

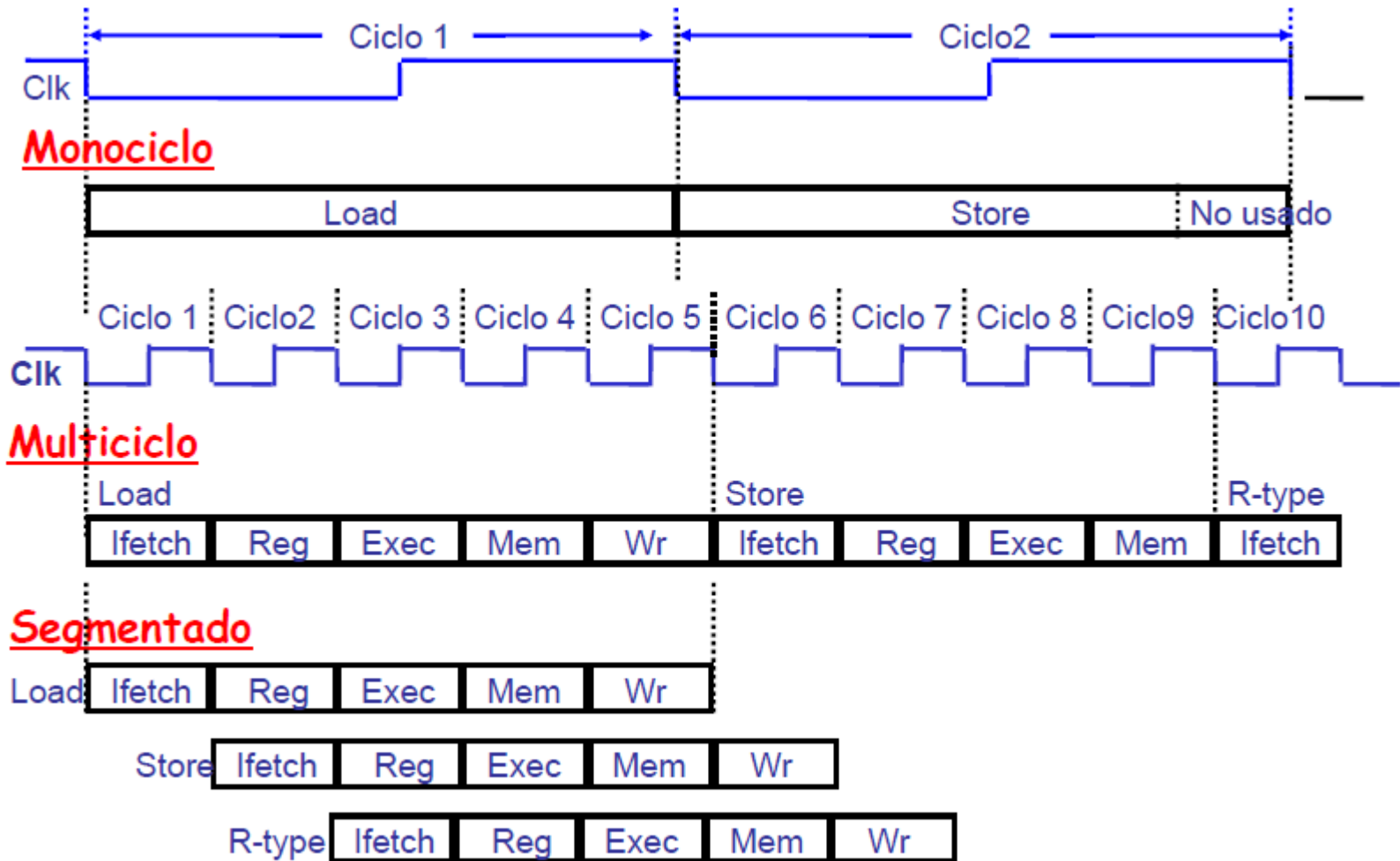
TRABAJO FINAL: PIPELINE PROCESADOR DLX (Familia MIPS)

Arquitectura de
Computadoras
2014

Consigna

- Implementar el pipeline del procesador MIPS

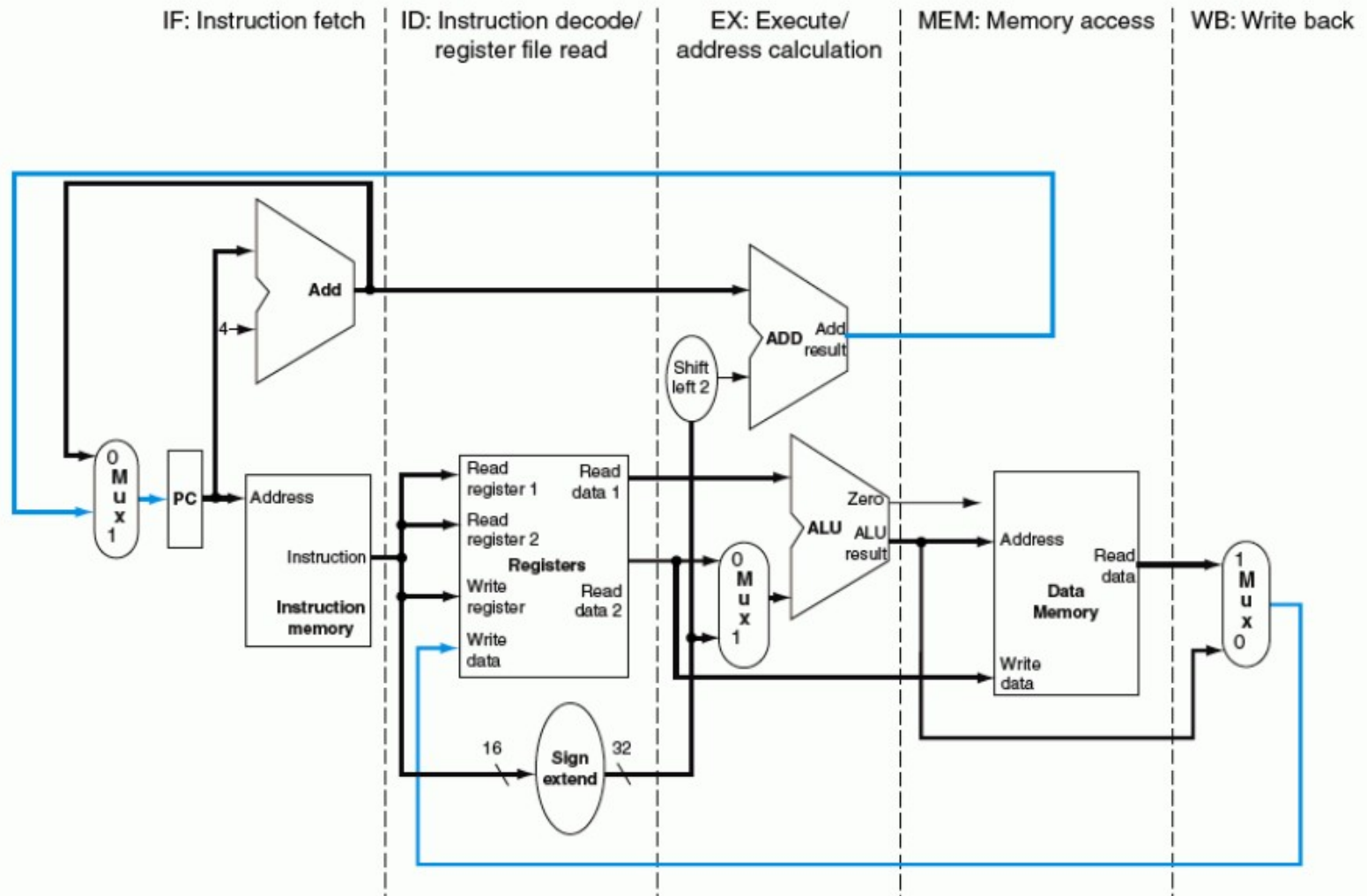
Marco teórico

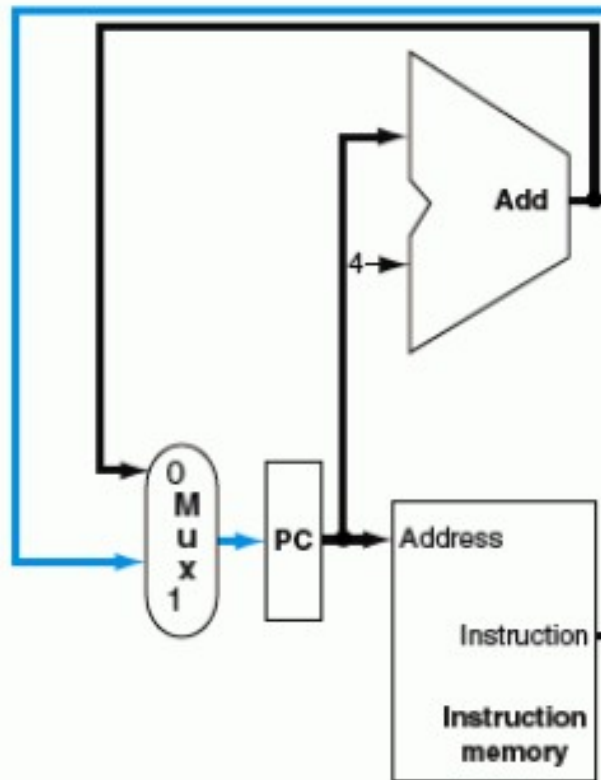


Etapas

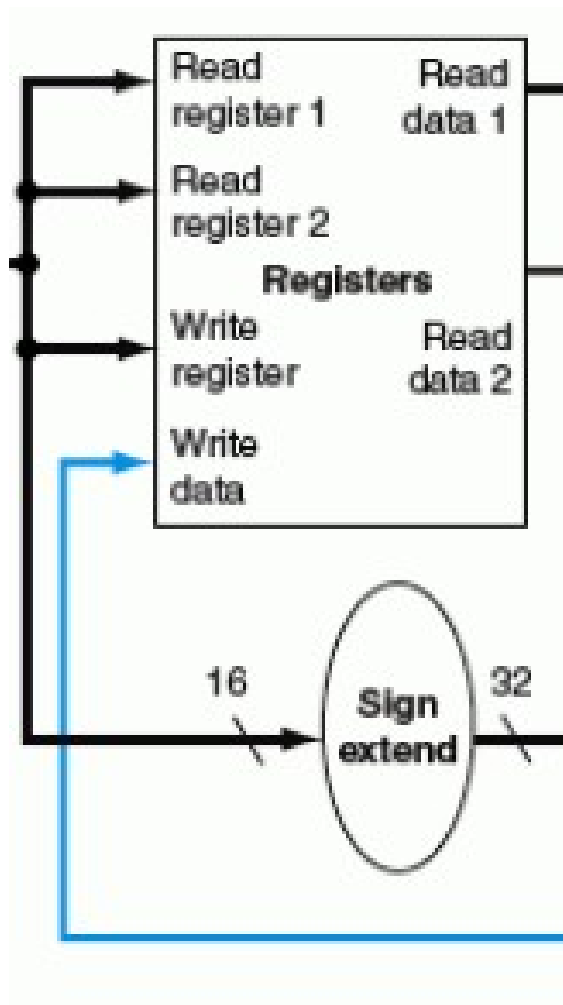
- **IF (Instruction Fetch):** Búsqueda de la instrucción en la memoria de programa.
- **ID (Instruction Decode):** Decodificación de la instrucción y lectura de registros.
- **EX (Execute):** Ejecución de la instrucción propiamente dicha.
- **MEM (Memory Access):** Lectura o escritura desde/hacia la memoria de datos.
- **WB (Write back):** Escritura de resultados en los registros.

Datapath

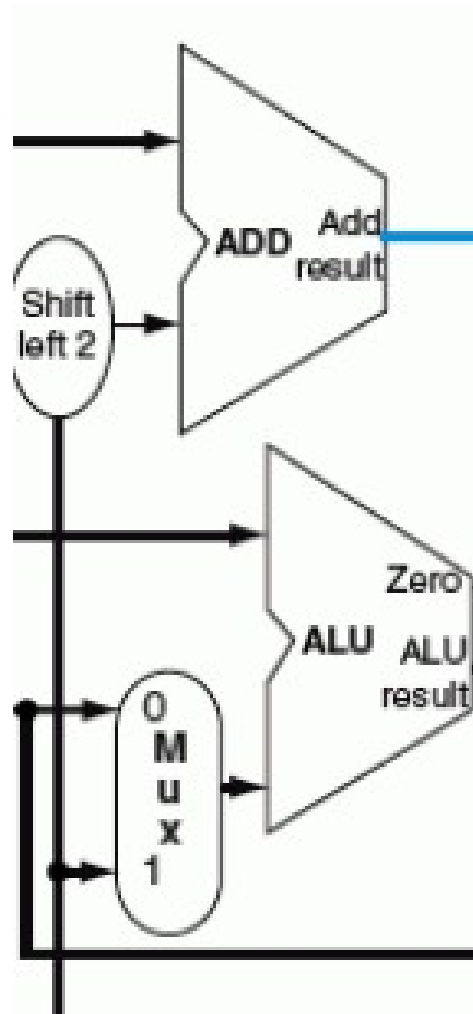




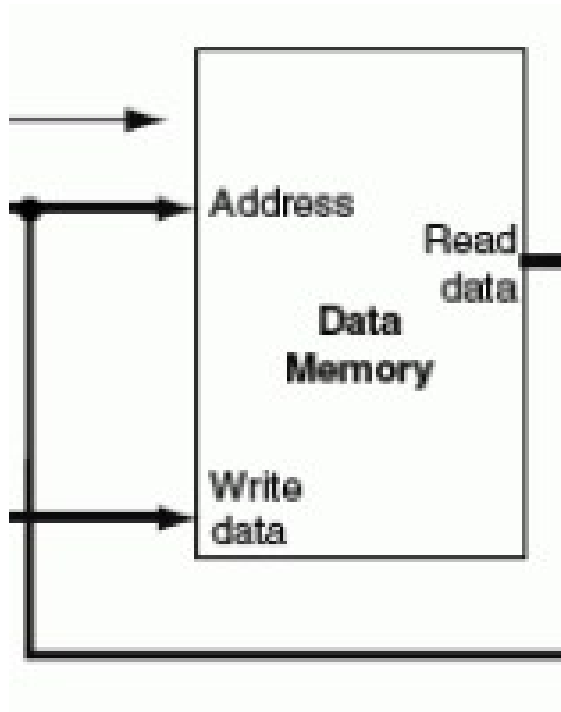
Instruction Fetch



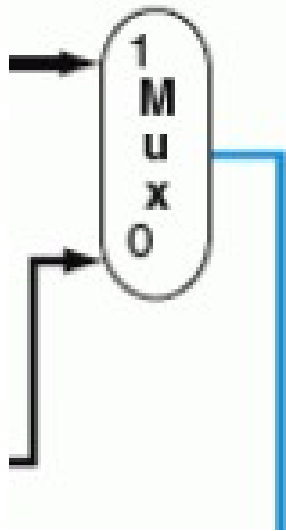
Instruction Decode



Execute

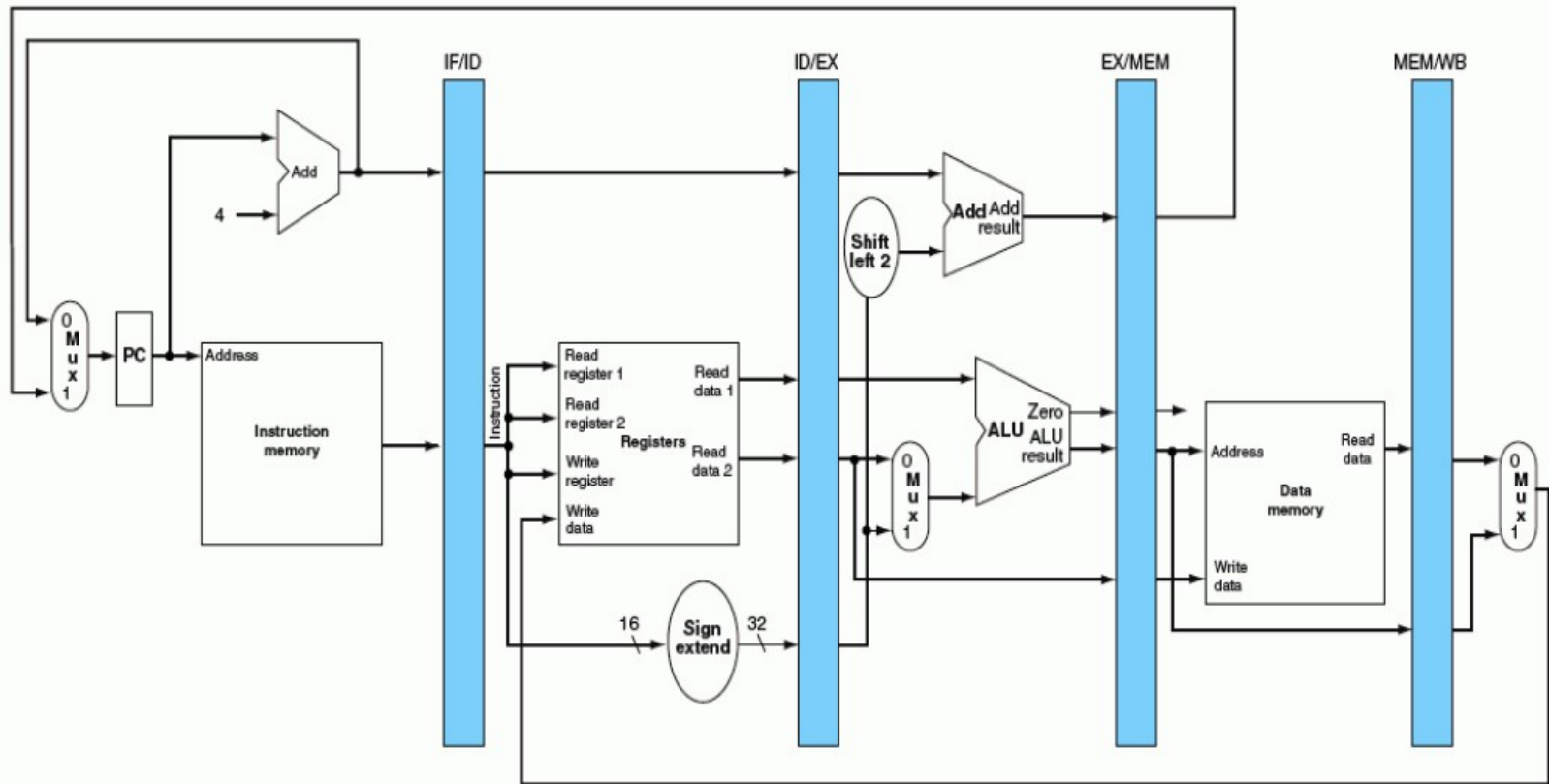


Memory access

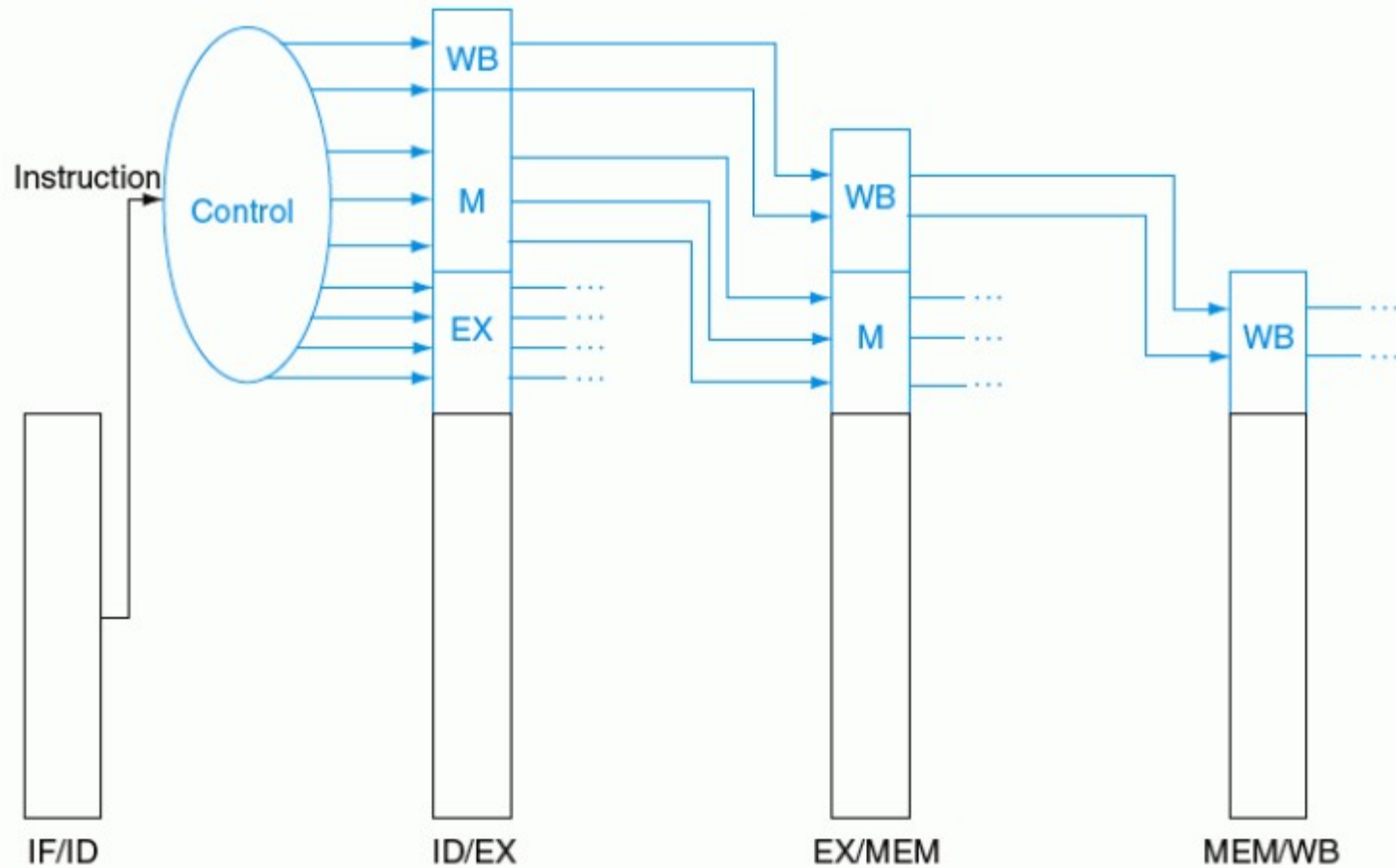


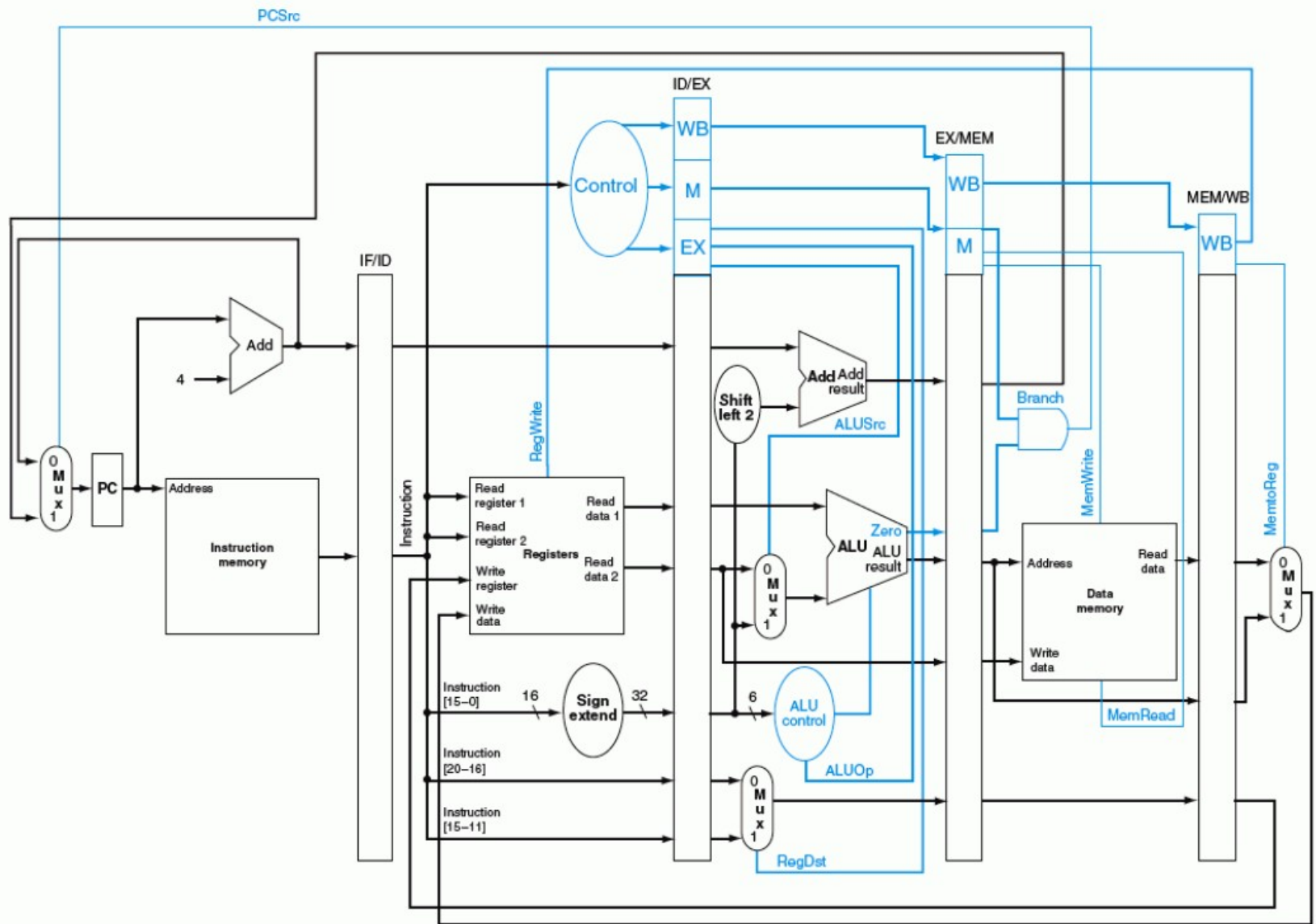
Write Back

Segmentación



Control Unit

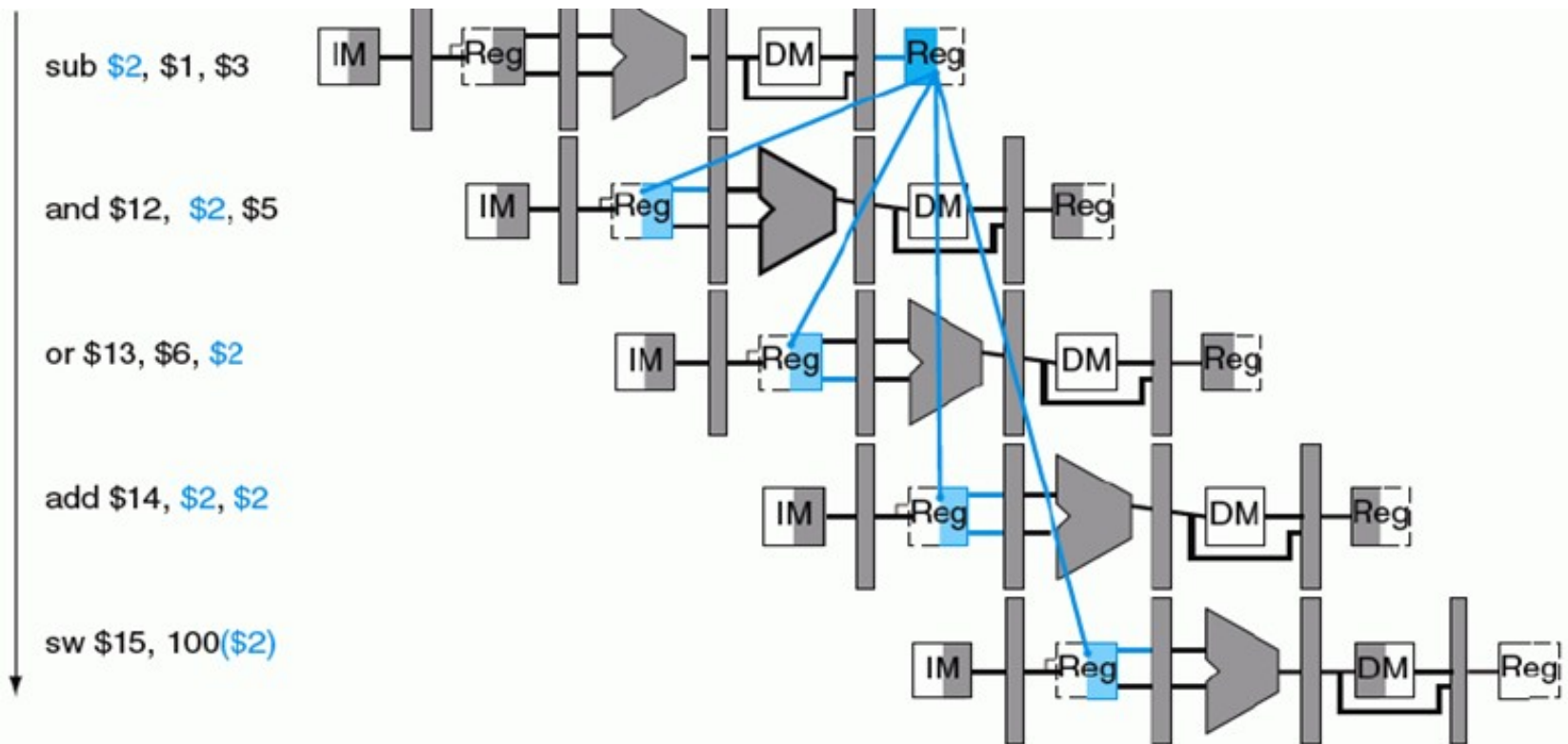




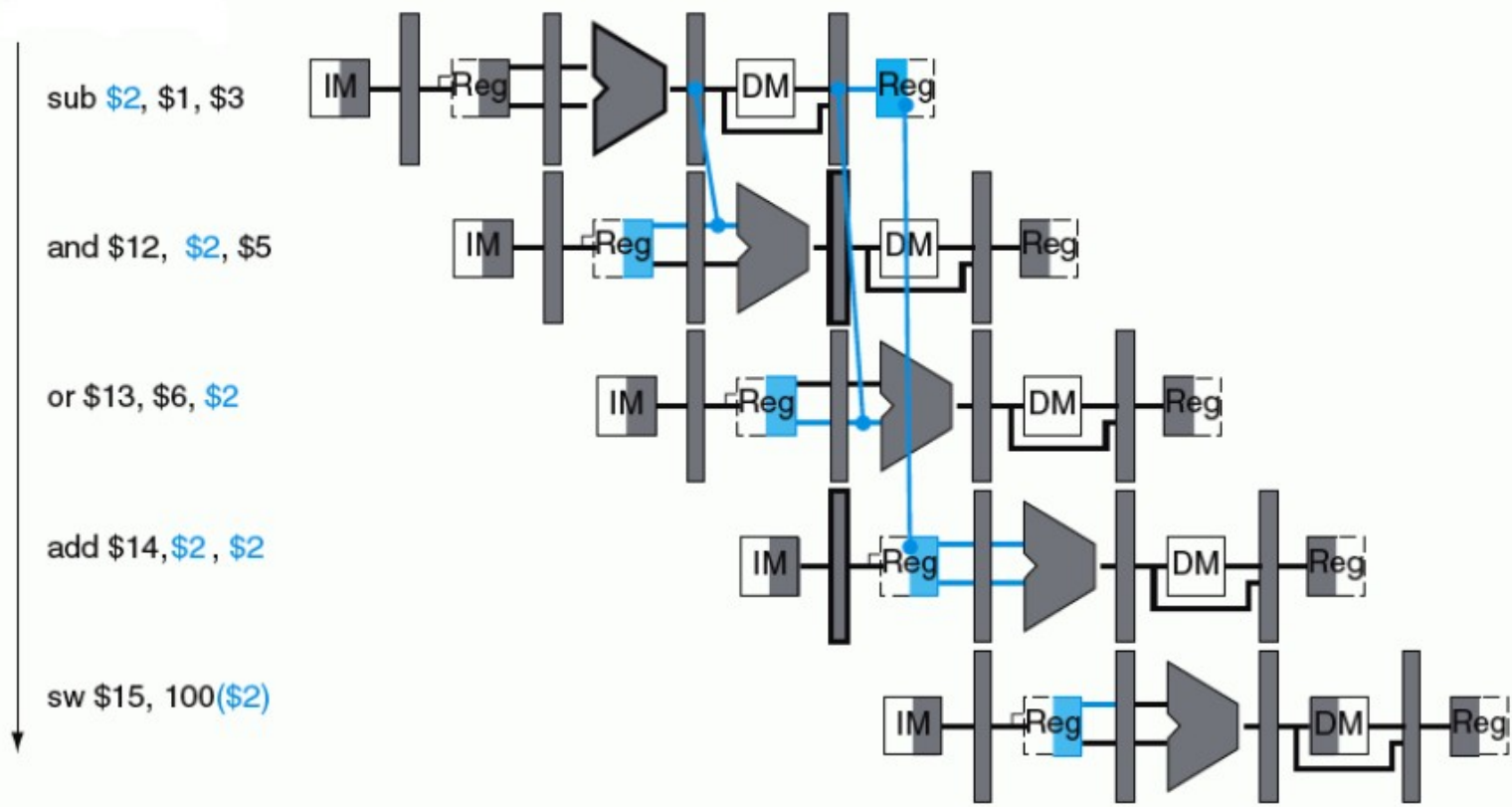
Riesgos

- Tipos:
 - Estructurales. Se producen cuando dos instrucciones tratan de utilizar el mismo recurso en el mismo ciclo.
 - De datos. Se intenta utilizar un dato antes de que este preparado. Mantenimiento del orden estricto de lecturas y escrituras.
 - De control. Intentar tomar una decisión sobre una condición todavía no evaluada.

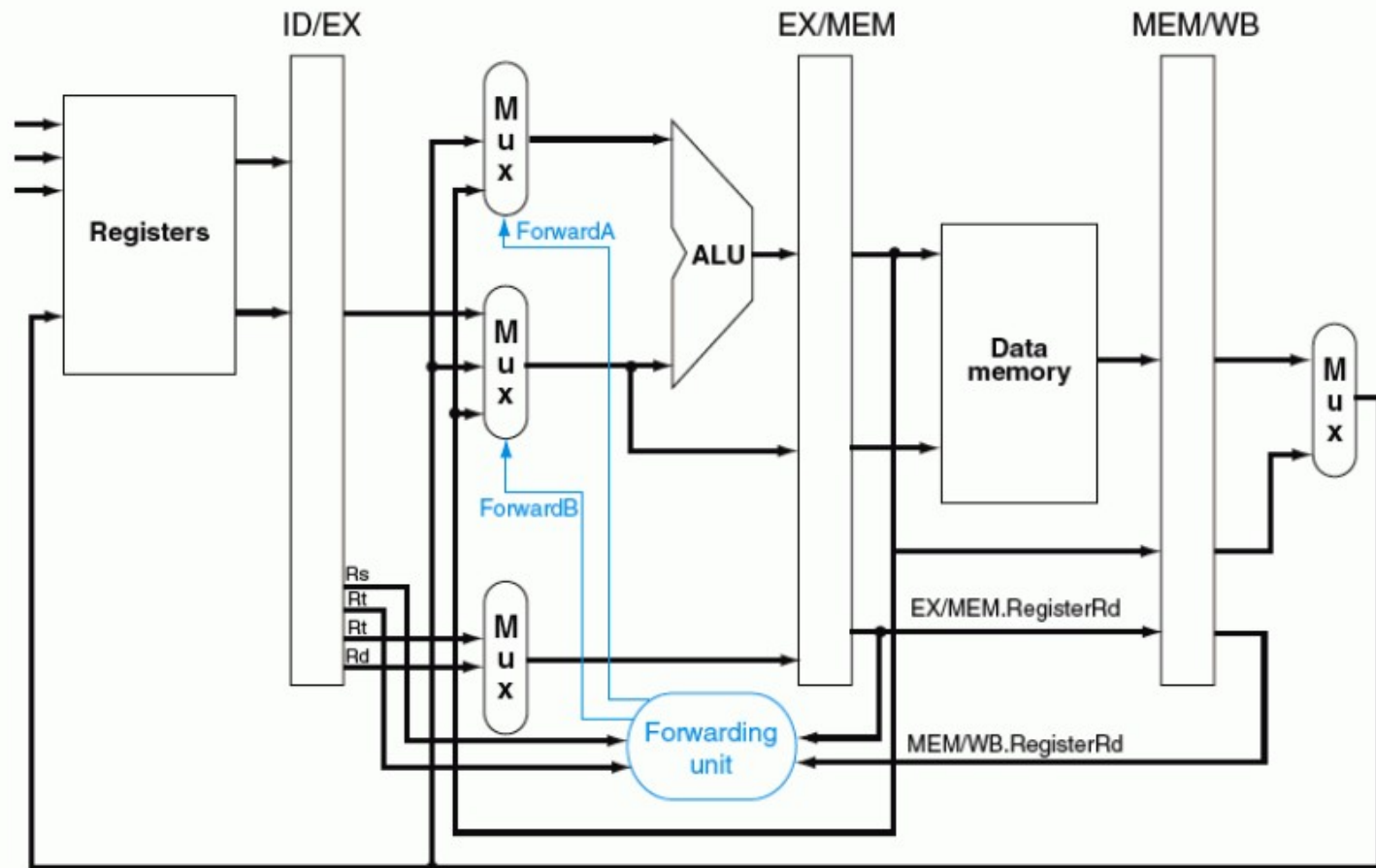
Riesgos de datos Dependencias de registros



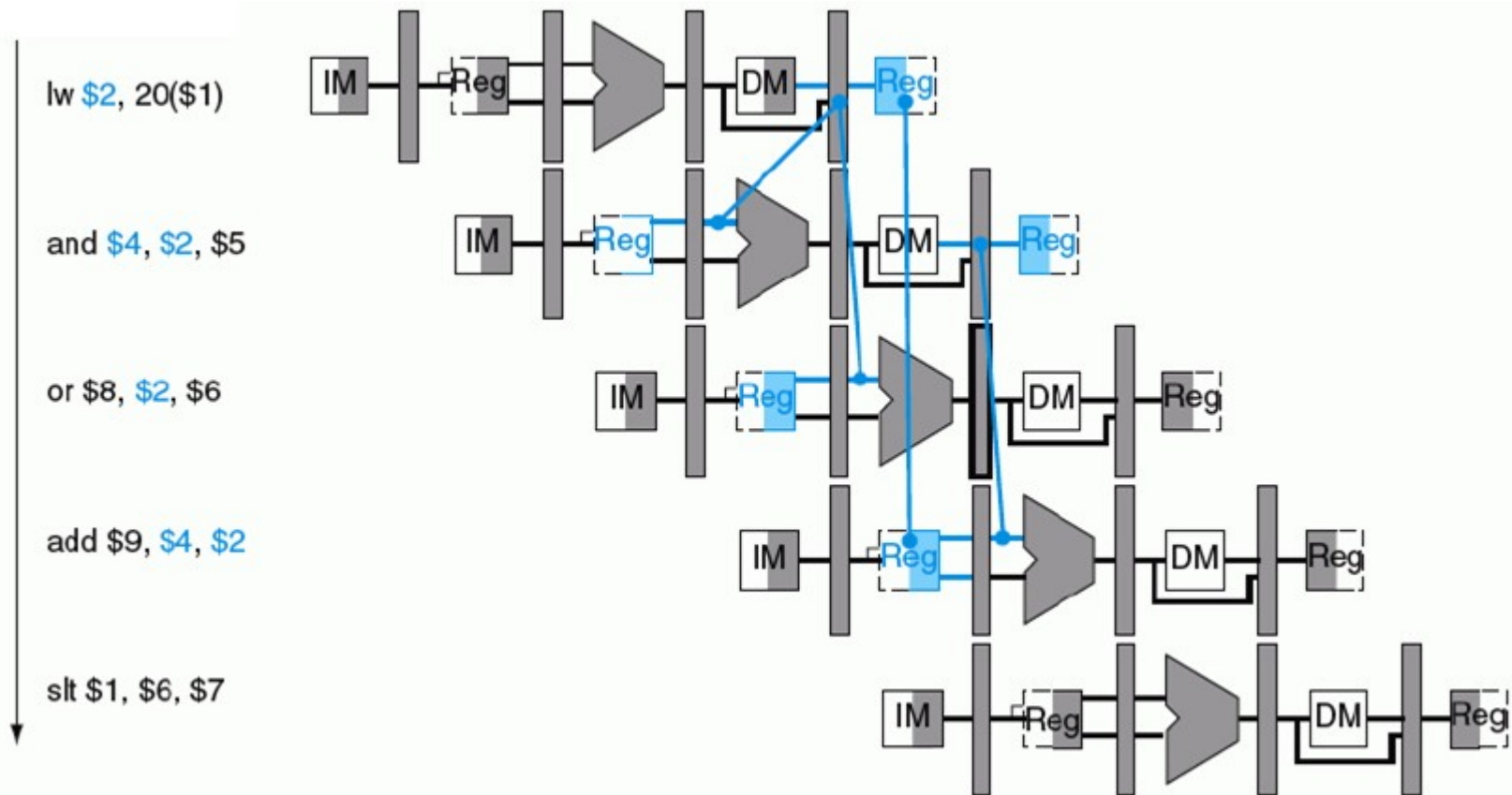
Solución



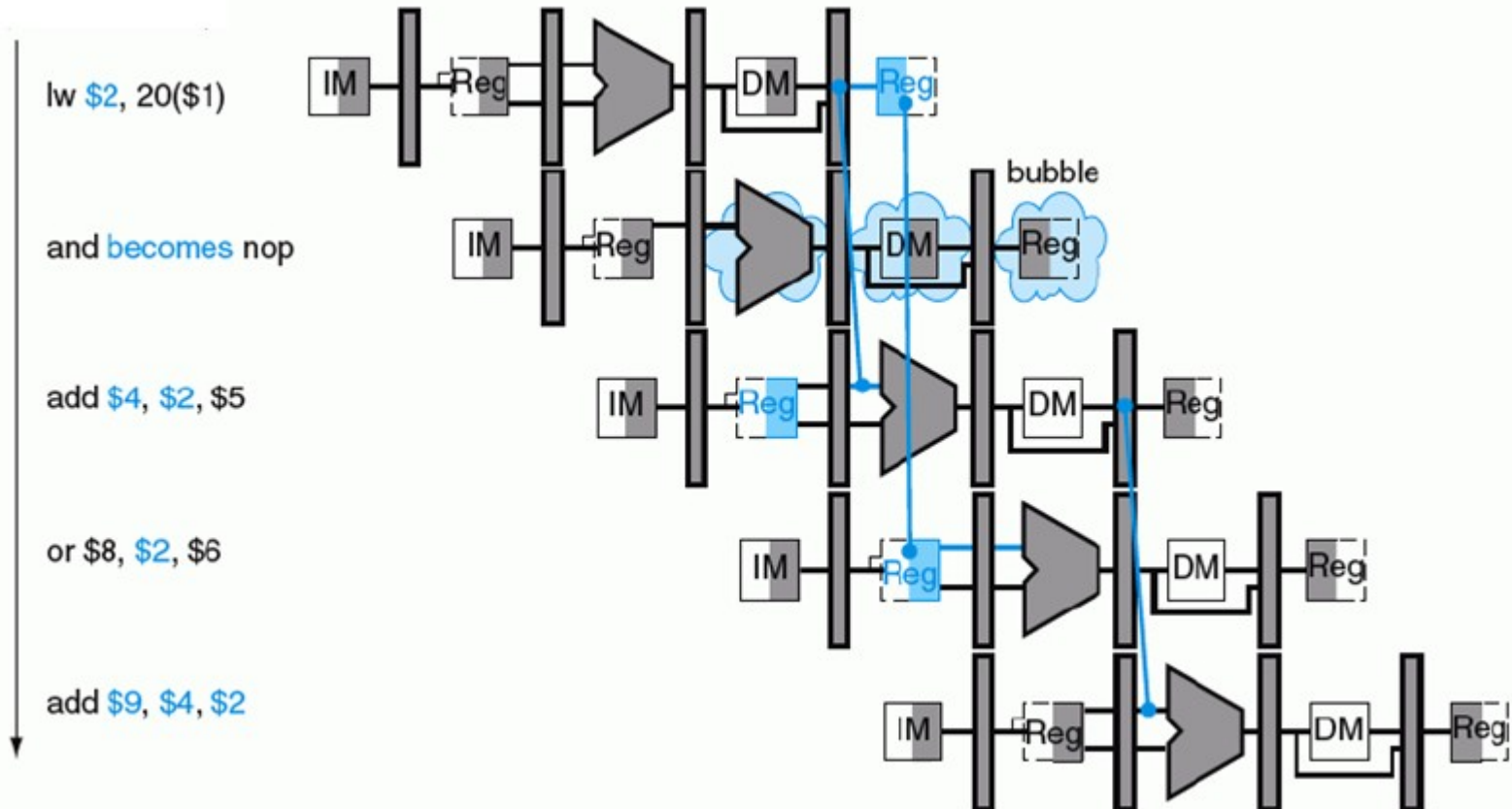
Unidad de cortocircuitos



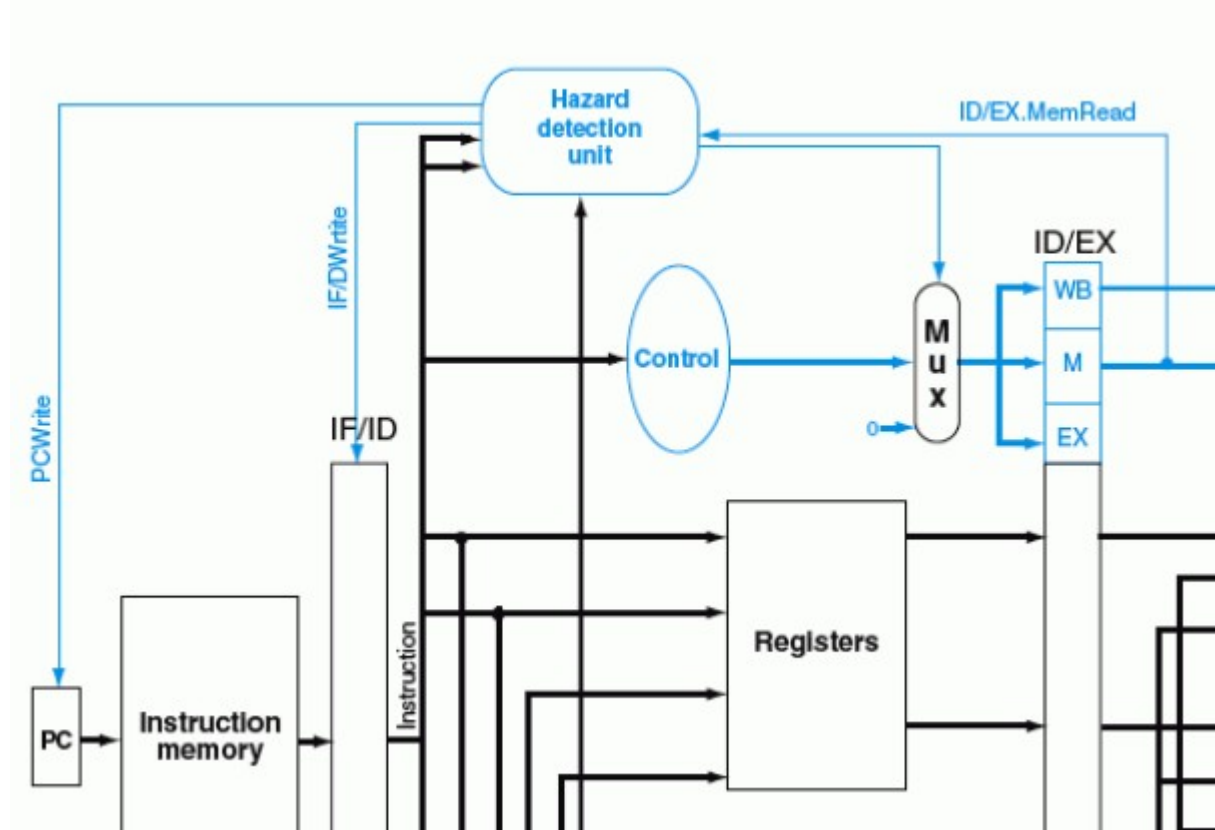
Riesgos de datos Dependencias de datos



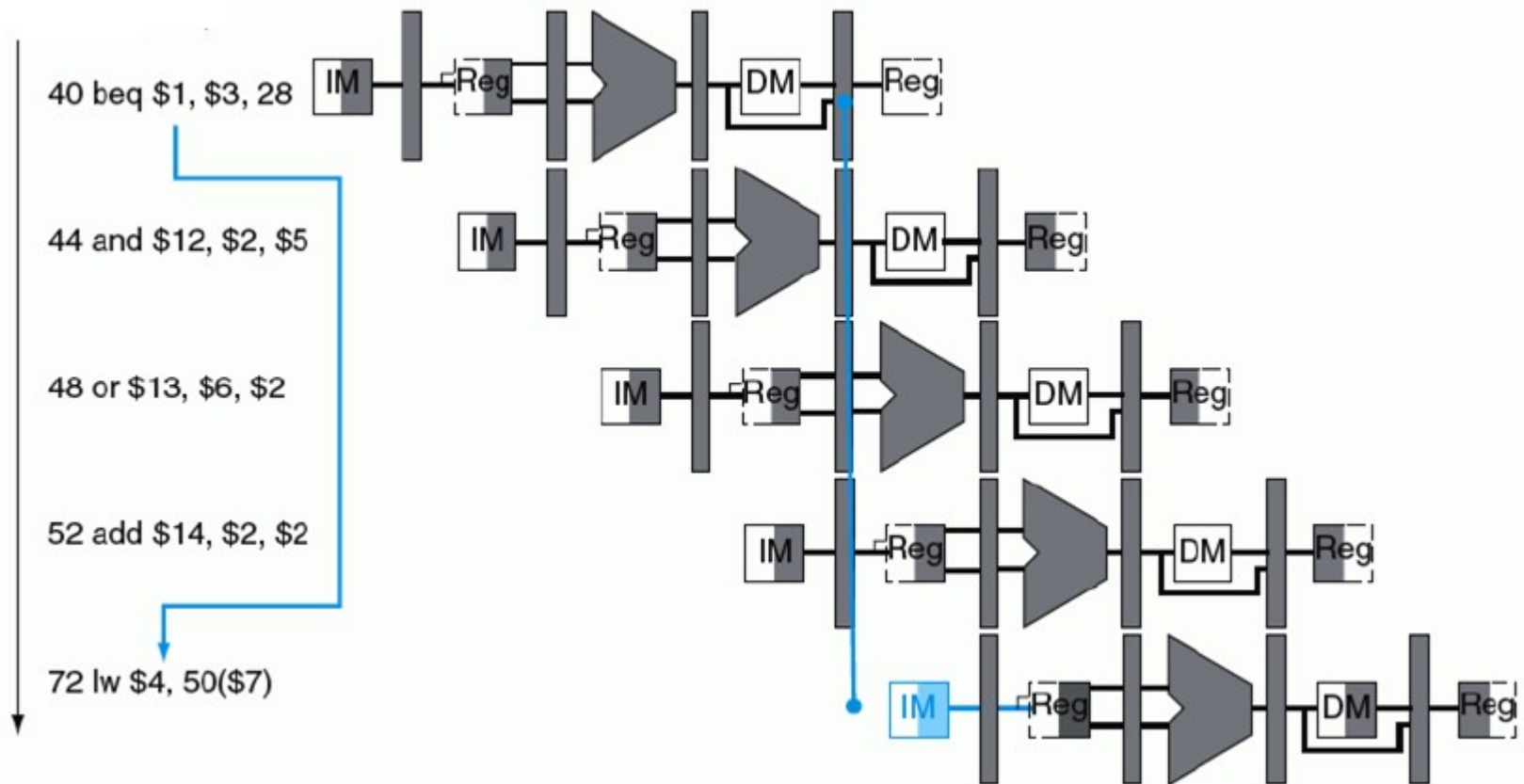
Solución



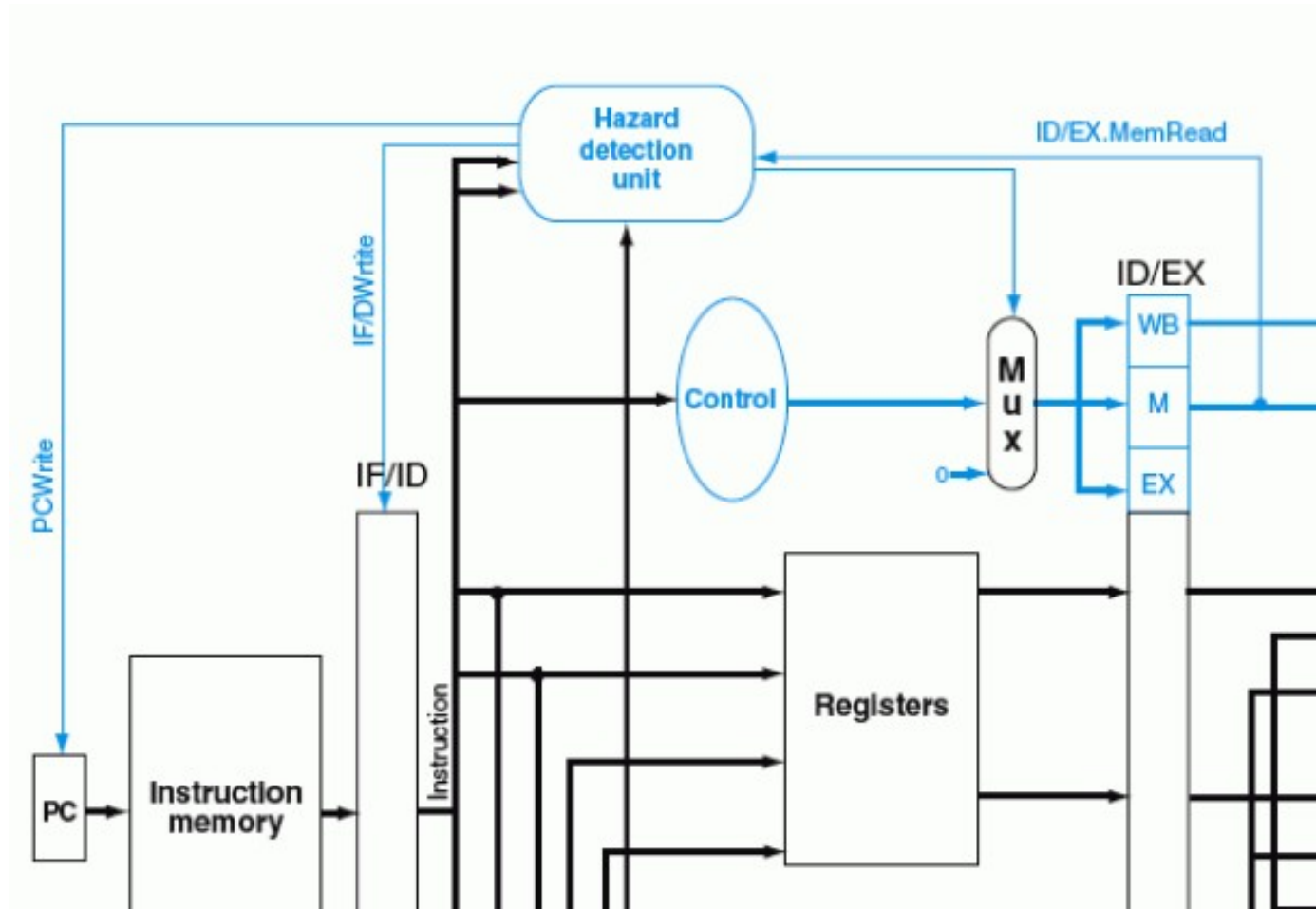
Unidad de detección de riesgos



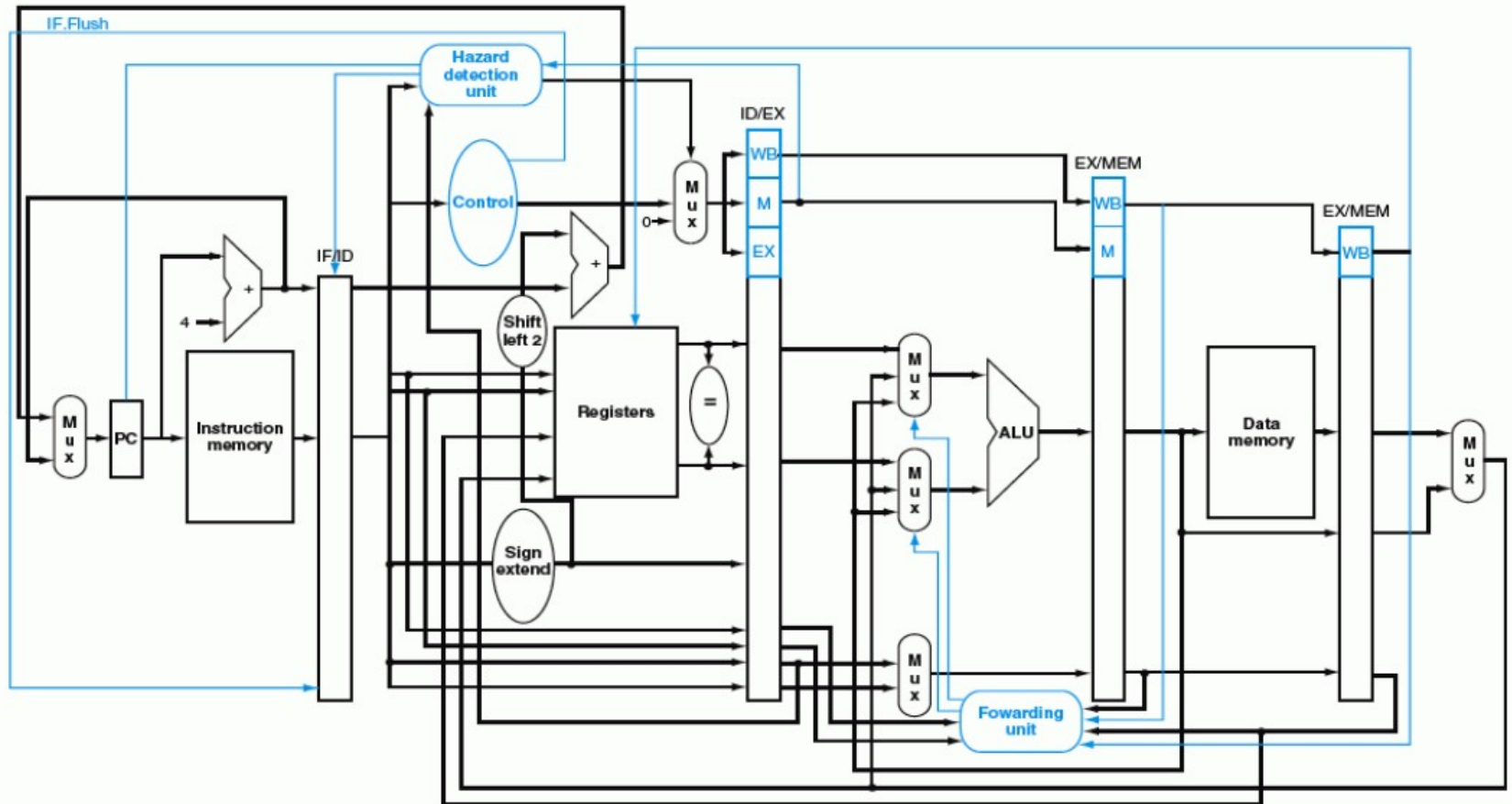
Riesgos de control



Solución



Pipeline



Tipo de Instrucciones

○ Tipo R

- Son operaciones aritméticas y lógicas
- OP Code = 000000
- Indican la operación en los 6 bits menos significativos



Tipo de Instrucciones

○ Tipo I

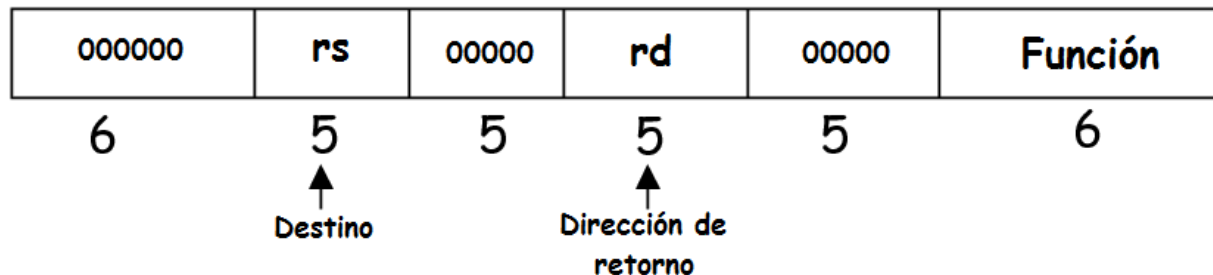
- Operaciones con un registro y un valor inmediato
- En operaciones lógicas y aritméticas se opera con «rs» y se almacena el resultado en «rt».
- En load y store «rs» es la base y el inmediato es el offset.



Tipo de Instrucciones

○ Tipo J

- Operaciones de salto incondicional
- La dirección a la que se salta es la almacenada en el registro «rs».





Requerimientos

Instrucciones a implementar

- R-type

SLL, SRL, SRA, SLLV, SRLV, SRAV

ADDU, SUBU

AND, OR, XOR, NOR

SLT, SLTU

- I-Type

LB, LH, LW, LWU, LBU, LHU, SB, SH, SW

ADDI, ADDIU, ANDI, ORI, XORI, LUI

SLTI, SLTIU, BEQ, BNE

J, JAL

- J-Type

JR, JALR

Otros requerimientos

- La memoria de datos debe estar separada de la memoria de instrucciones. Ambas deben ser implementadas con IP Cores
- El programa a ejecutar debe ser cargado en la memoria de programa mediante un archivo «.coe»
- Se debe incluir una unidad de debug que envíe información a la pc mediante la uart.

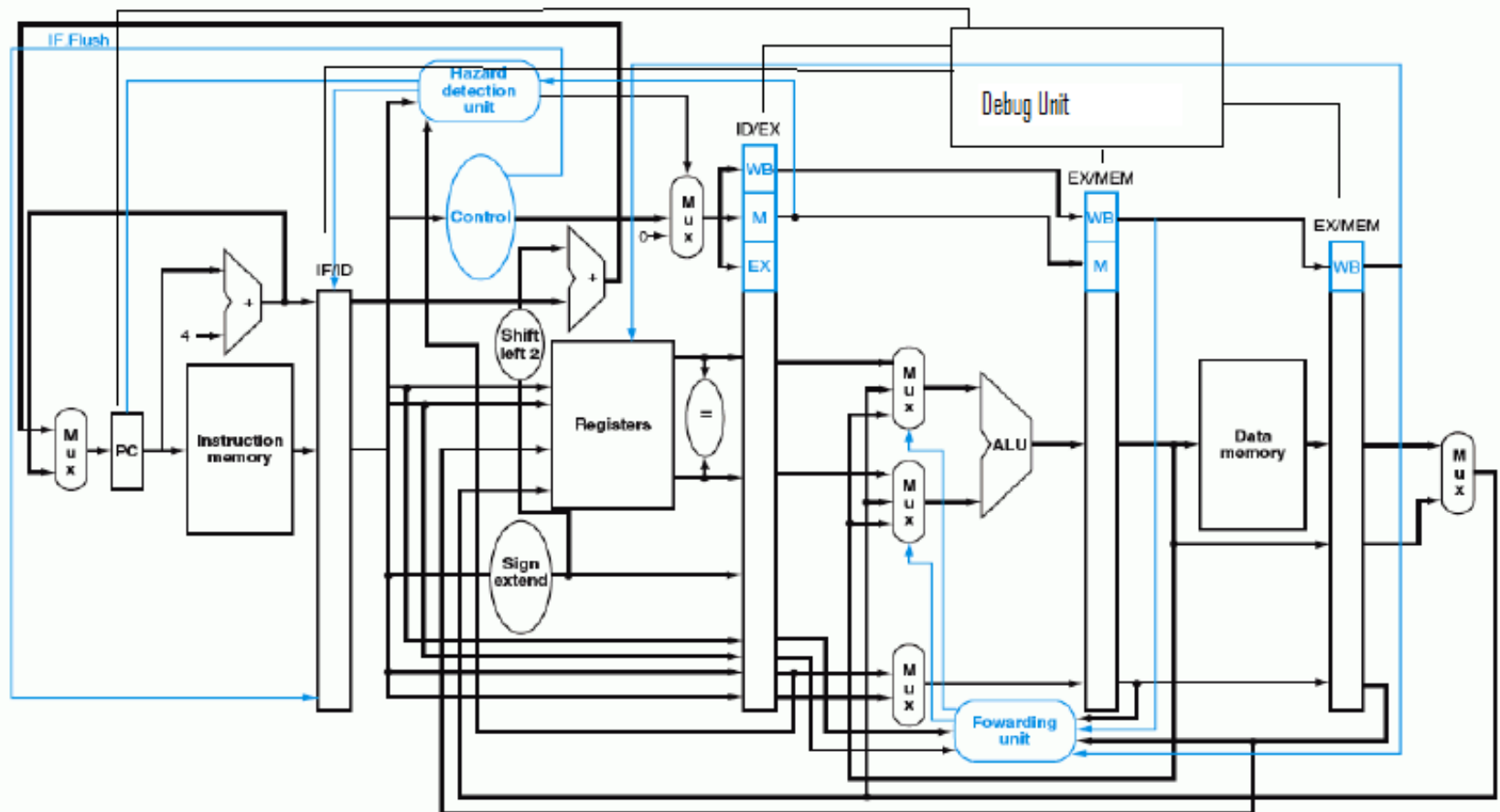
Debug unit

- Se deben enviar a la PC a través de la uart:
 - El contenido de los 32 registros.
 - El contenido de los latches intermedios.
 - Contenido de la memoria de datos usada.

Modos de operación

- Debe permitir dos modos de operación:
 - Continuo, se envía un comando a la fpga por la uart y esta inicia la ejecución del programa hasta llegar al final del mismo. Llegado ese punto se muestran todos los valores indicados en pantalla.
 - Paso a paso: Enviando un comando por la uart se ejecuta un ciclo de clock. Se debe mostrar a cada paso los valores indicados.

Pipeline final



Bibliografía

- *Instrucciones:*
 - *MIPS IV Instruction Set*
- *Pipeline:*
 - Computer Organization and Design 3rd Edition. Chapter 6. Hennessy- Patterson