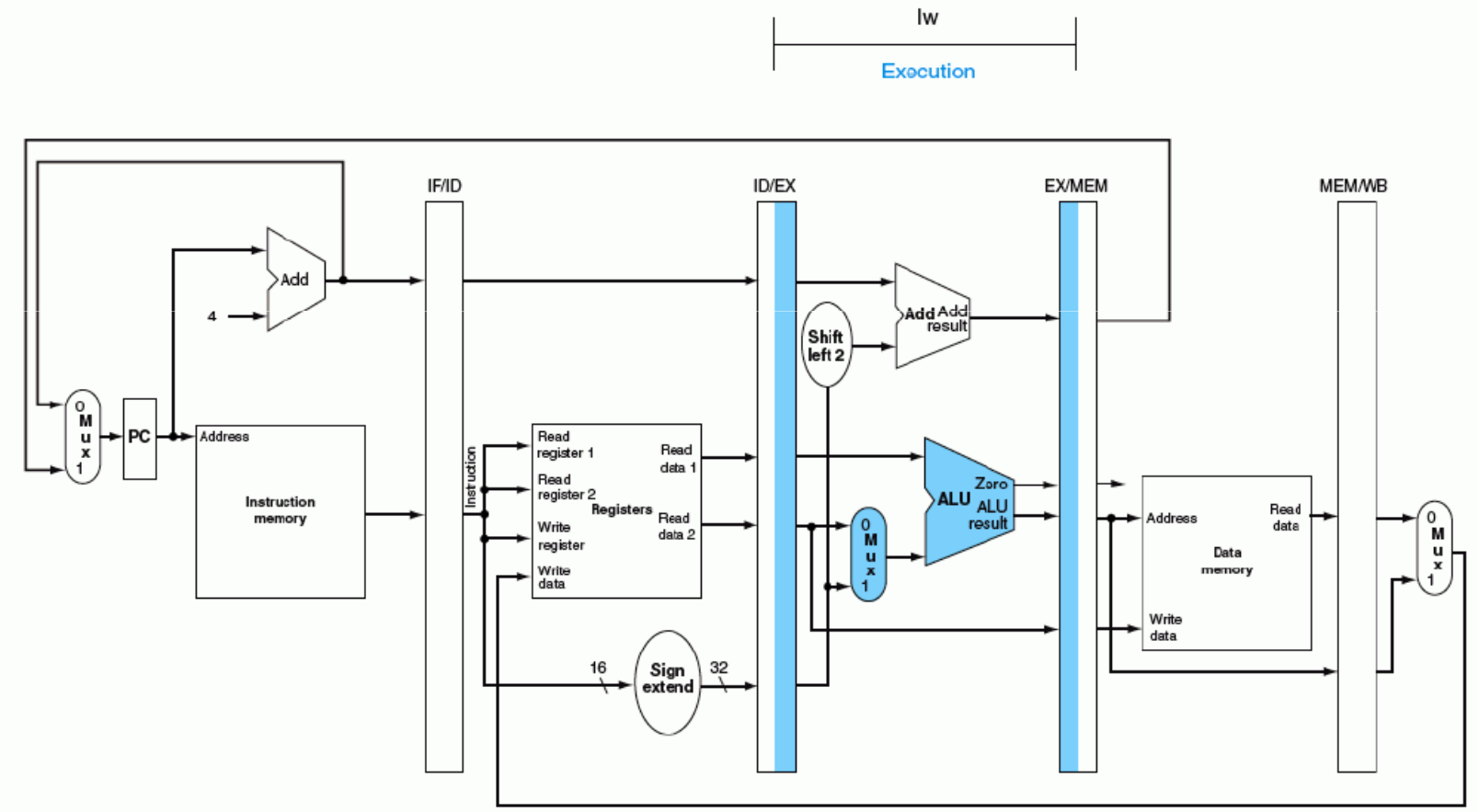
José Luis Hamkalo
Depto. de Electrónica

Etapa ID de una instrucción lw



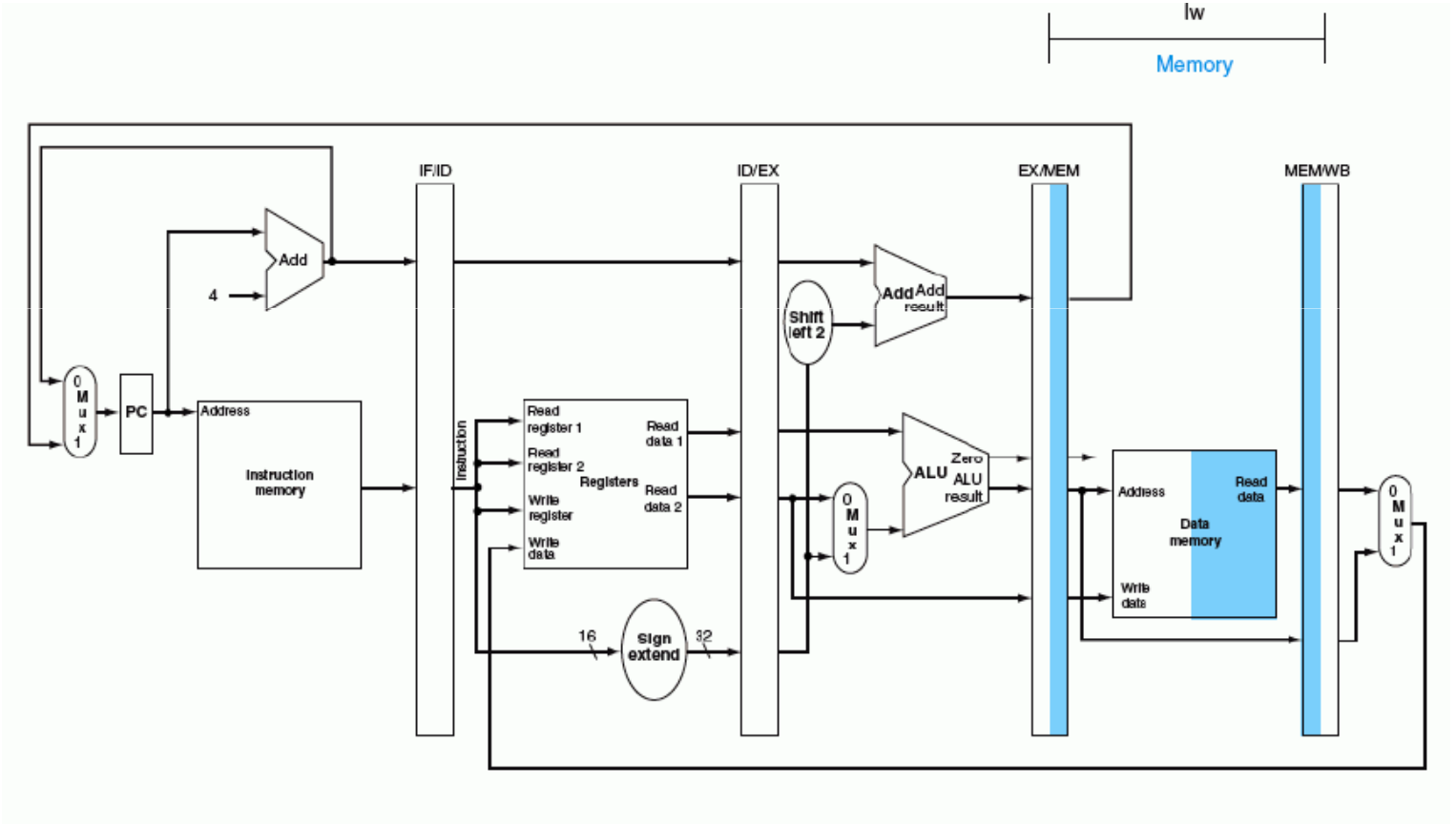
José Luis Hamkalo
Depto. de Electrónica

Etapa EX de una instrucción lw



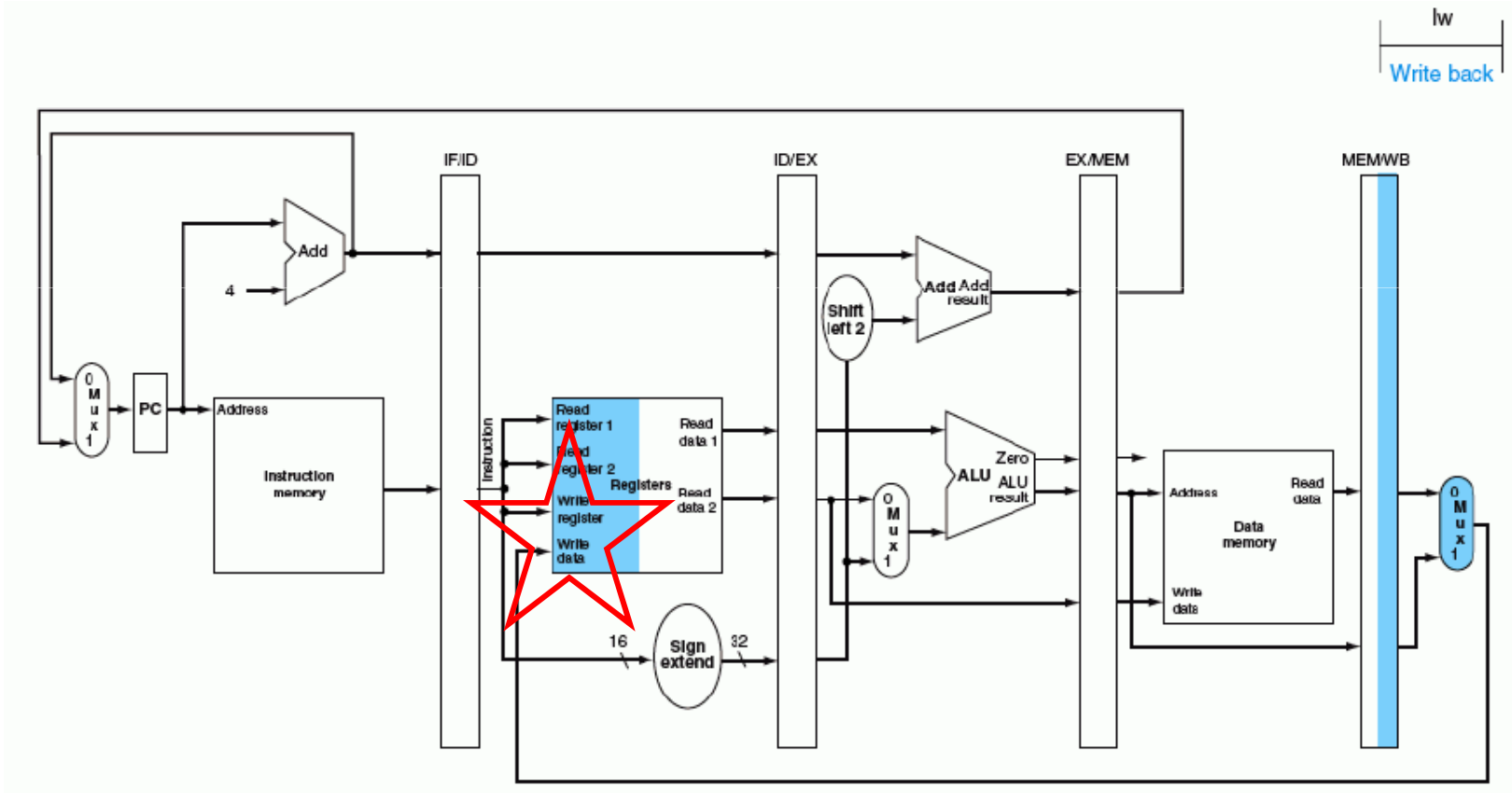
José Luis Hamkalo
Depto. de Electrónica

Etapa MEM de una instrucción lw



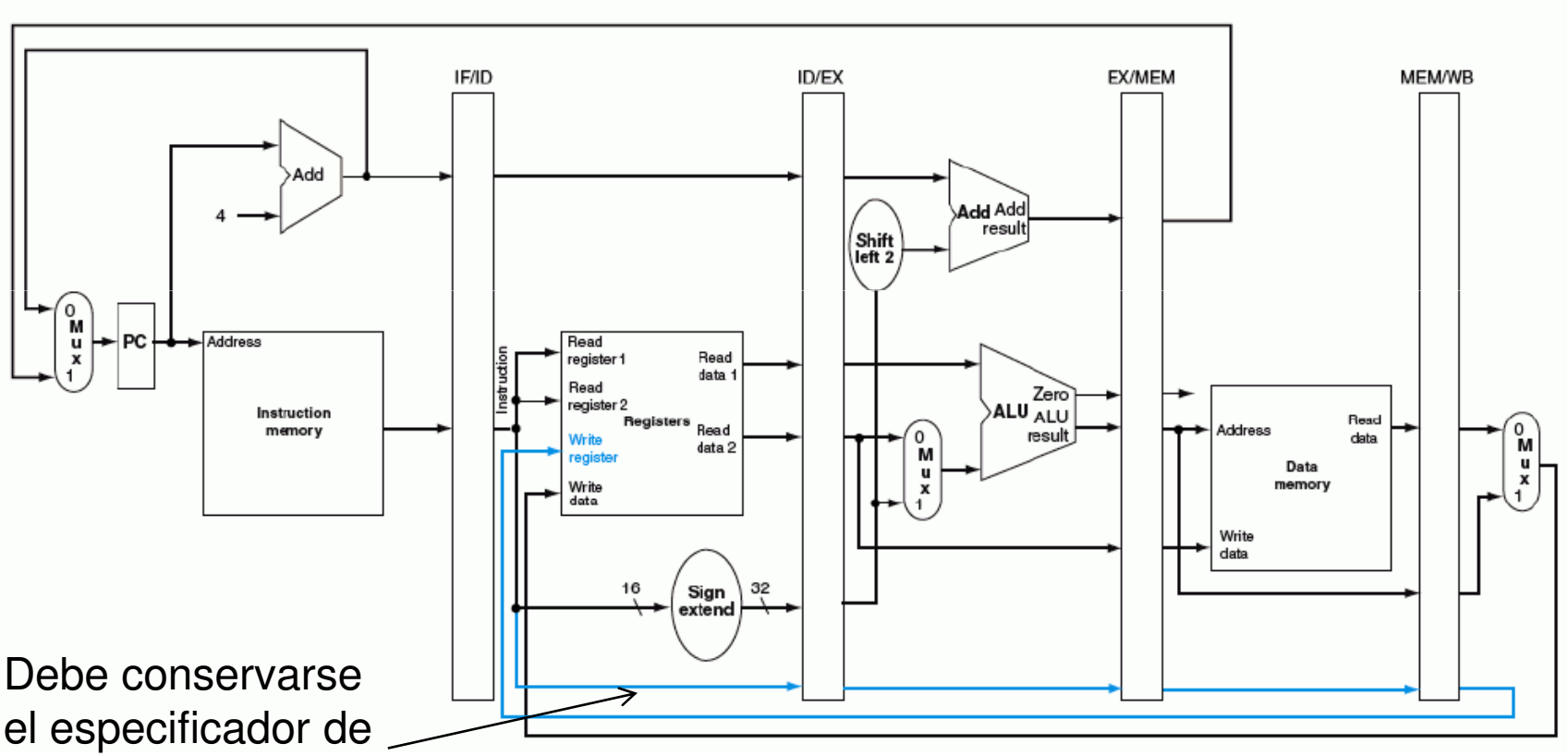
José Luis Hamkalo
Depto. de Electrónica

Etapa WB de una instrucción lw



José Luis Hamkalo
Depto. de Electrónica

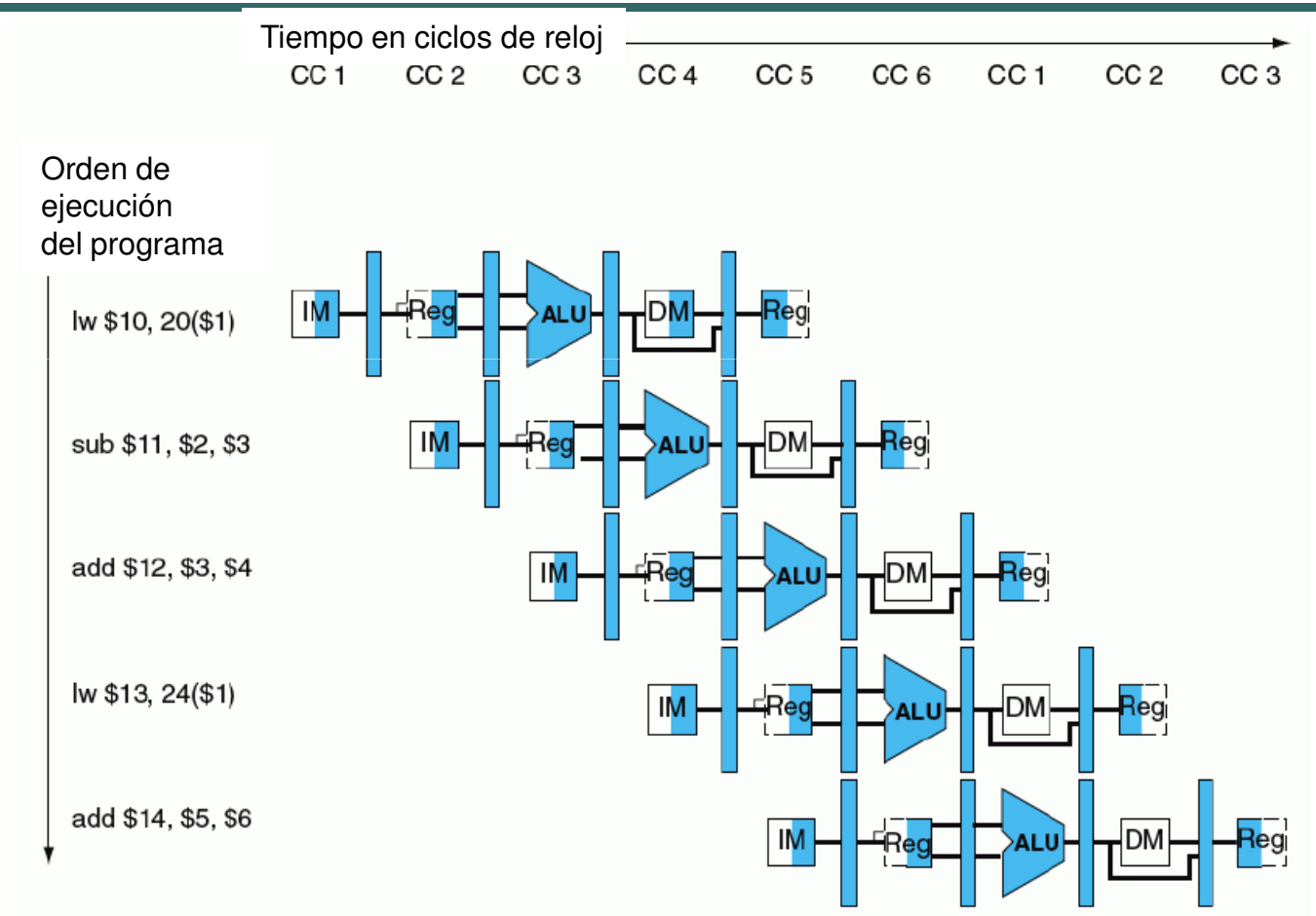
Corrección para la instrucción lw



Debe conservarse el especificador de registro destino

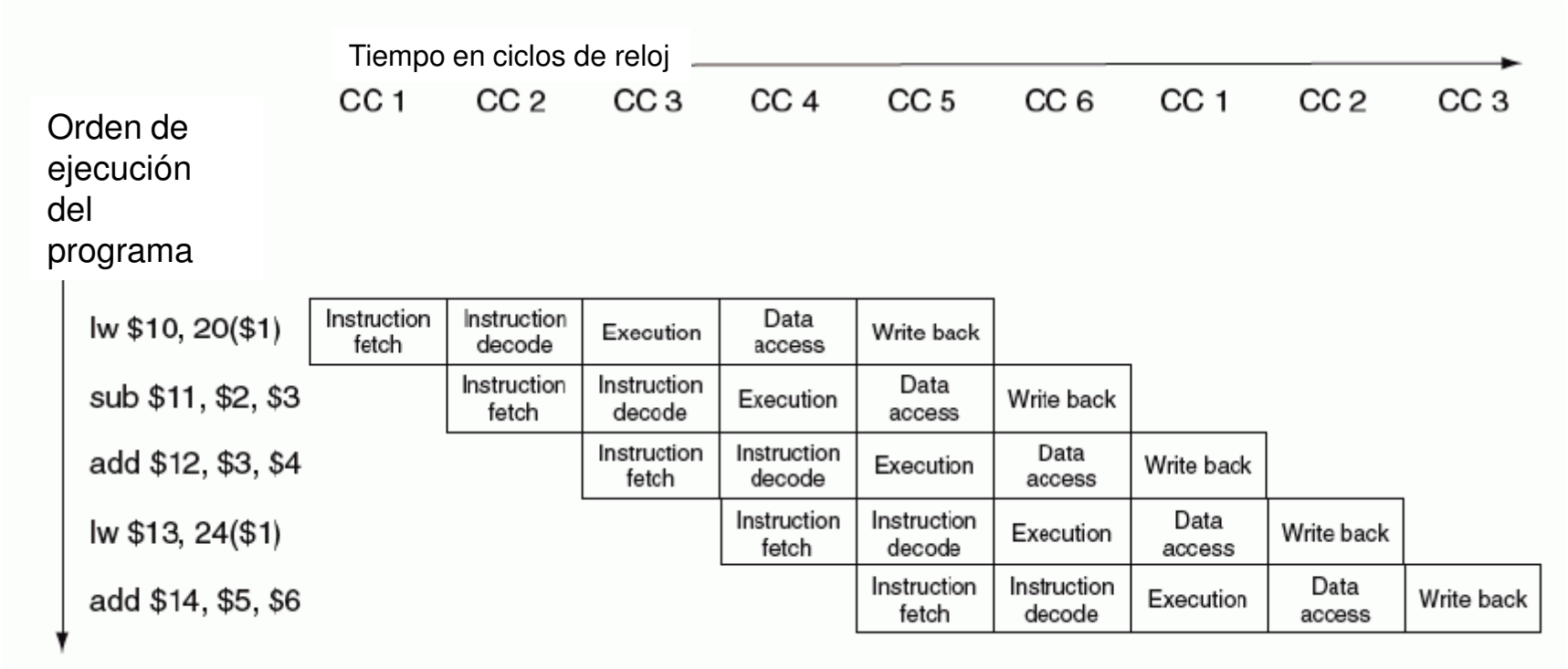
José Luis Hamkalo
Depto. de Electrónica

Una Secuencia de Instrucciones



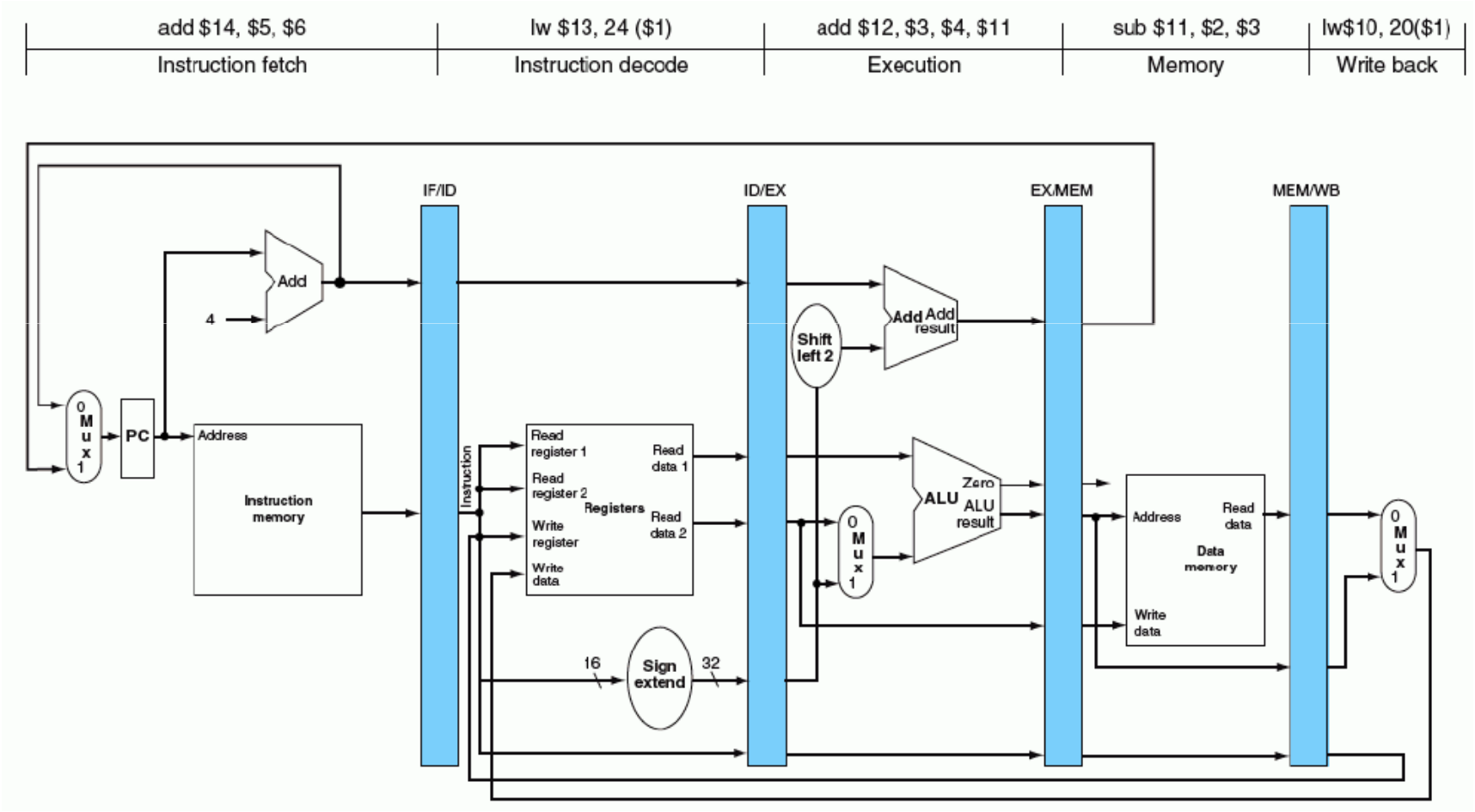
José Luis Hamkalo
Depto. de Electrónica

Diagrama Pipeline Tradicional



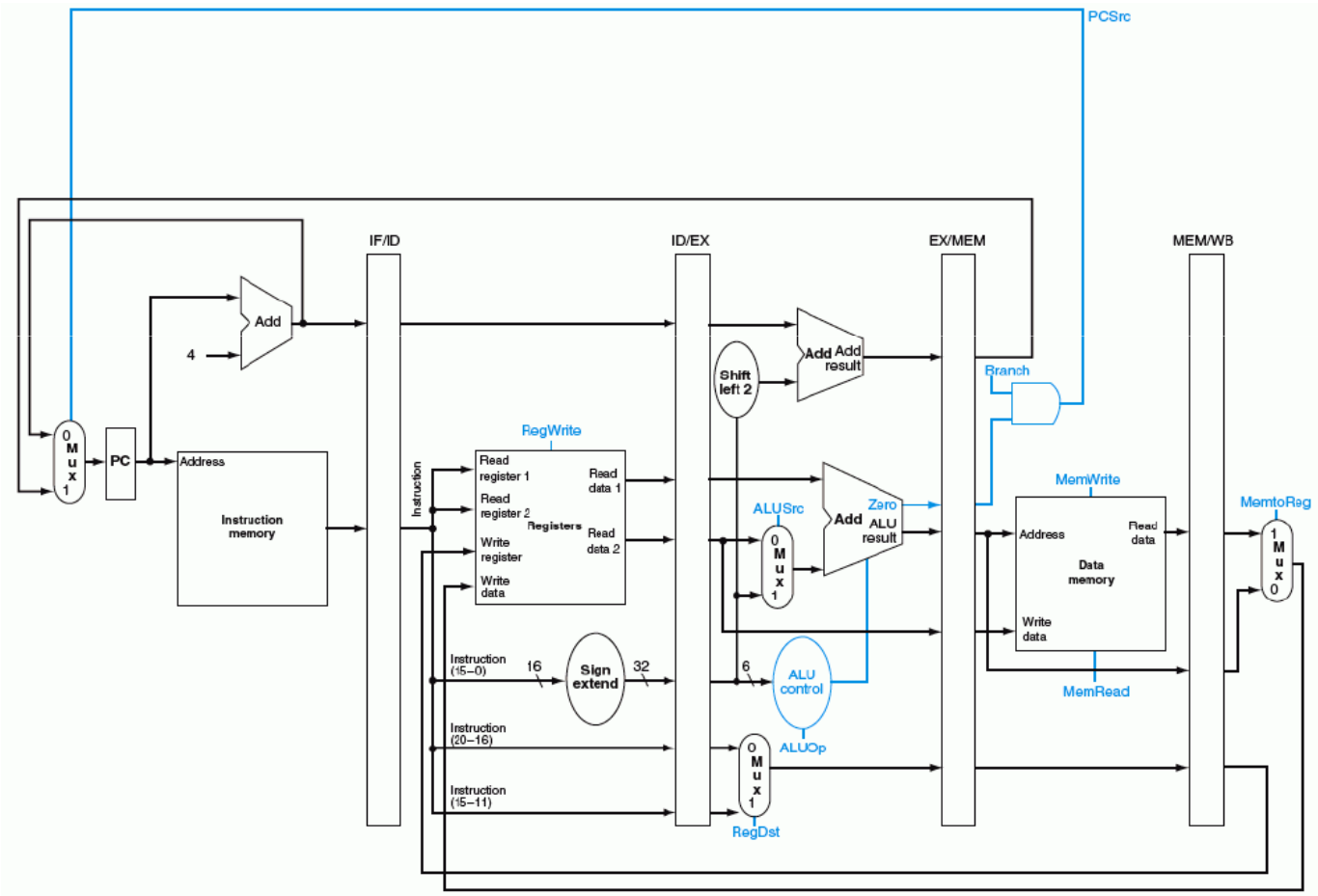
José Luis Hamkalo
Depto. de Electrónica

Estado del Pipe en el 5to Clock



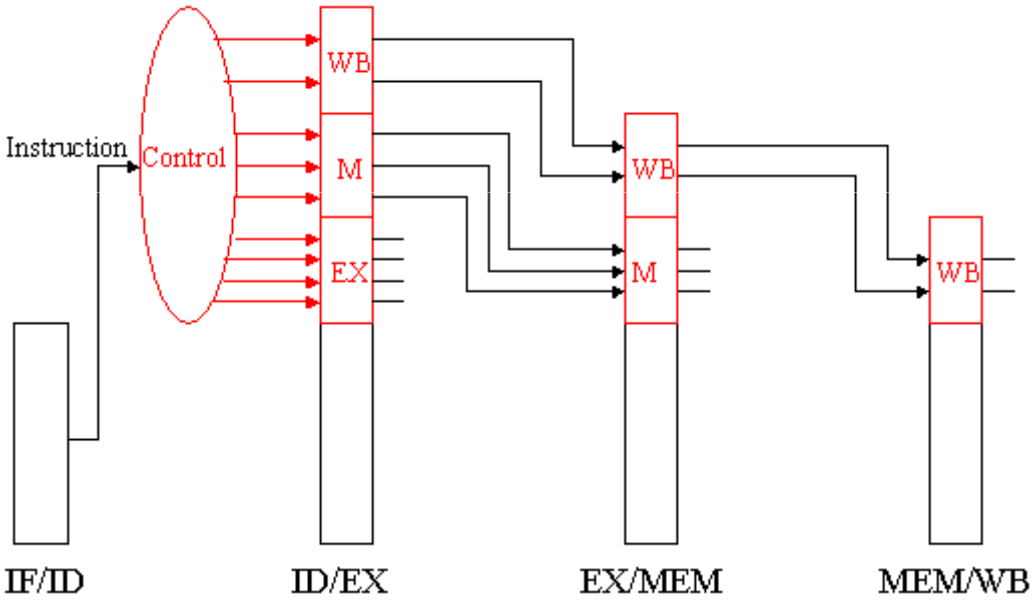
José Luis Hamkalo
Depto. de Electrónica

Con Señales de Control



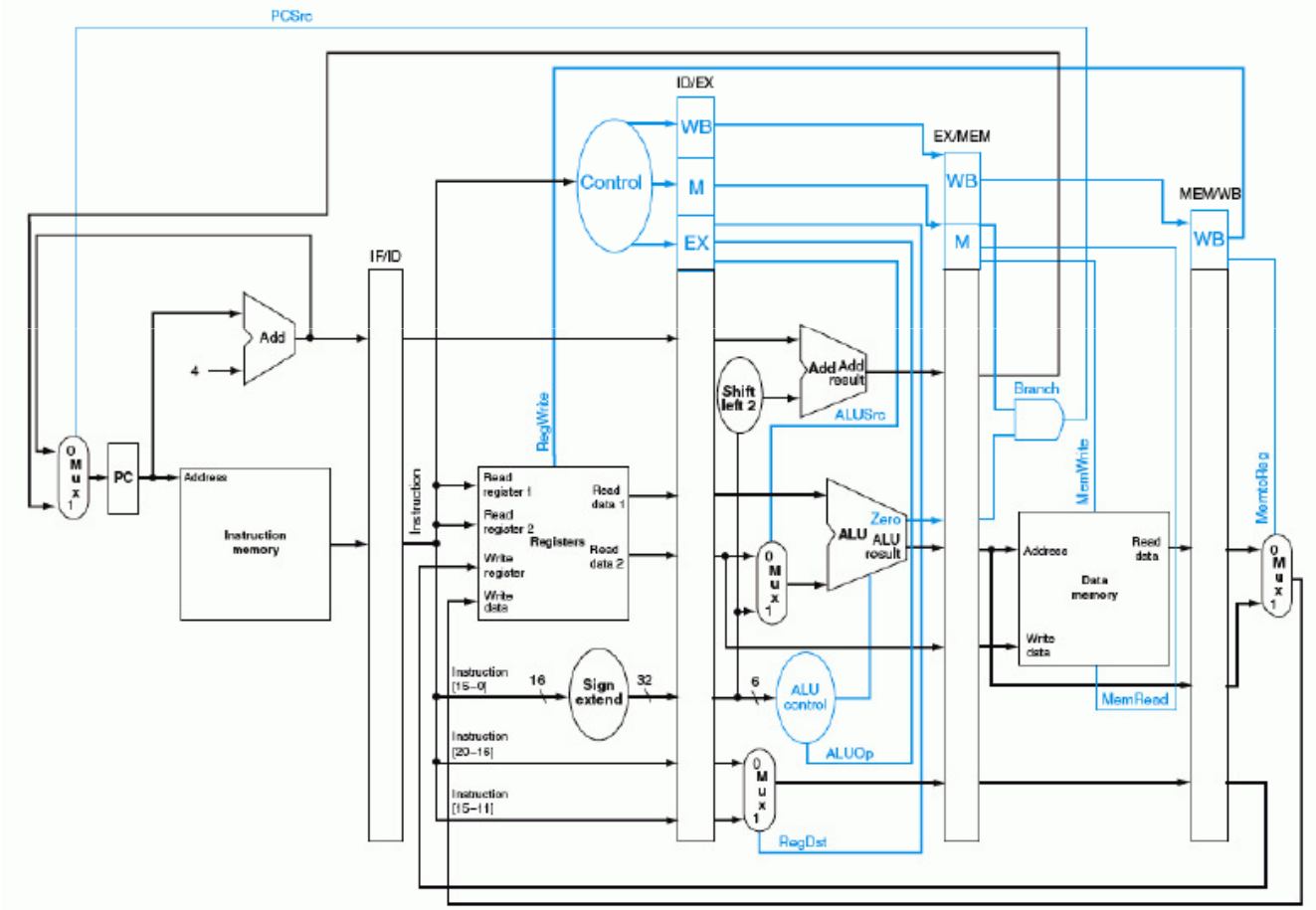
José Luis Hamkalo
Depto. de Electrónica

Transporte de las Señales de Control



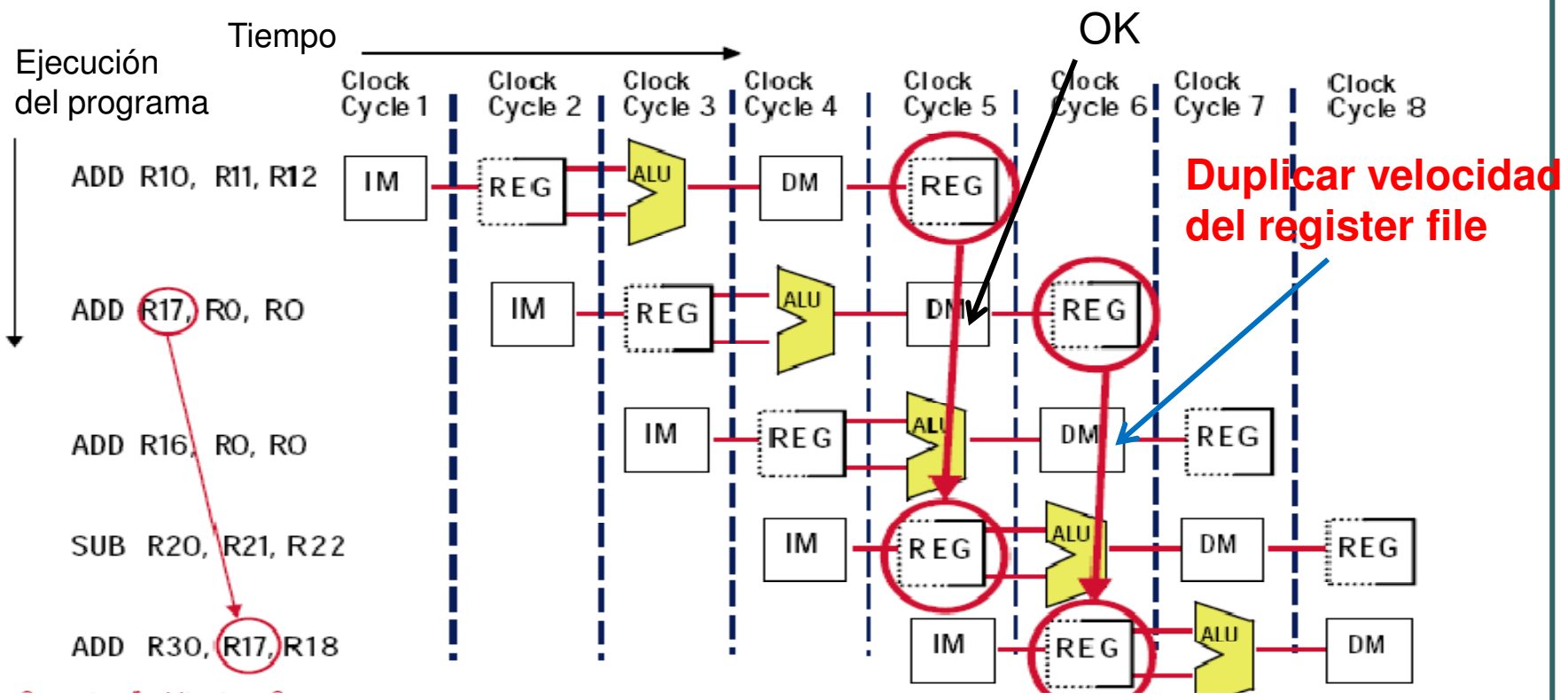
José Luis Hamkalo
Depto. de Electrónica

Con Señales de Control Generadas y Transportadas.



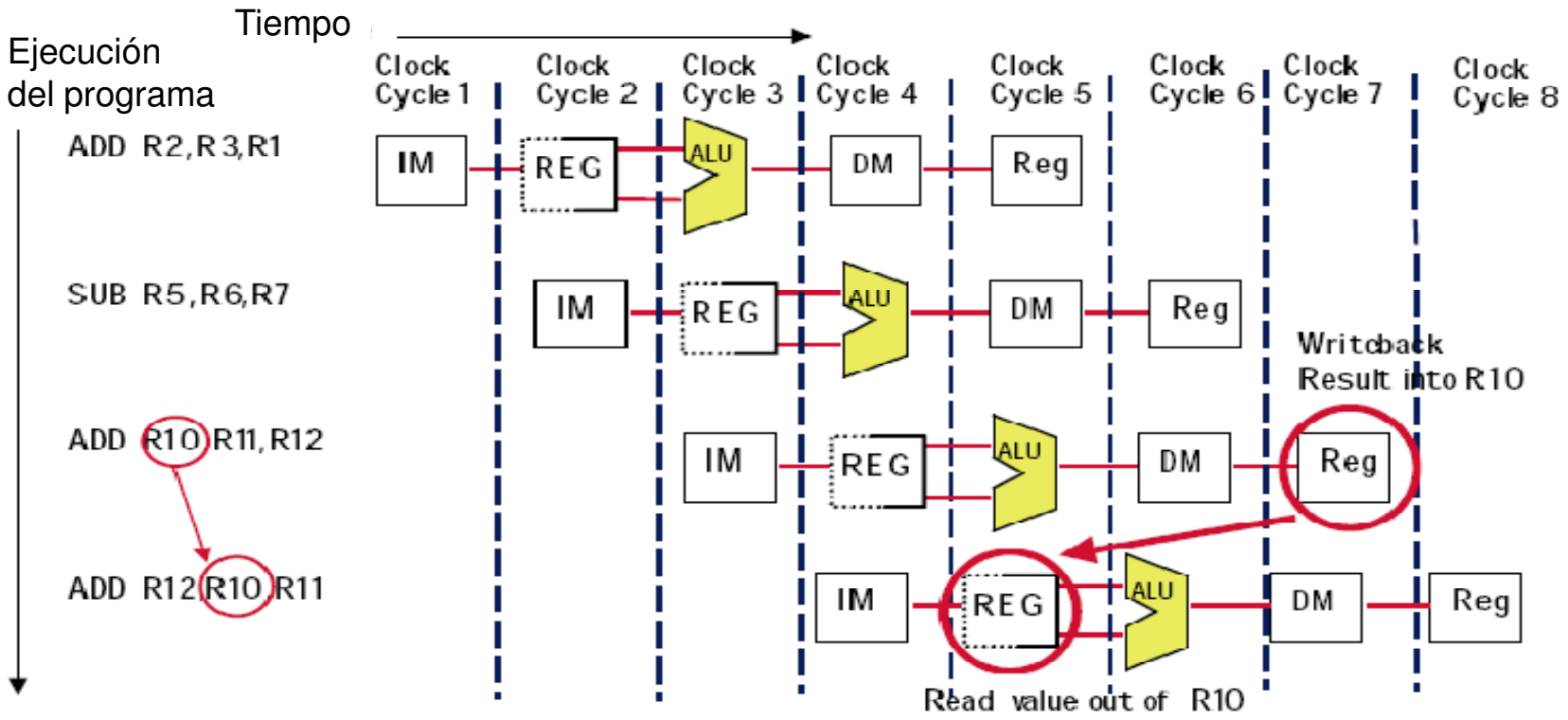
José Luis Hamkalo
Depto. de Electrónica

Contención en el Register File



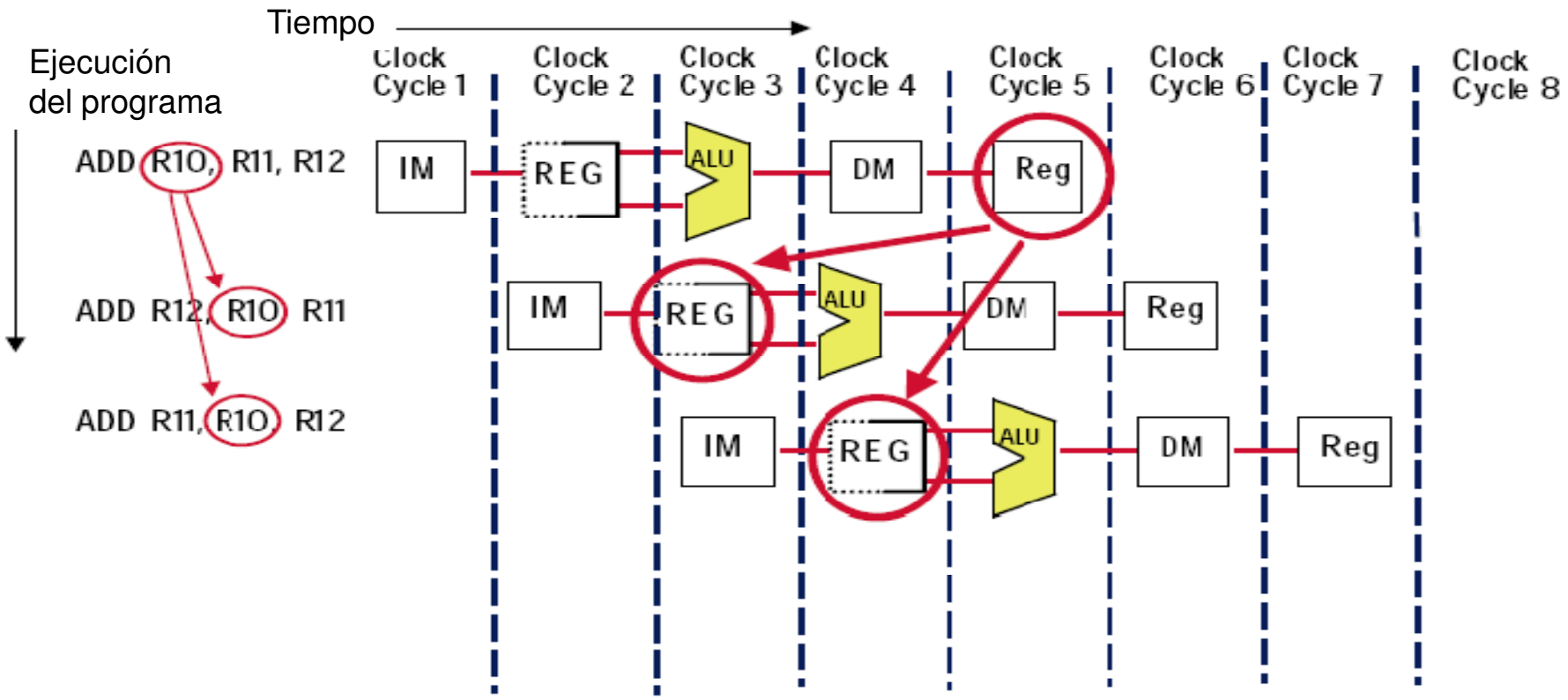
José Luis Hamkalo
Depto. de Electrónica

A Veces los Resultados no Están Listos



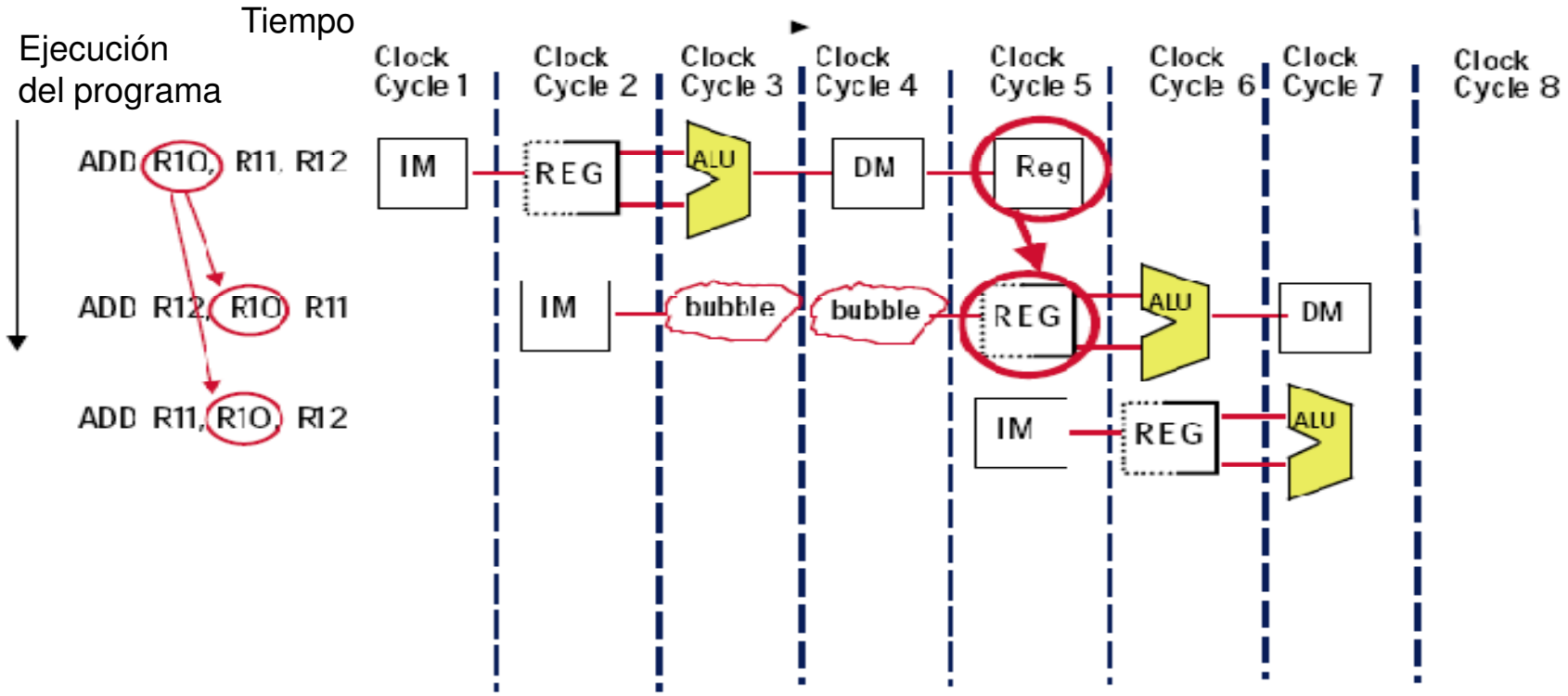
José Luis Hamkalo
Depto. de Electrónica

Ejemplo



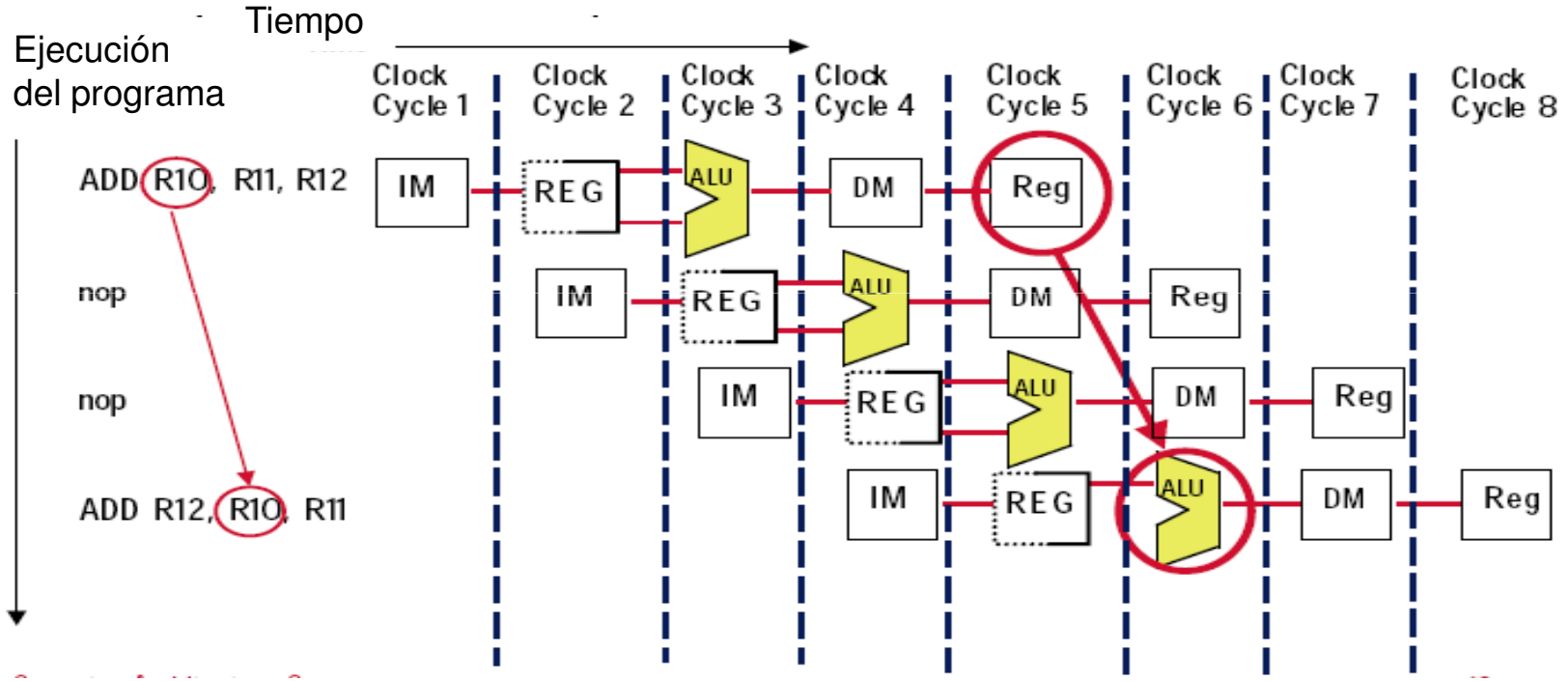
José Luis Hamkalo
Depto. de Electrónica

Solución por Hardware: Parar el Pipeline



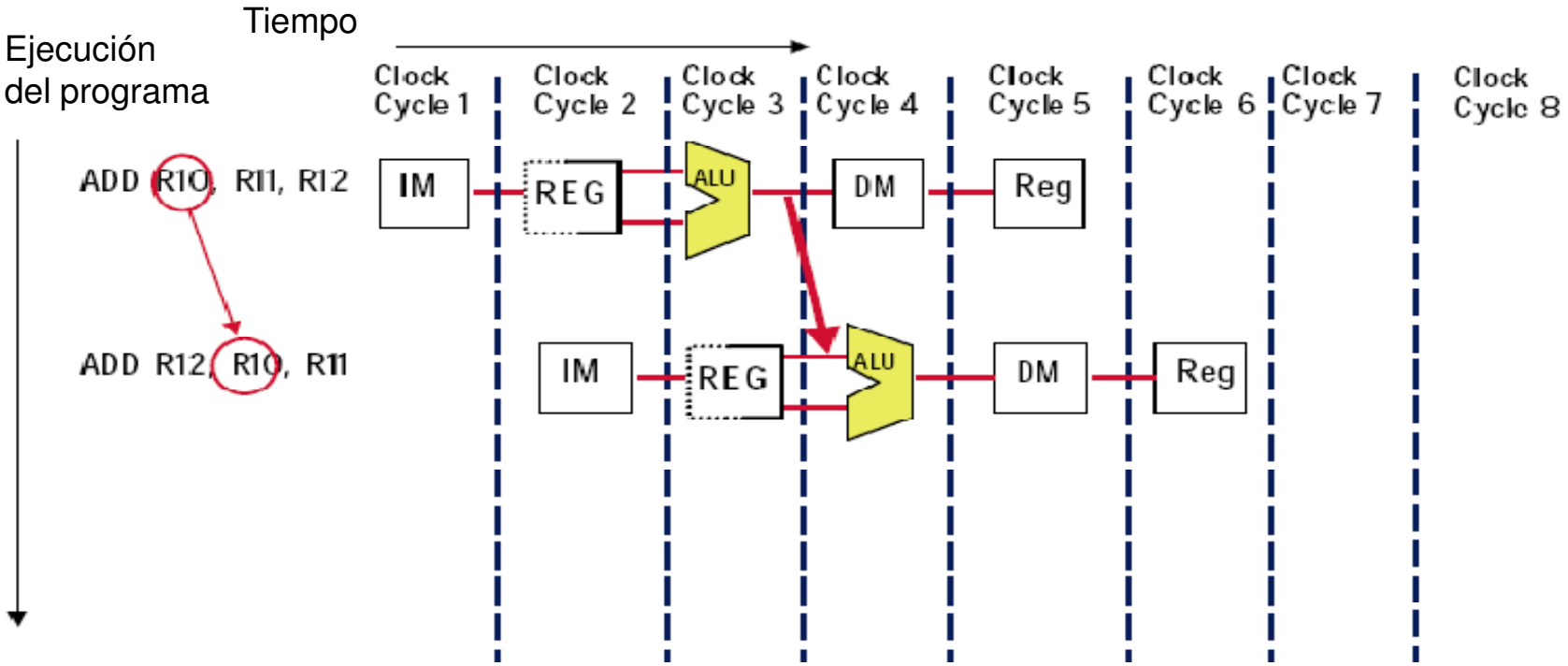
José Luis Hamkalo
Depto. de Electrónica

Solución por Software: El Compilador Inserta No-OPeraciones (NOP)



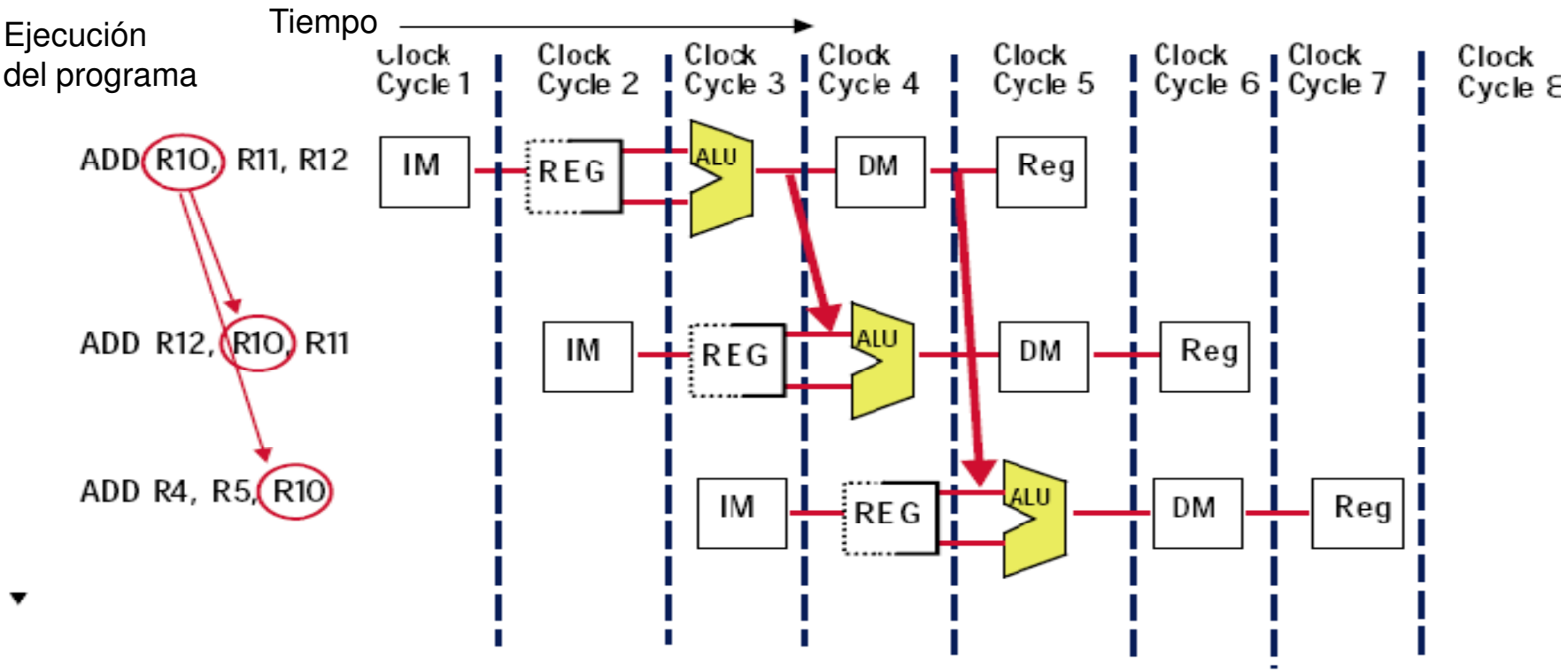
José Luis Hamkalo
Depto. de Electrónica

Evitar los Ciclos de Stall: Forwarding



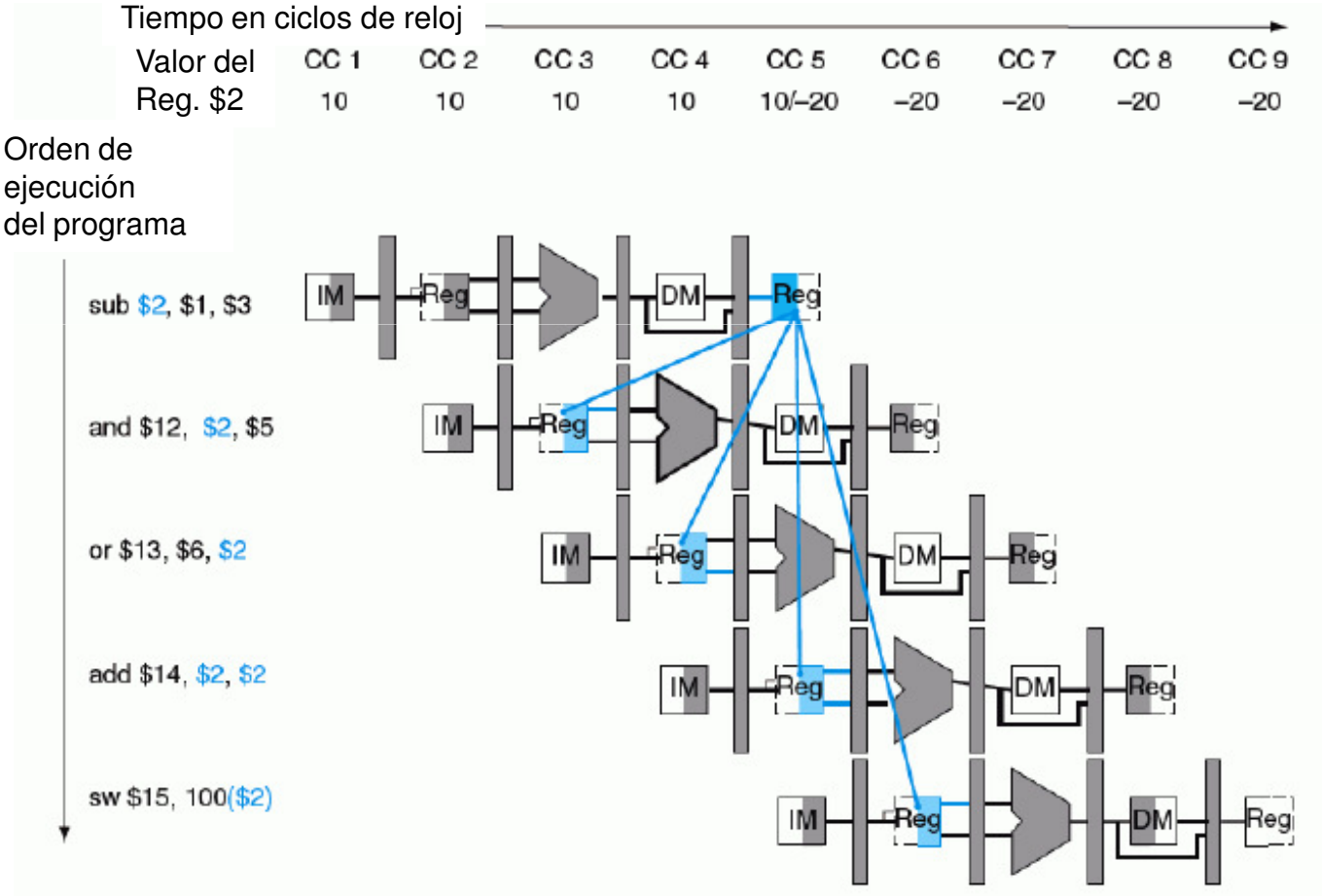
José Luis Hamkalo
Depto. de Electrónica

Forwarding, continuación (2)



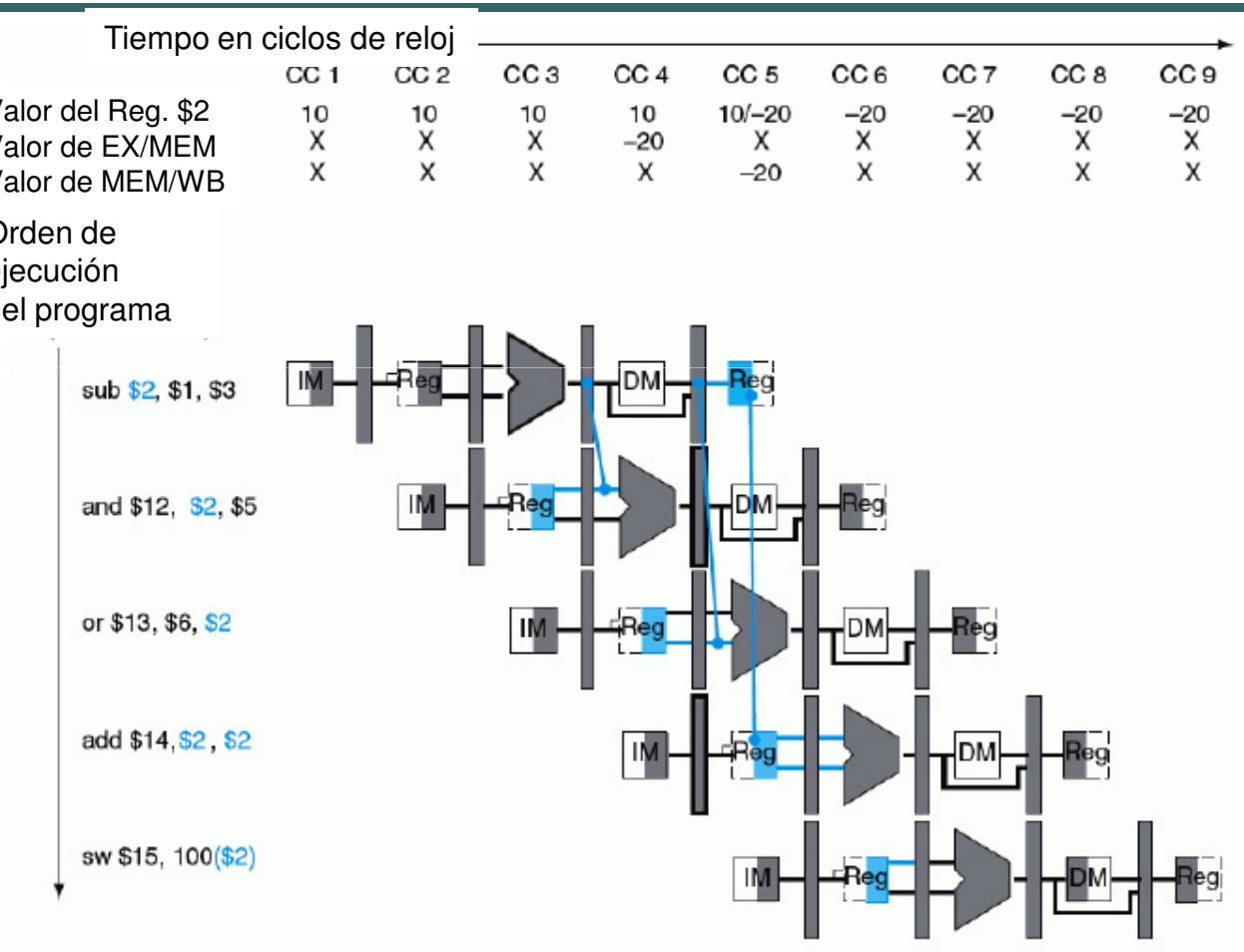
José Luis Hamkalo
Depto. de Electrónica

Dependencias en una Secuencia de 5 Instrucciones



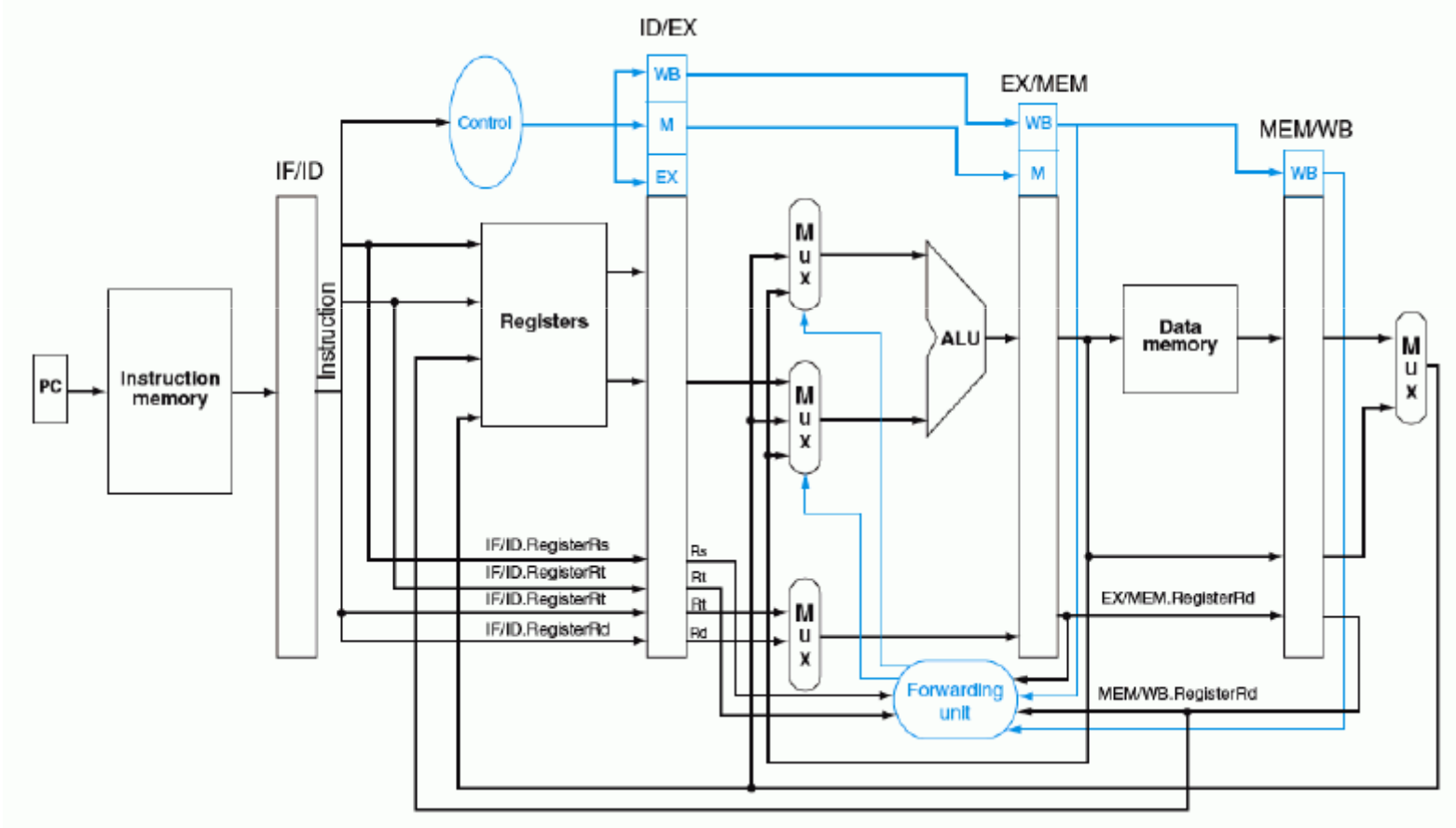
José Luis Hamkalo
Depto. de Electrónica

Dependencias en una Secuencia de 5 Instrucciones: los Datos Pueden Proveerse a Tiempo.



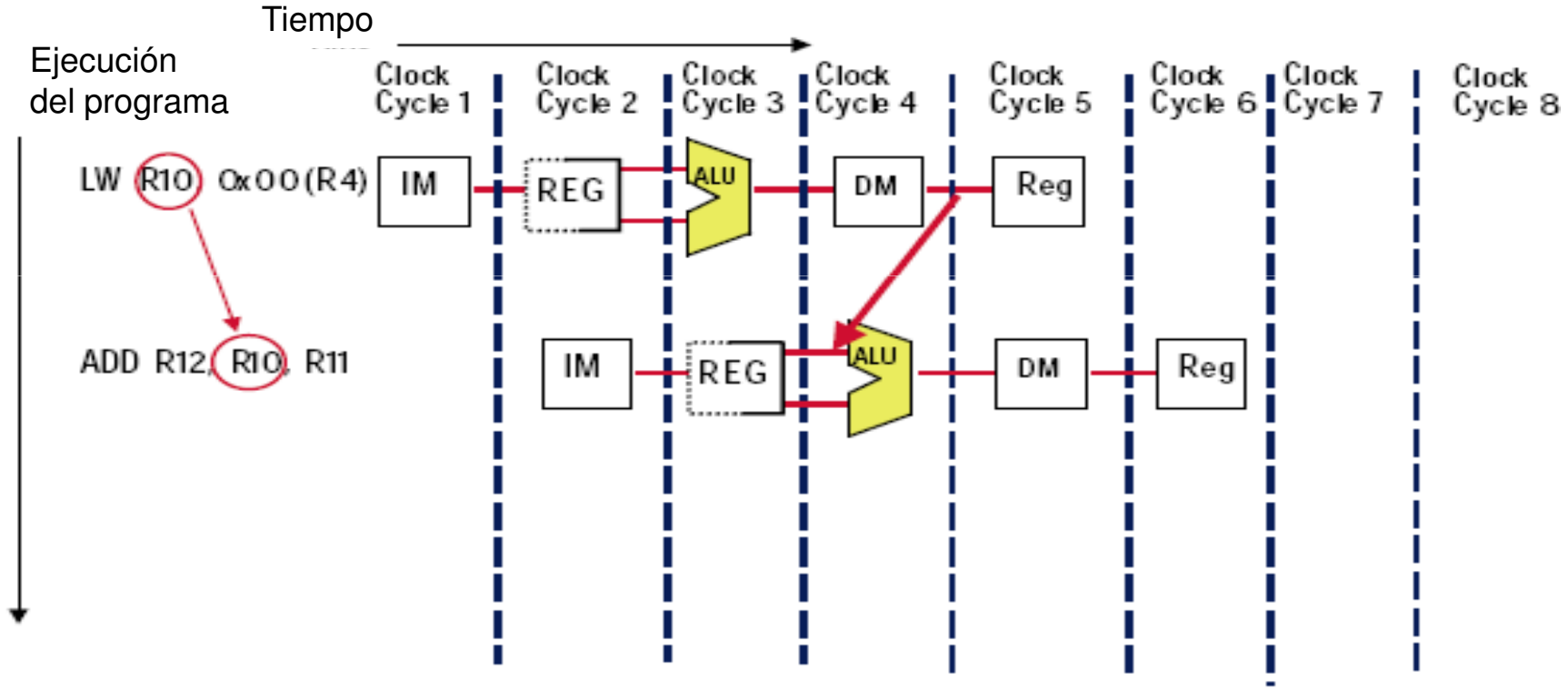
José Luis Hamkalo
Depto. de Electrónica

Agregado del Hardware de Forwarding al Camino de Datos



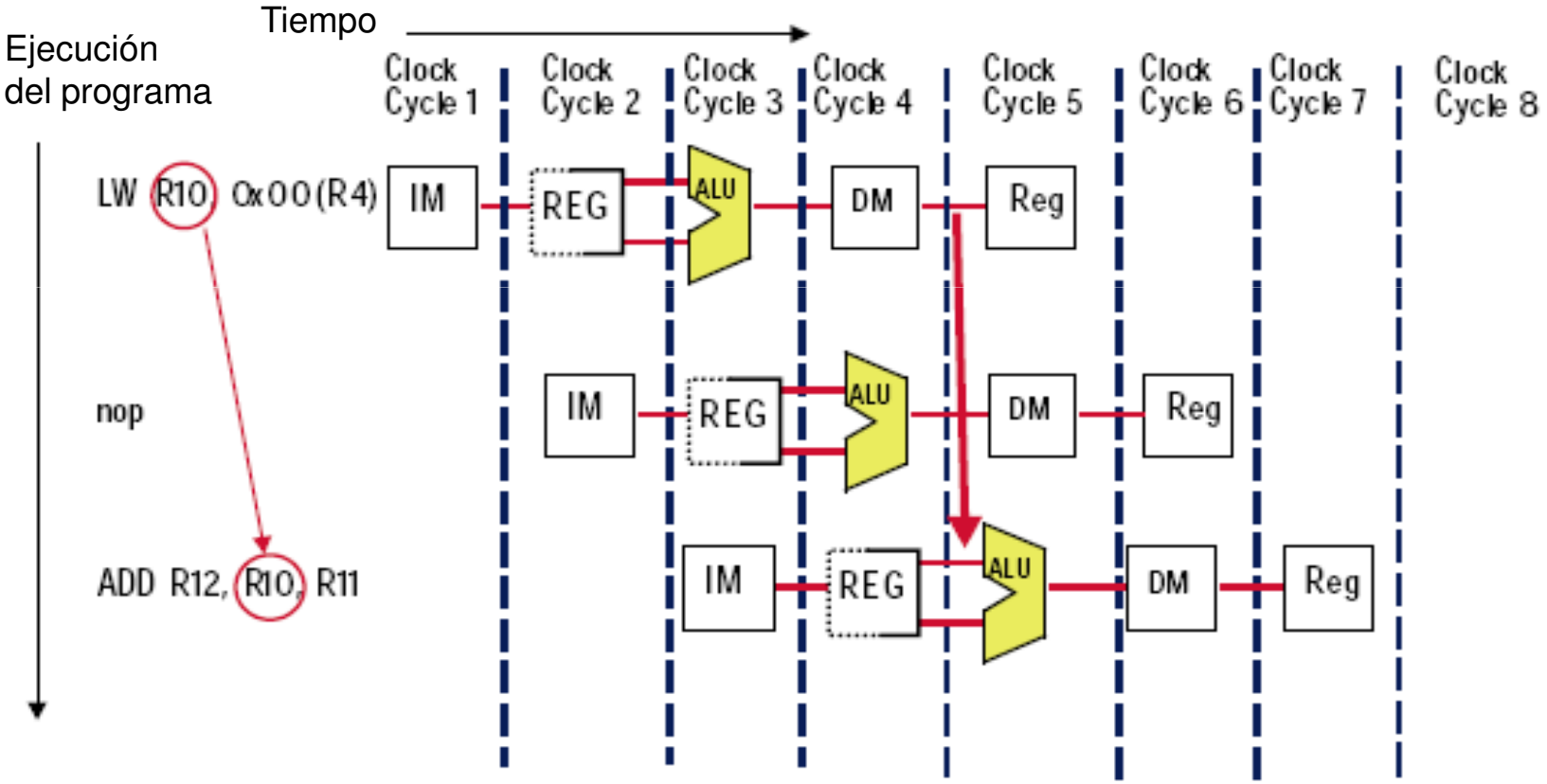
José Luis Hamkalo
Depto. de Electrónica

Forwarding, no siempre aplicable.



José Luis Hamkalo
Depto. de Electrónica

MIPS de 5 etapas requiere de un “load delay slot”



José Luis Hamkalo
Depto. de Electrónica

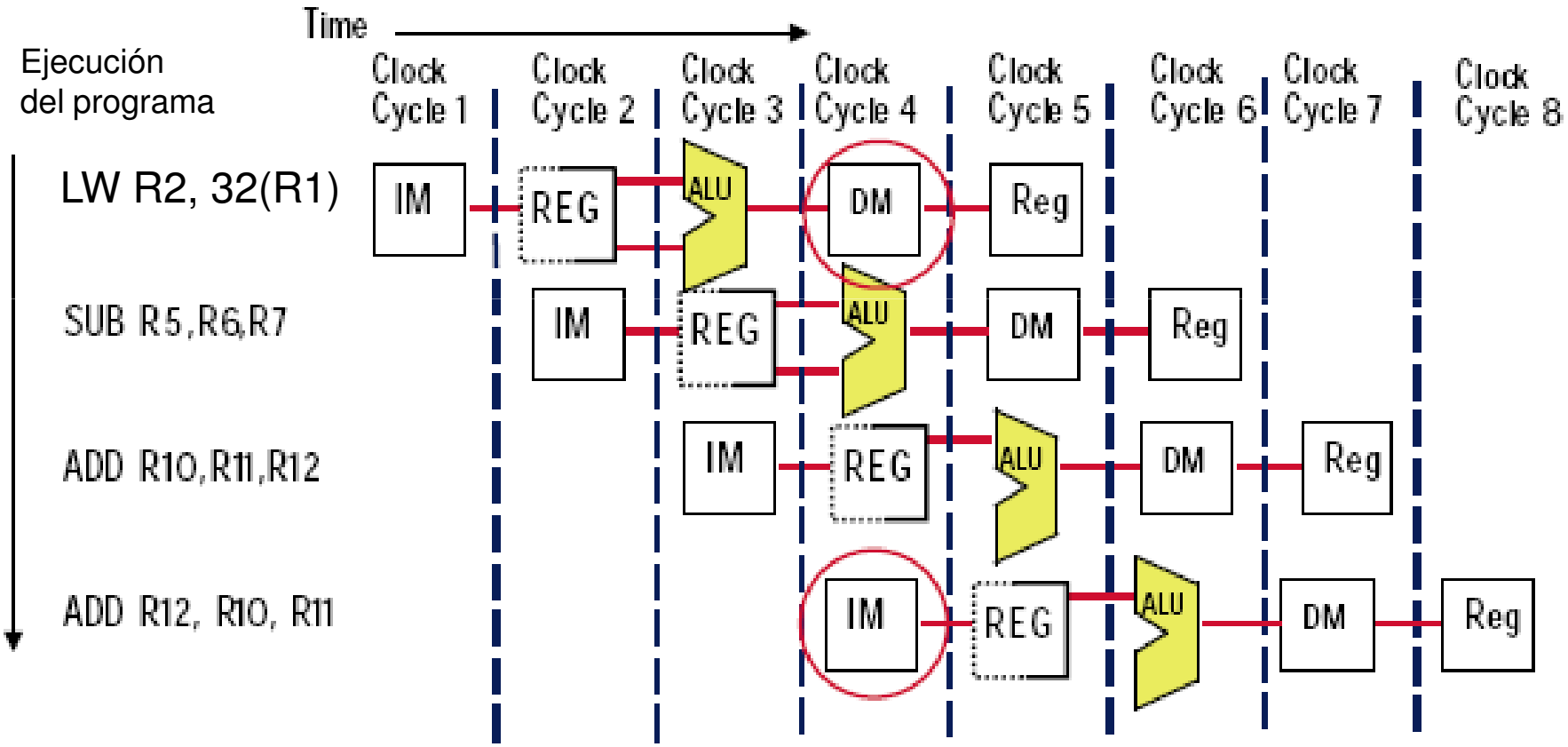
Taxonomía de los Riesgos

- Estructurales
- De Datos
- De Control

Riesgos Estructurales

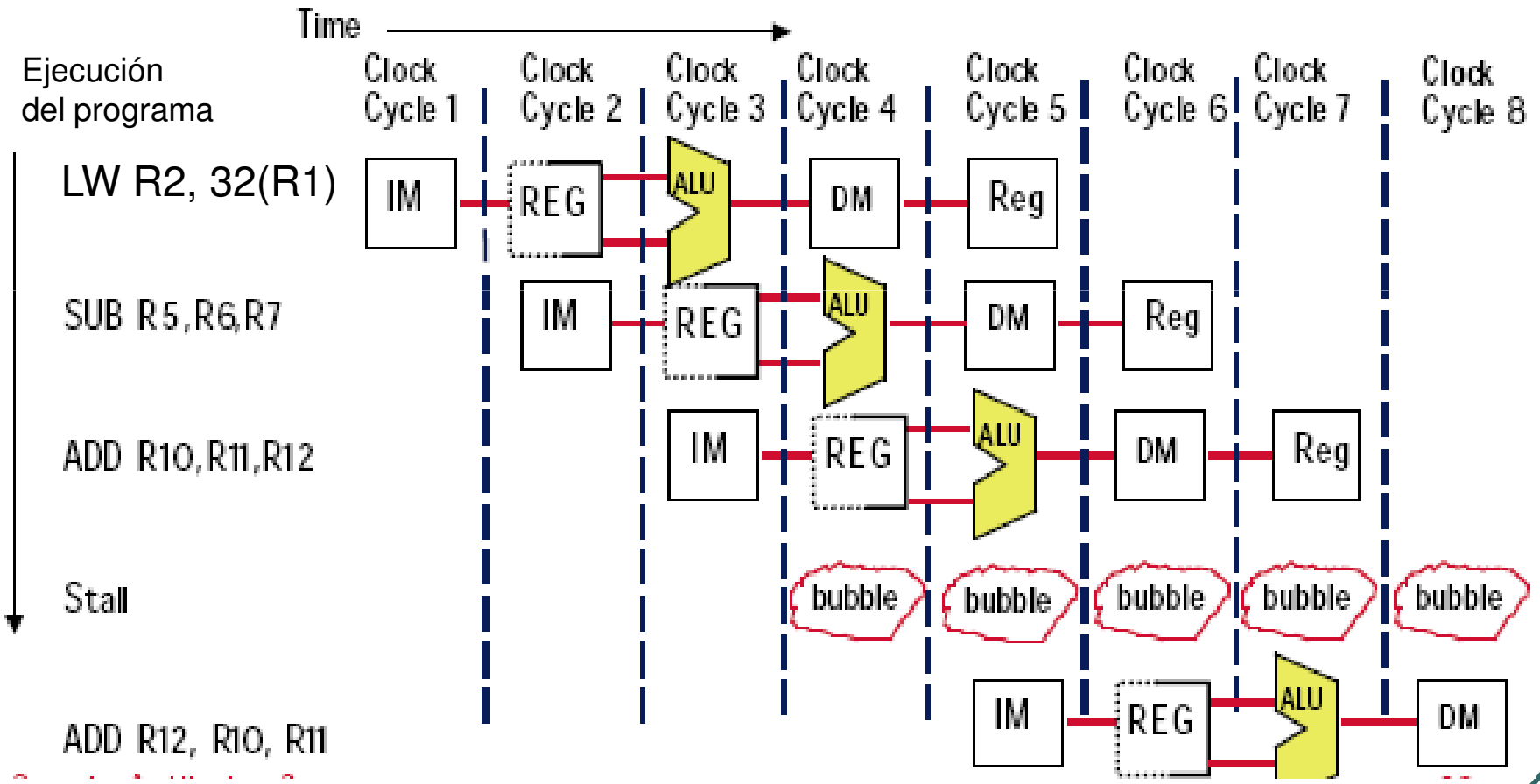
- Las instruciones no se pueden ejecutar por no haber suficientes recursos.
- Ejemplos
  - Memoria unificada
  - Register file no multipuerto
  - Unidad funcional requiere más de un ciclo

Riesgo estructural, ejemplo: memoria unificada (inst. y datos)



José Luis Hamkalo
Depto. de Electrónica

Ejemplo (continuación). Solución: Para el Pipeline



Riesgos de Datos

- Lectura después de Escritura (RAW)
  - La inst *i+1* lee después que la inst. *i* escribe
- Escritura después de Lectura (WAR)
  - La inst *i+1* escribe después que la inst. *i* lee
- Escritura después de Escritura (WAW)
  - La inst *i+1* escribe después que la inst. *i* escribe
- Lectura después de Lectura (RAR)
  - La inst *i+1* lee después que la inst. *i* lee (nunca es riesgo)

Riesgos de Control

Instrucciones que alteran el contenido del PC

Ejemplo: branch

Ejemplo de Código

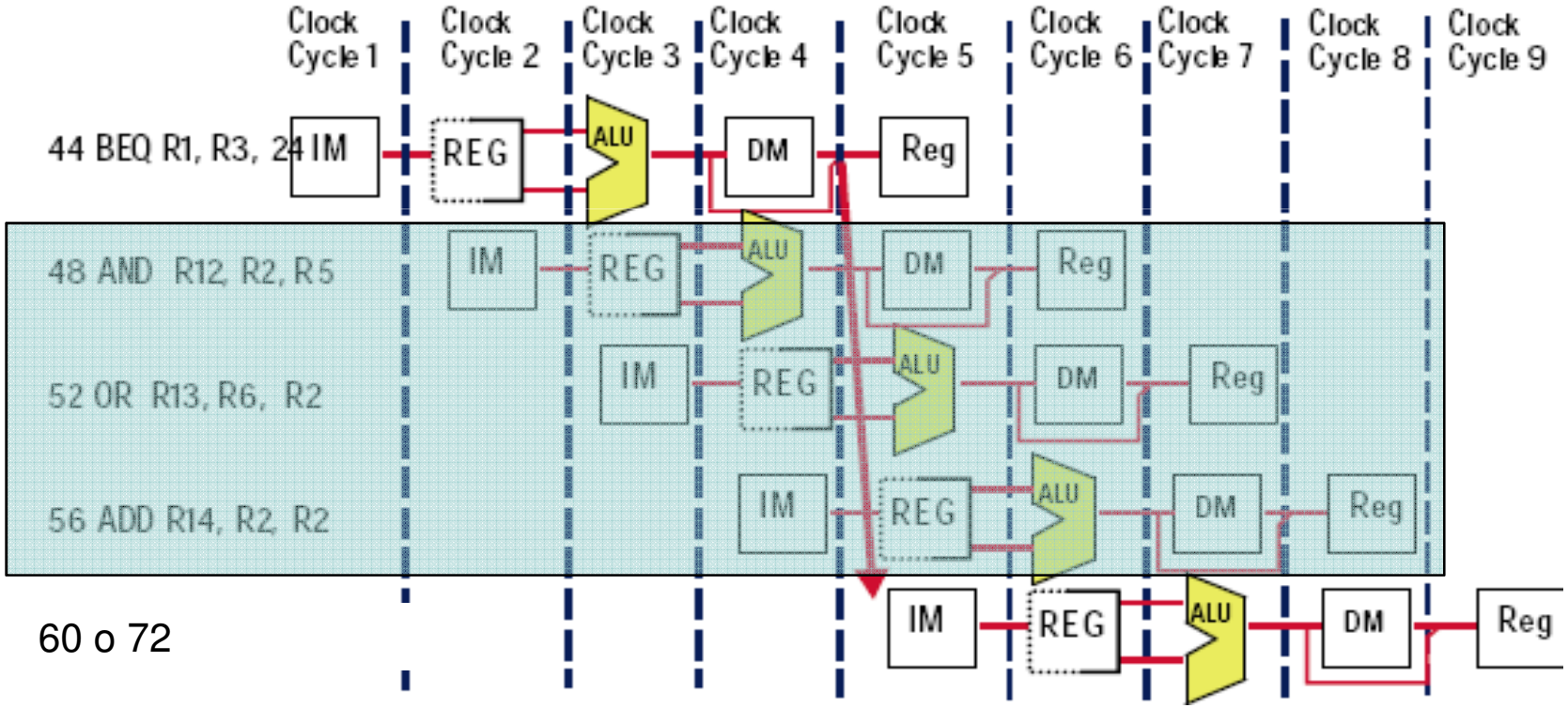
```
36 NOP
40 ADD R30, R30, R30
44 BEQ R1, R3, 24
48 AND R12, R2, R5
52 OR R13, R6, R2
56 ADD R14, R2, R2
60 …
64 …
68 …
72 LW R4, 50(R7)
76 …
```

Flujo de instrucciones si el branch es tomado: 36, 40, 44, 72

Flujo de instrucciones si el branch es no tomado: 36, 40, 44, 48

Ejemplo, Riesgos Debido al branch

Flujo de instrucciones si el branch es tomado: 36, 40, 44, 72
Flujo de instrucciones si el branch es no tomado: 36, 40, 44, 48

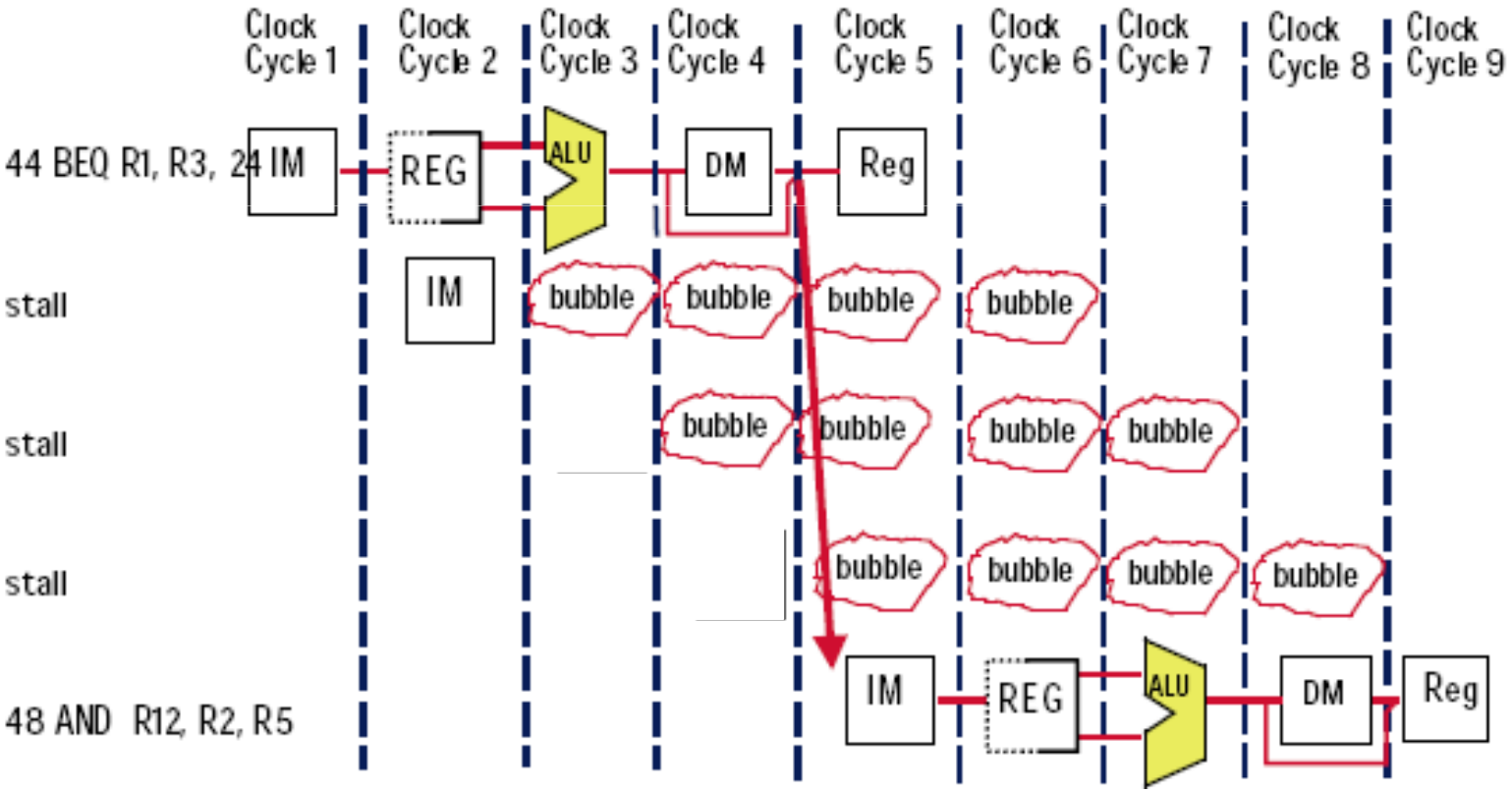


José Luis Hamkalo
Depto. de Electrónica

Predicción de Saltos

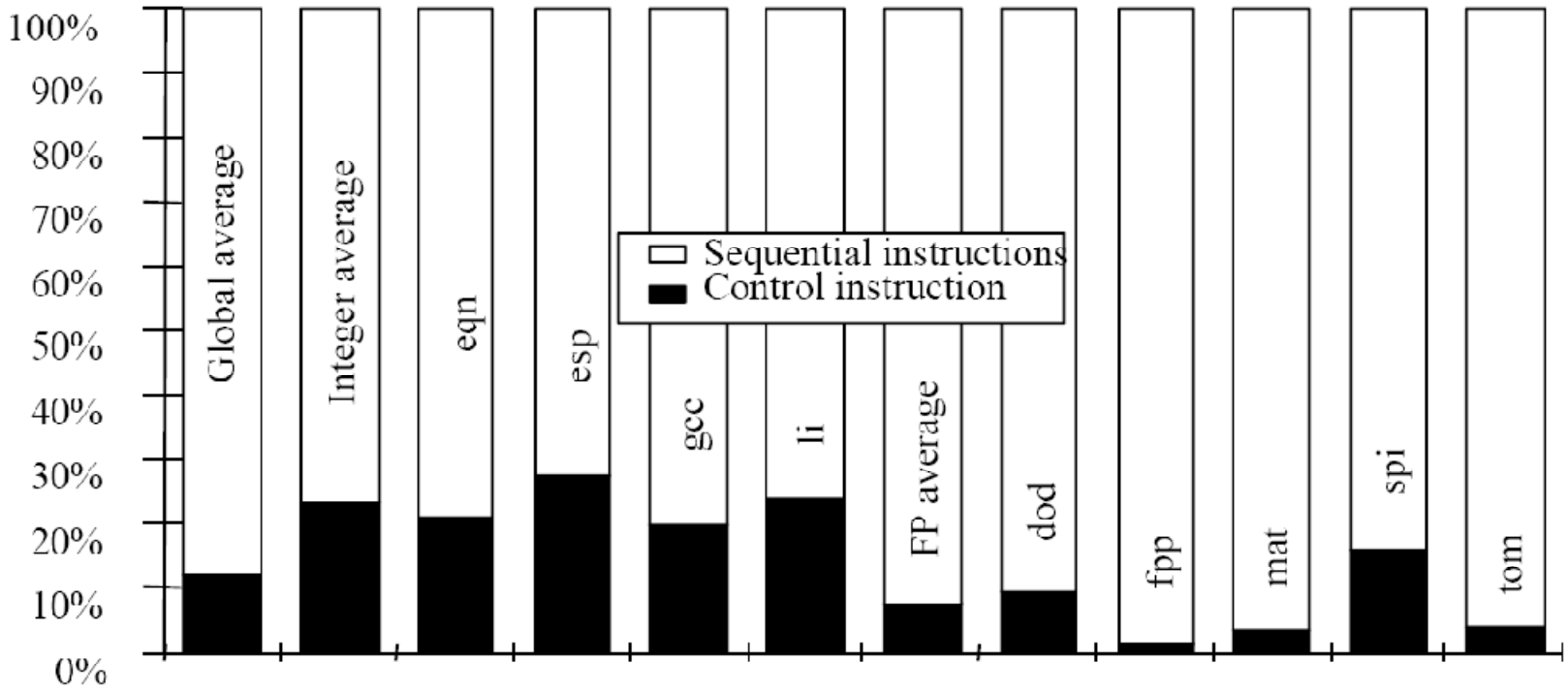
- Predicción Estática
  - Hardware
  - Software
    - Información conocida en tiempo de compilación.
    - Profiling del programa
- Predicción Dinámica
  - Predictores locales
    - predictores de 1 o 2 bits
  - Predictores globales
    - Historia de saltos
  - Predictores en 2 niveles

Solución 1, Parar el Pipeline



José Luis Hamkalo
Depto. de Electrónica

Instrucciones de Control

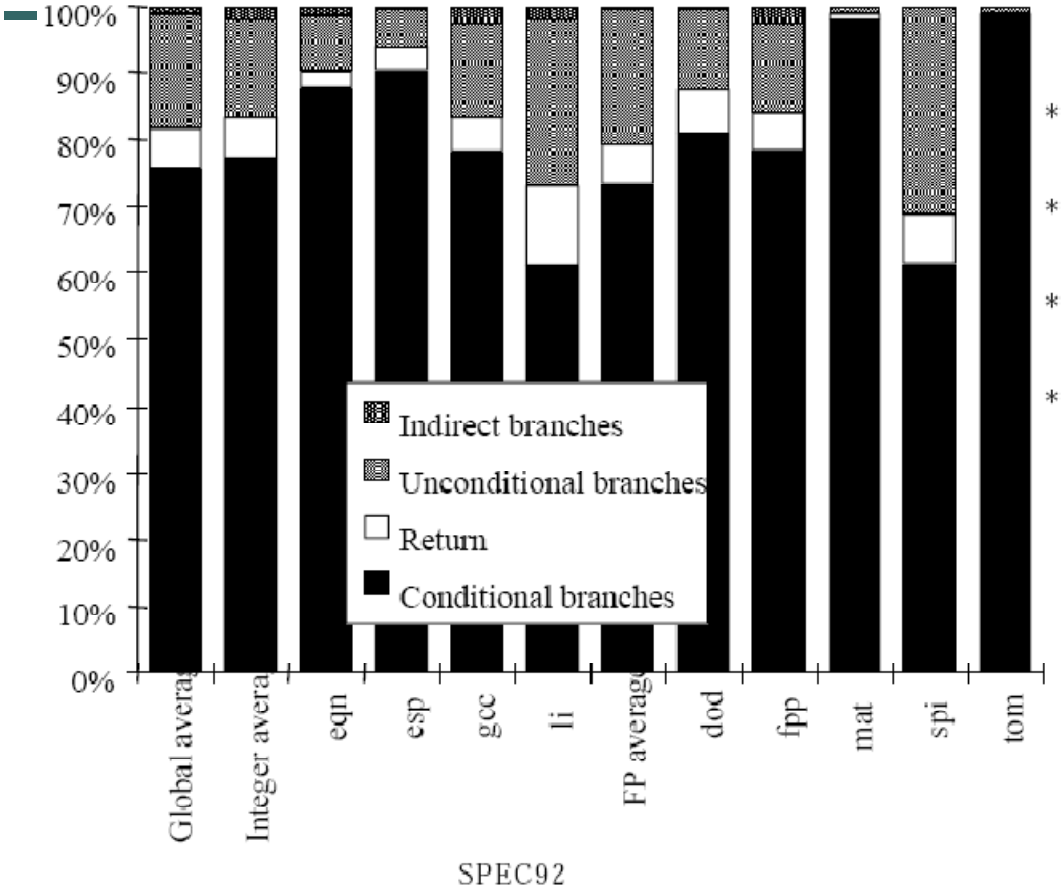


Muy frecuentes: alto impacto en el desempeño

SPEC92

José Luis Hamkalo
Depto. de Electrónica

Mezcla de Saltos

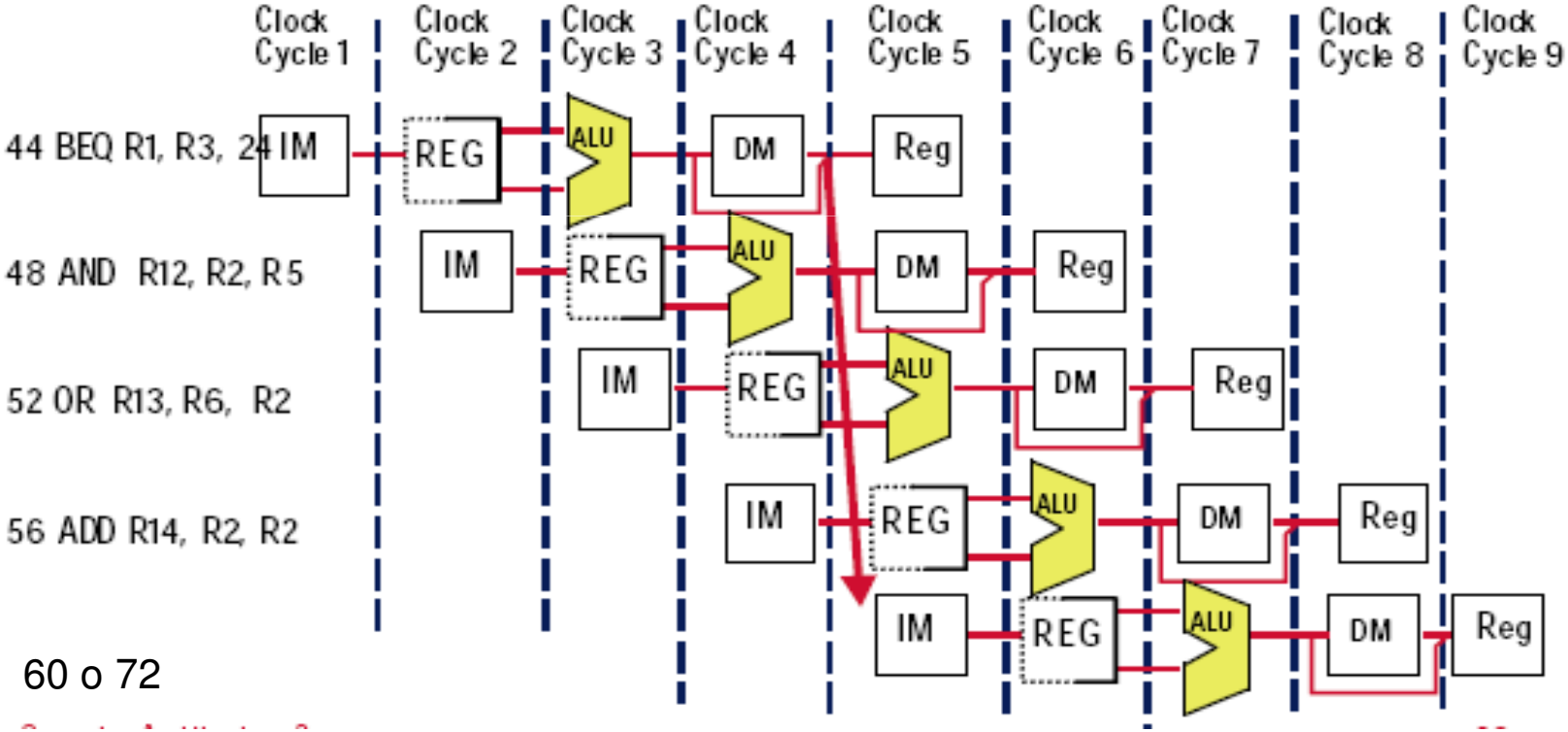


* CONDITIONAL BRANCHES
  Bcond Ri, offset
* RETURN
  RET or JMP R31
* UNCONDITIONAL BRANCHES
  B offset
* INDIRECT BRANCHES
  JMP Ri

José Luis Hamkalo
Depto. de Electrónica

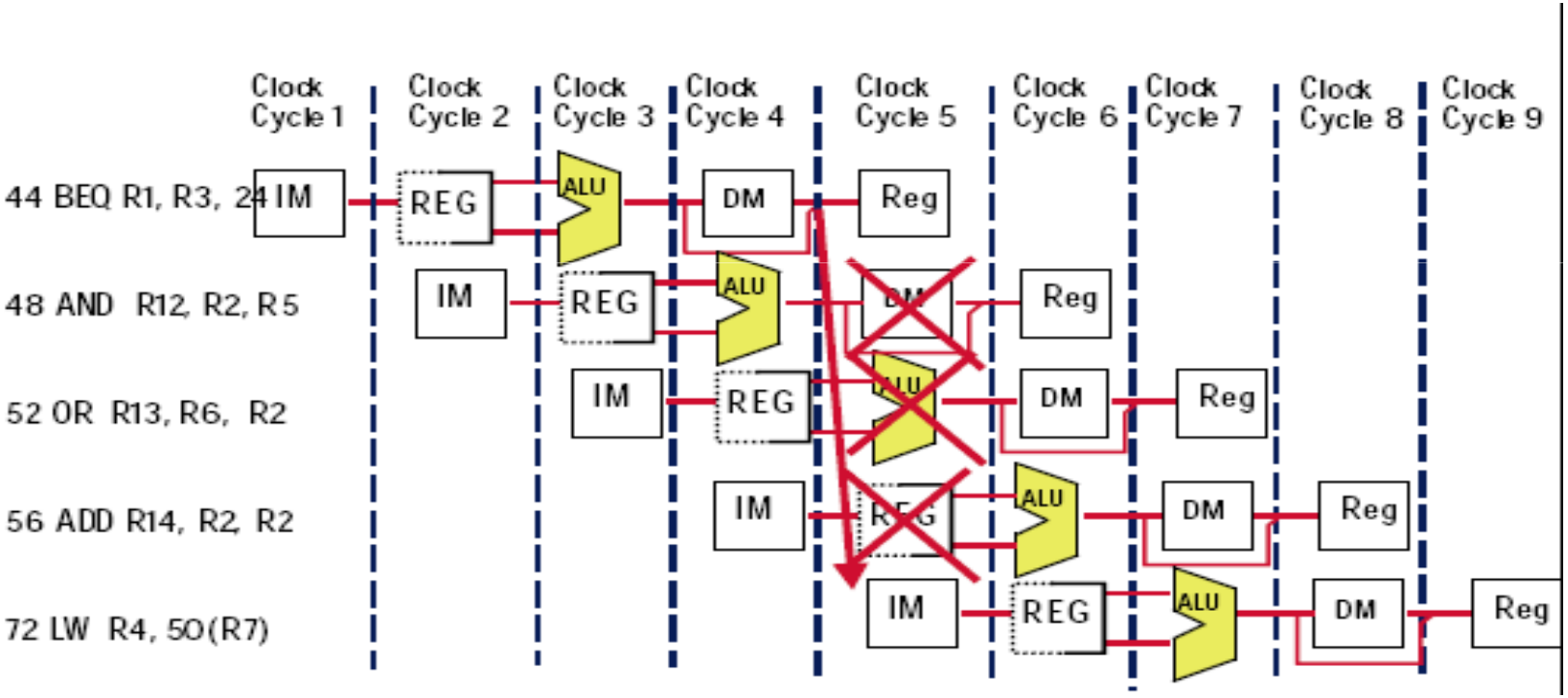
Solución 2, Asumir Salto NO Tomado

Flujo de instrucciones si el branch es tomado: 36, 40, 44, 72
Flujo de instrucciones si el branch es no tomado: 36, 40, 44, 48



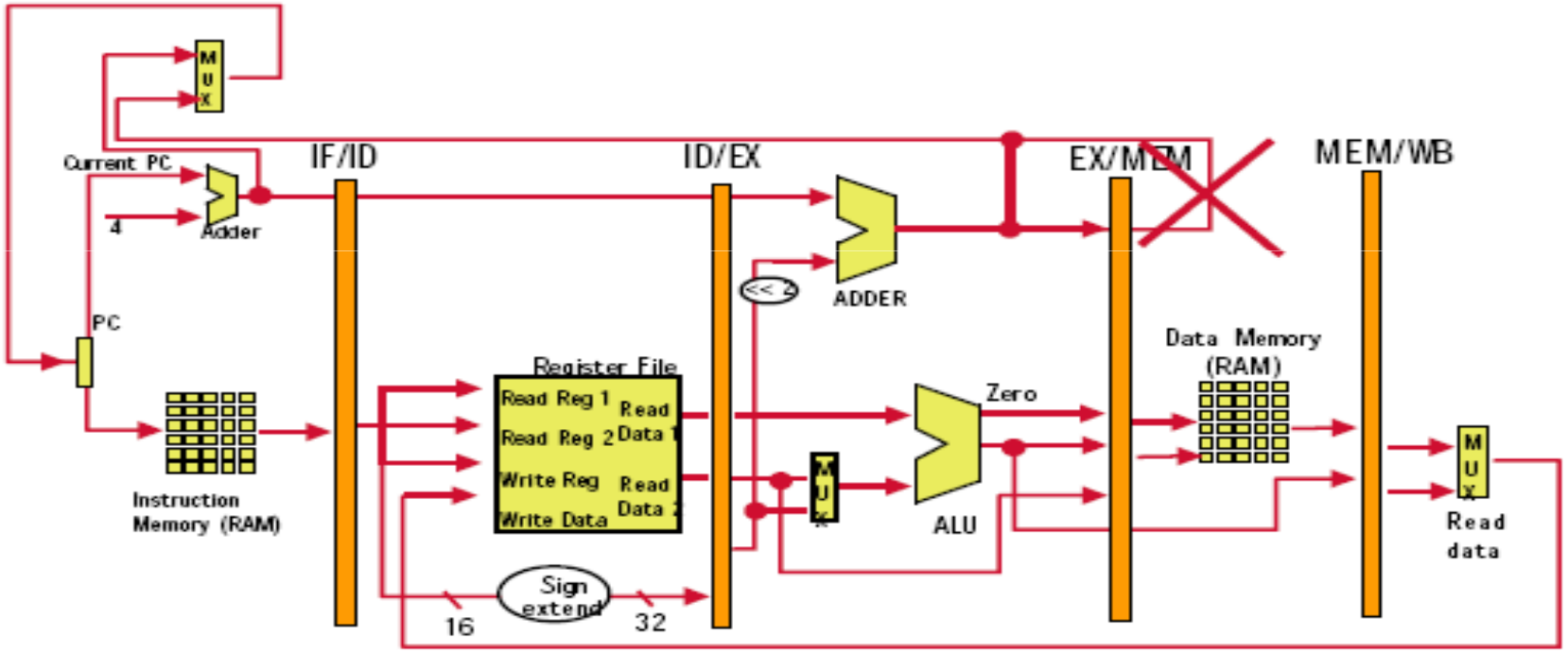
José Luis Hamkalo
Depto. de Electrónica

Qué Pasa Si el Salto es Tomado?



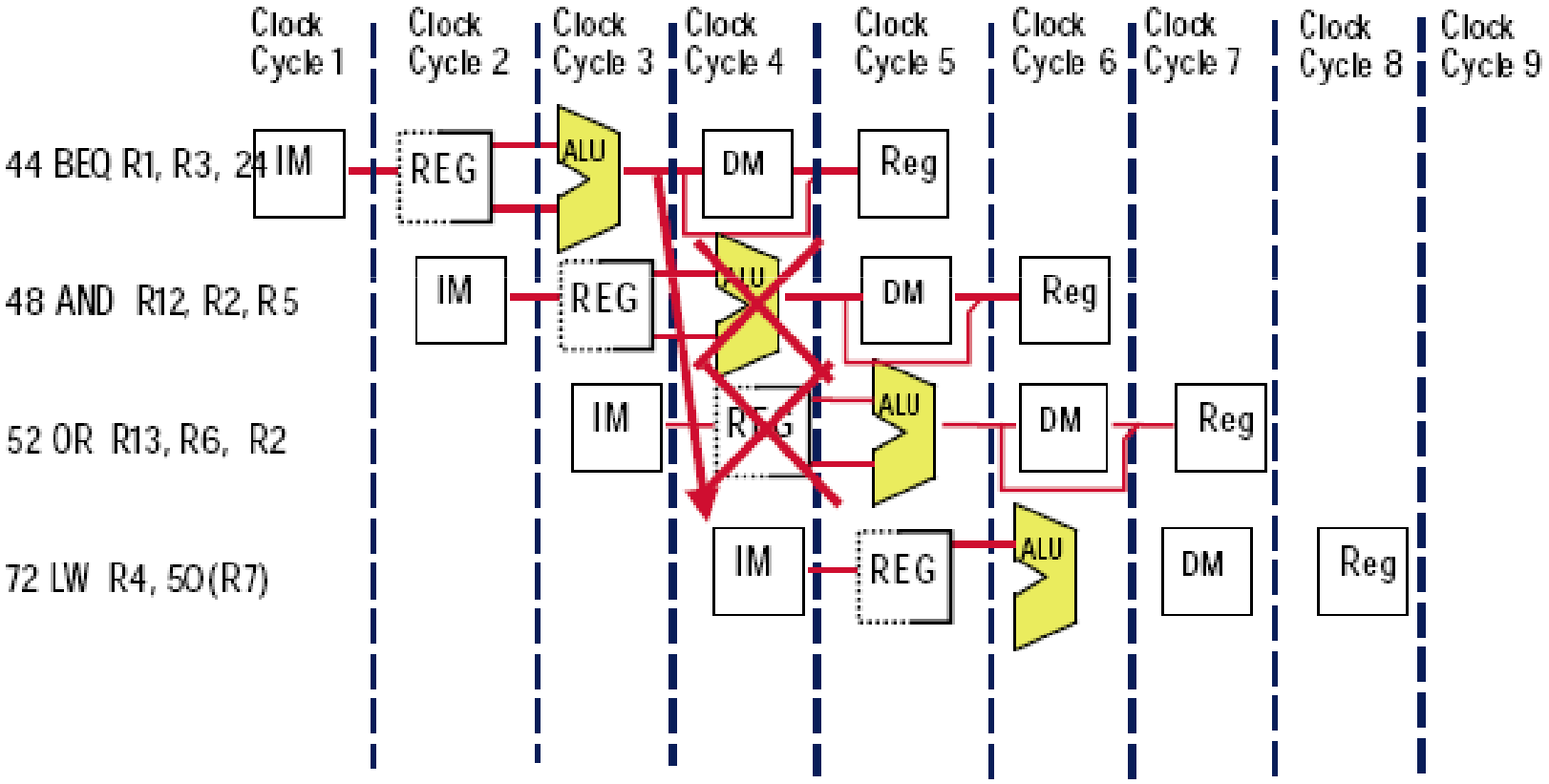
José Luis Hamkalo
Depto. de Electrónica

Mejora: Mover la Señal que Gobierna el Multiplexor del PC a la Etapa EX



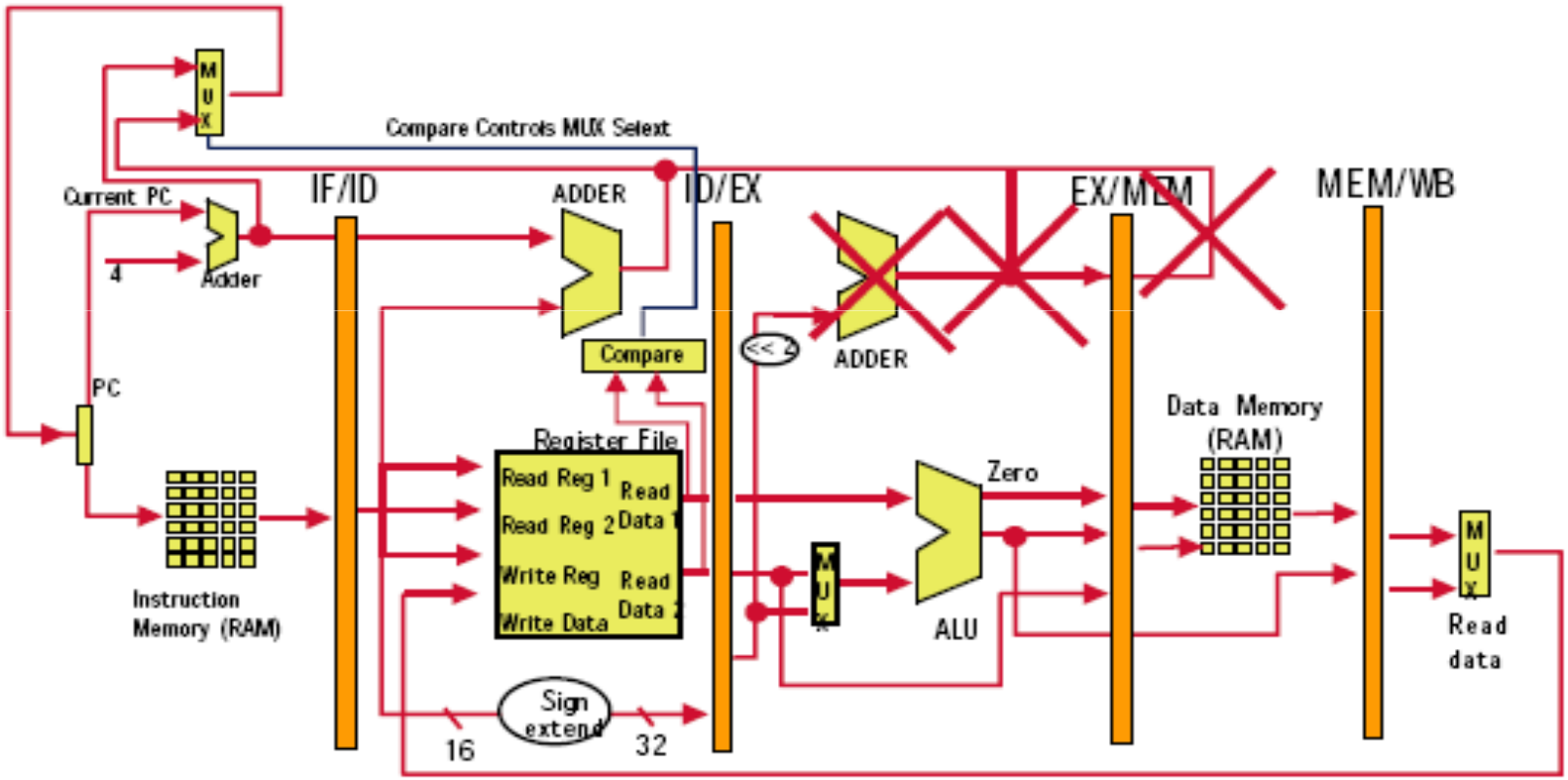
José Luis Hamkalo
Depto. de Electrónica

Reduce Penalidad en 1 Ciclo



José Luis Hamkalo
Depto. de Electrónica

Mover Aún Más Hacia Adelante para Mejorar el Desempeño.

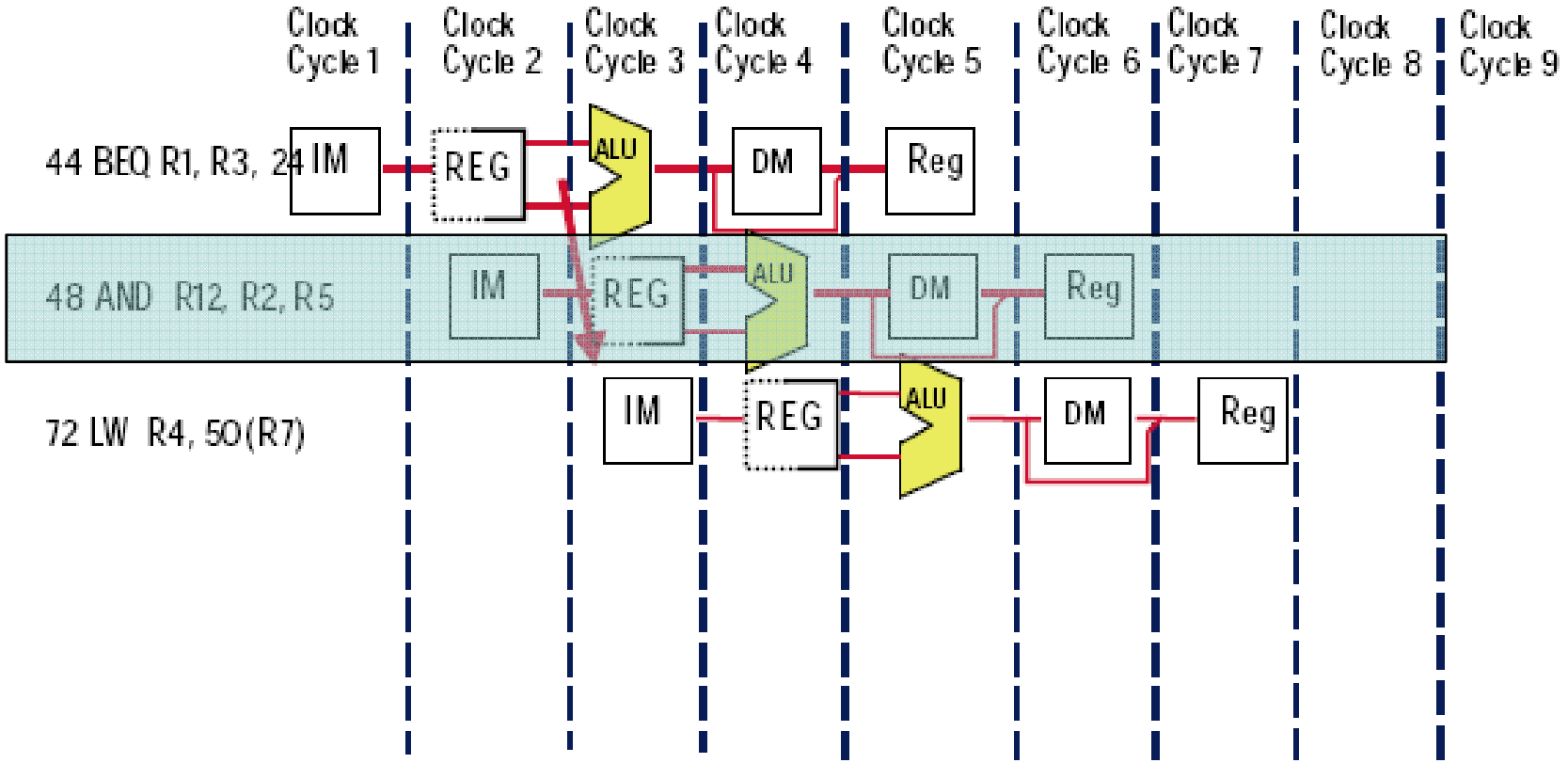


José Luis Hamkalo
Depto. de Electrónica

El Esquema del Salto Demorado

- Técnica que pasa la responsabilidad de los riesgos de control al compilador.
- El hardware se desentiende de los riesgos de control.
- El procesador siempre ejecuta la instrucción siguiente al branch (1 delay slot).

El “branch delay slot”

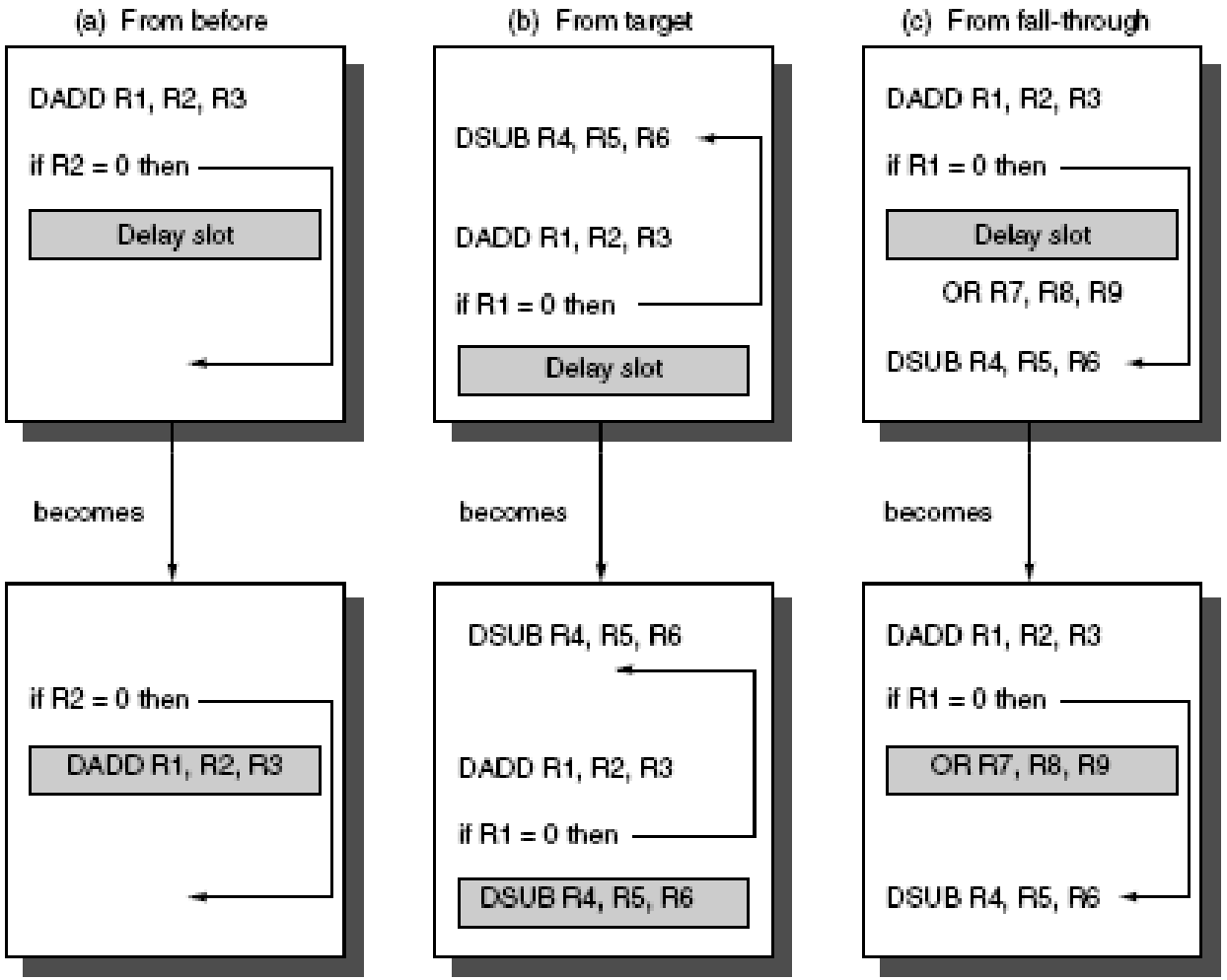


José Luis Hamkalo
Depto. de Electrónica

Manejo del “branch delay slot”

- Se debe garantizar que el programa se ejecute correctamente.
- Tratar de reordenar las instrucciones sin alterar la lógica del programa.
- Colocar una NOP luego de cada salto en la que no fue posible garantizar que el programa se ejecute correctamente.

Branch delay slot: tres posibilidades



José Luis Hamkalo
Depto. de Electrónica

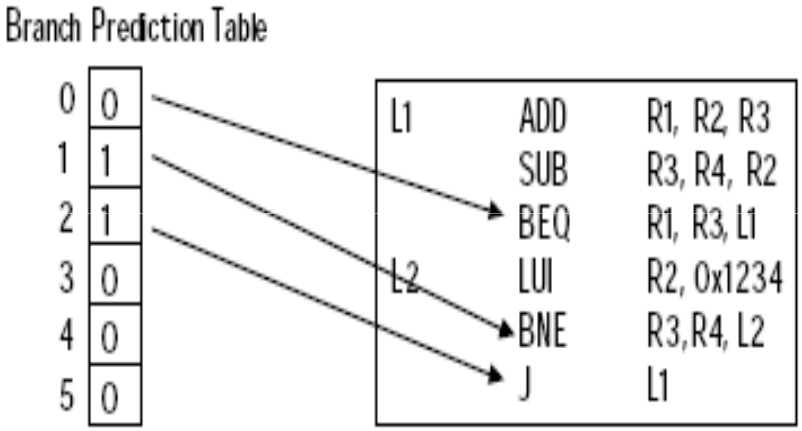
Predictores Dinámicos de Saltos

- Consideraciones
- Predicción de saltos
- Eliminación de saltos

José Luis Hamkalo
Depto. de Electrónica

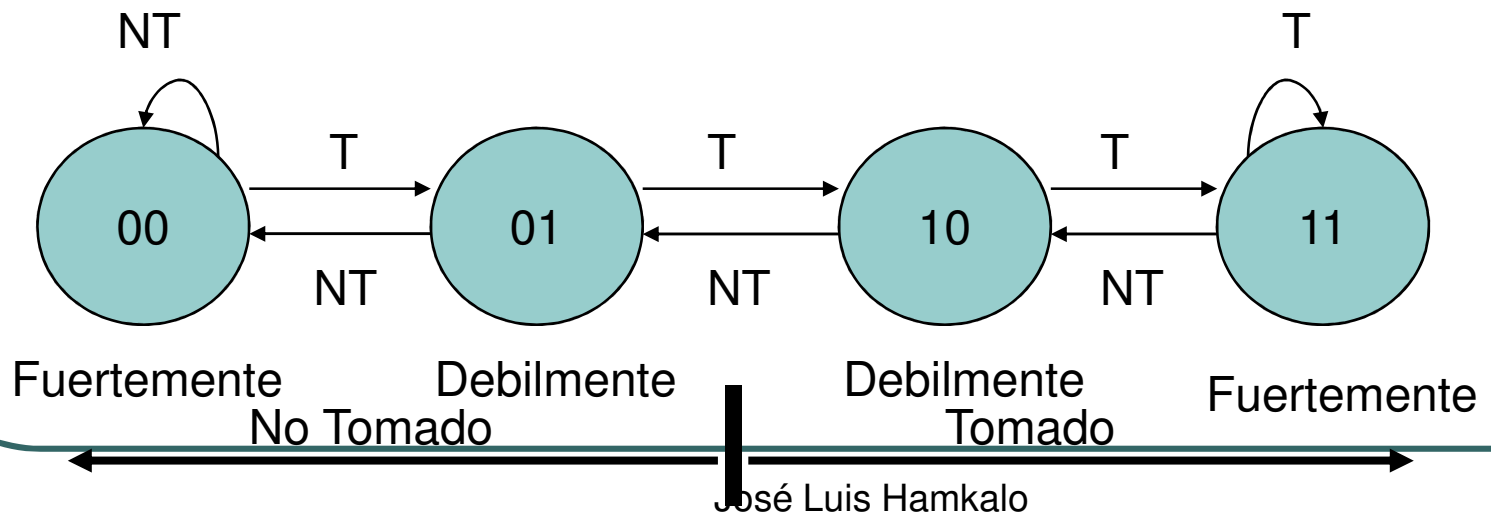
Predictor de Salto de un Bit

- Se implemneta mediante una tabla de 1 bit
- A cada salto le corresponde una entrada en la tabla
  - Si un bit es 1, entonces el salto se predice tomado
  - Si un bit es 0, entonces el salto se predice no tomado
- Cada bit de la tabla se va estableciendo dinámicamente: si un salto efectivamente se toma, entonces el bit correspondiente de la tabla se pone en 1, sino en 0.



Predictor de Saltos de 2 bits

- La tabla tiene 2 bits por entrada en lugar de 1
- Con 2 bits puede distinguir 4 estados posibles para predecir el salto.



Prediciendo el Resultado de Saltos en Programas de Punto Flotante

- Ejemplo

```
FOR I = 1 to N
      FOR J = 1 to N
            FOR K = 1 to N
            C(I,J) = C(I,J) + A(I,K) x B(K,J)
```

```
Li
    Lj
        Lk
         BNEZ Rk, Lk
         BNEZ Rj, Lj
         BNEZ Ri, Li
```

Each branch is taken (n-1) times and not taken 1 times

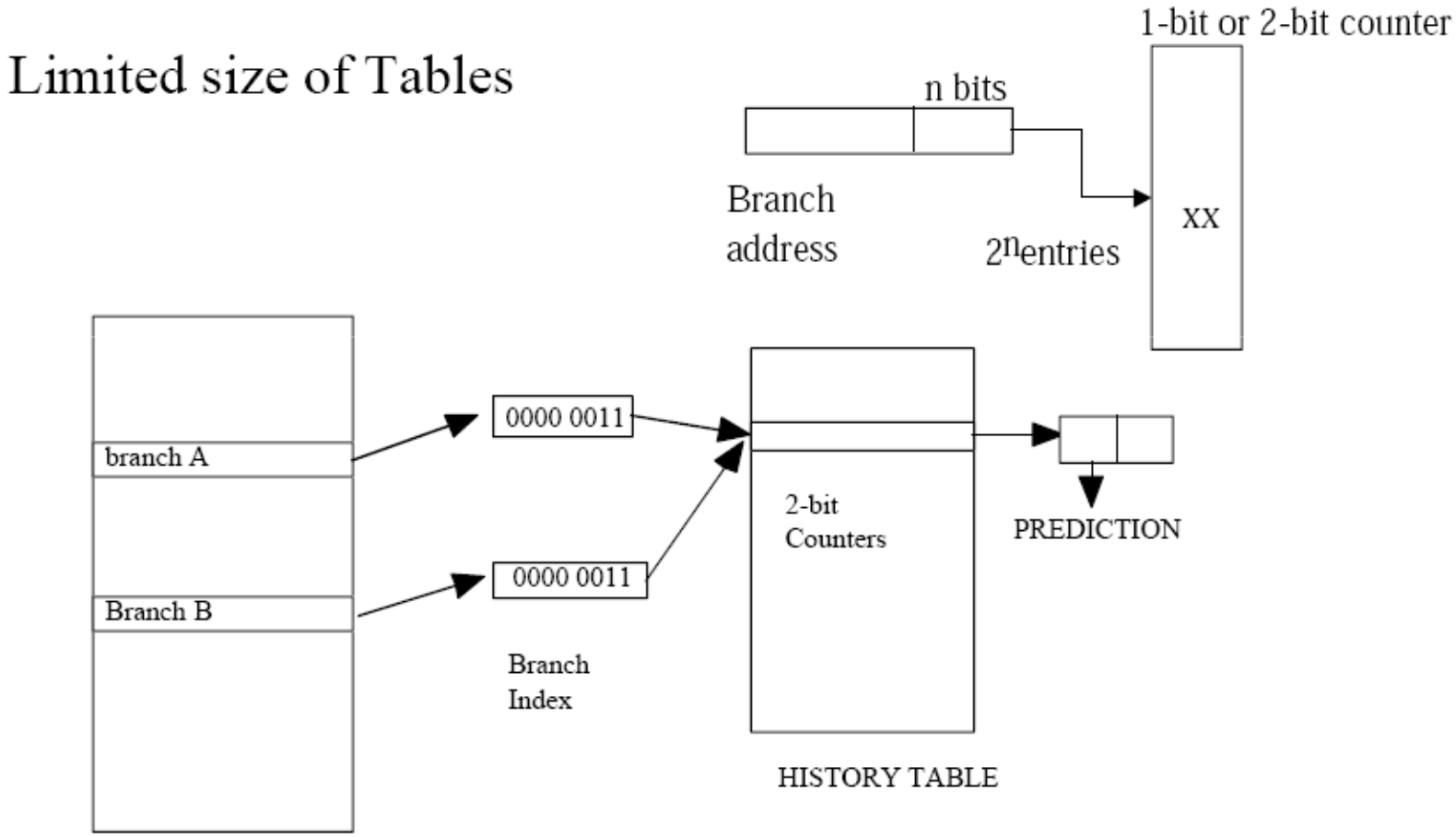
1-bit predictor: 2 mispredictions out of n predictions

2-bit predictor : 1 misprediction out of n predictions

Cómo Busco en la Tabla?

- Cada branch debería tener una entrada exclusiva en la tabla.
- Es deseable acceder por un mapeo de bits del PC (dirección del branch).
- En la práctica solo uso los “n” bits más bajos del PC para acceder a la tabla para limitar su tamaño.

Interferencia en Predictores



José Luis Hamkalo
Depto. de Electrónica

Interferencia (continuación)

- Interferencia en prdictores de 2 niveles.
  - Neutra
  - Positiva
  - Negativa

- Formas de minimizar la interferencia
  - Tablas de historia más grandes.
  - Idexado de tablas con mejor distribución
  - Predictores separados para distintos tipos de saltos

Saltos Correlacionados

- Example : eqntott (SPEC92)

```
if (aa==2)
        aa = 0 ;
if (bb==2)
        bb = 0 ;
if (aa != bb {
```

```
        SUBI    R3,R1,#2
 b1     BNEZ    R3,L1
        ADD     R1,R0,R0
L1  :   SUBI    R3,R2,#2
 b2     BNEZ    R3,L2
        ADD     R2,R0,R0
L2:     SUB     R3,R1,R2
 b3     BEQZ    R3,L3
```

b1 TAKEN if a ne 2

b2 TAKEN if b ne2

b1 and b2 NOT TAKEN (aa = bb = 0) => b3 TAKEN

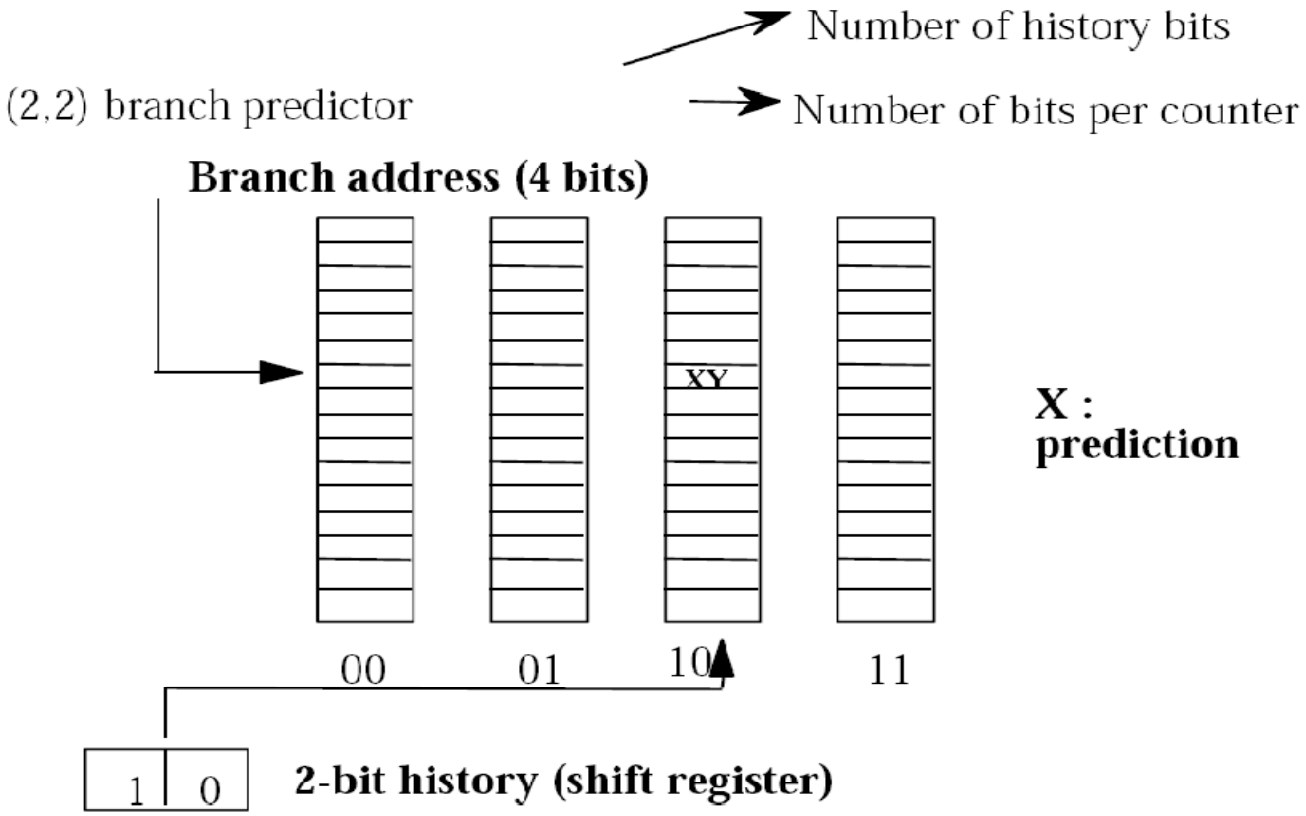
Correlation between behaviors of b1, b2 and b3

José Luis Hamkalo
Depto. de Electrónica

Predictor Basado en la Historia Global

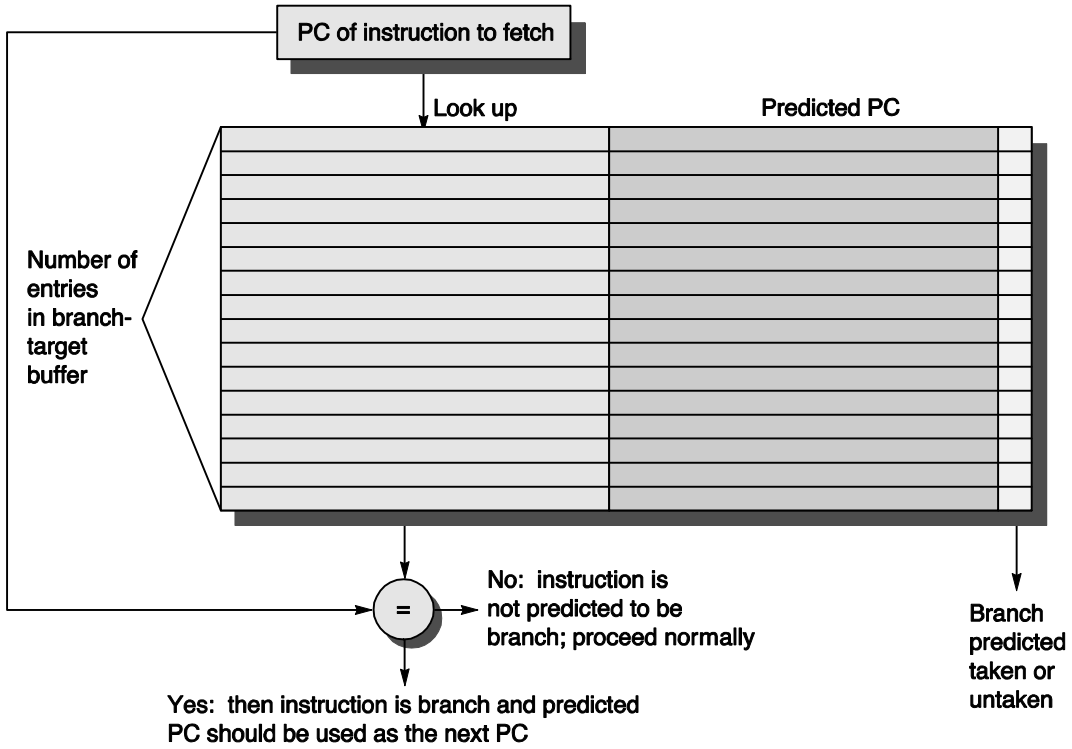
- Puedo crear una tabla de predicción indexada usando exclusivamente la historia global de todos los saltos.
- Se predice si un salto se va a tomar o no en base al resultado de los últimos “n” saltos dinámicamente ejecutados (de todo el programa).

Predictores de 2 niveles



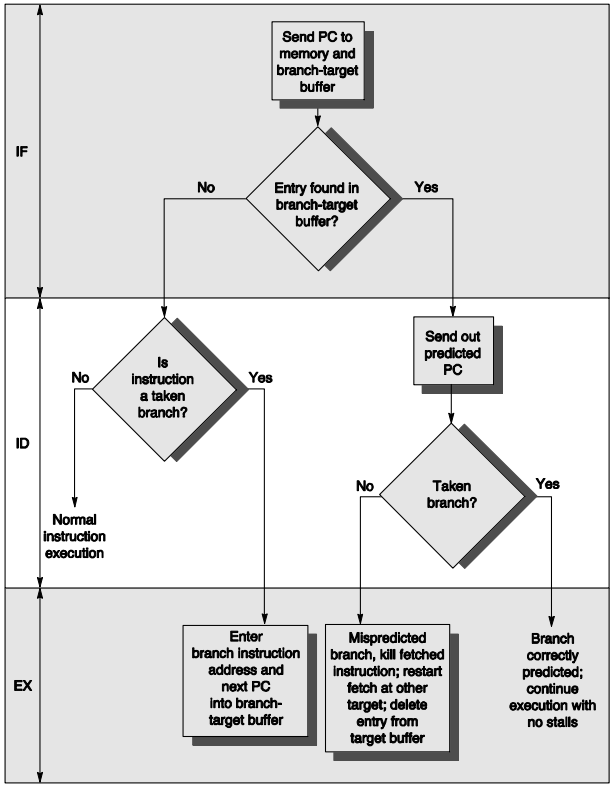
José Luis Hamkalo
Depto. de Electrónica

BTB (Branch Target Buffer)



José Luis Hamkalo
Depto. de Electrónica

BTB (continuación)



José Luis Hamkalo
Depto. de Electrónica

Eliminación de saltos

- Eliminación de saltos
  - Instrucciones condicionales
- Reducción del número de saltos
  - Desenrrollado de lazos

FIN

José Luis Hamkalo
Depto. de Electrónica