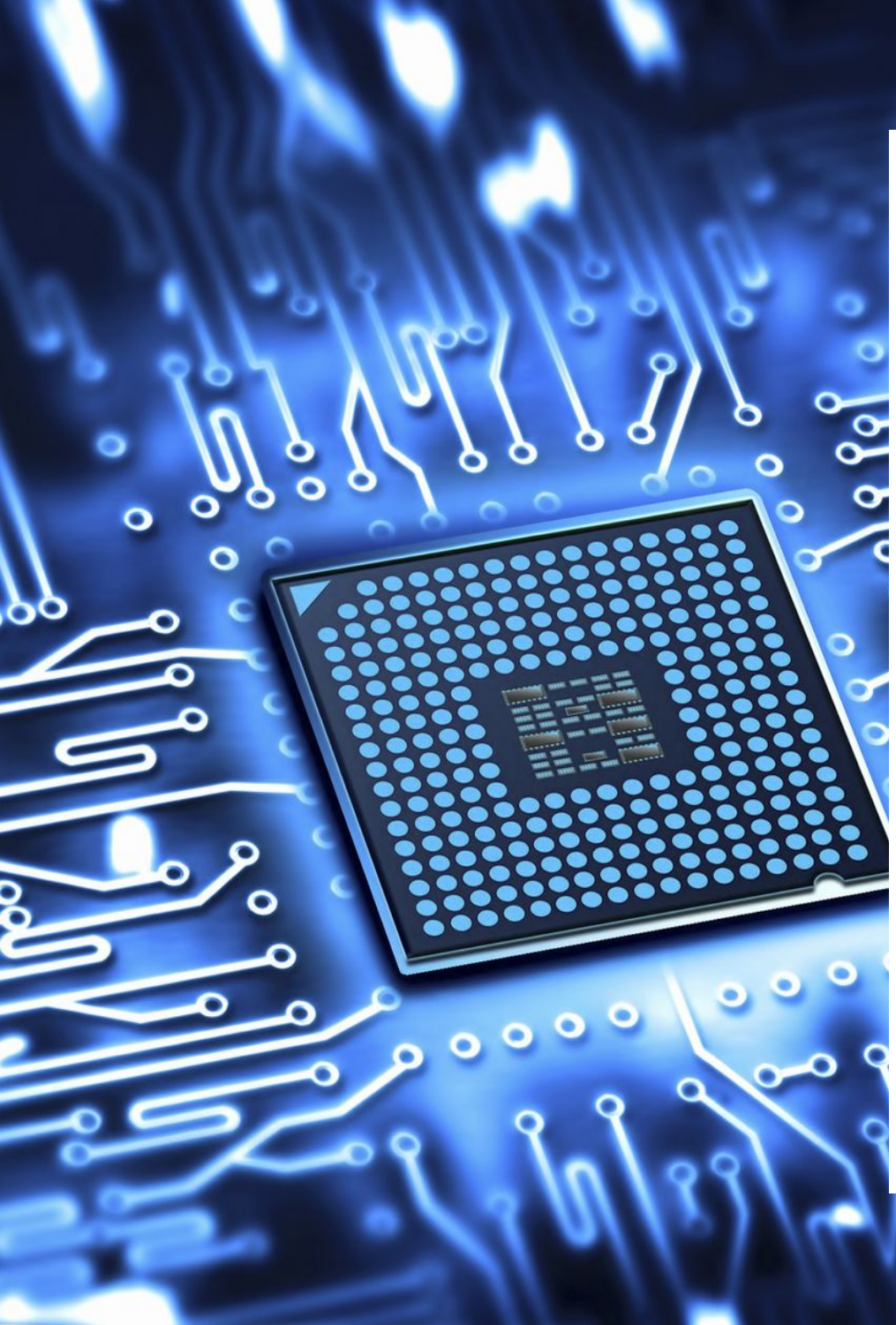


Arquitectura de computadoras

Paralelismo a nivel de instrucciones

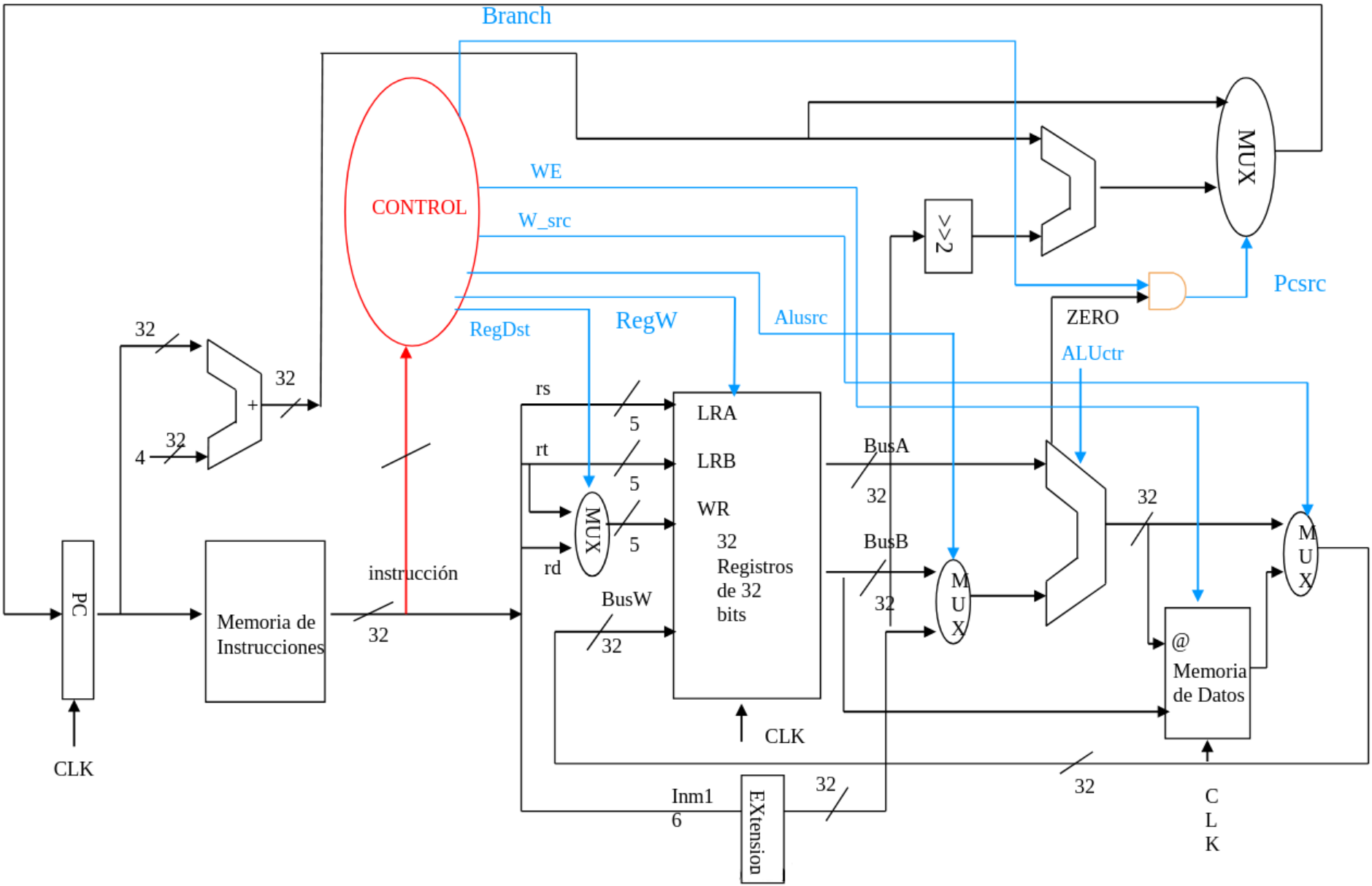
Universidad Nacional de la Patagonia

Procesador monociclo

- Cada instrucción se ejecuta en un único ciclo de reloj
- CPI=1 para todas las instrucciones
- El período de reloj debe ser el de la instrucción más costosa: camino crítico
- Diseño poco flexible, pero fácil de comprender! Y de implementar!!

Conjunto de instrucciones

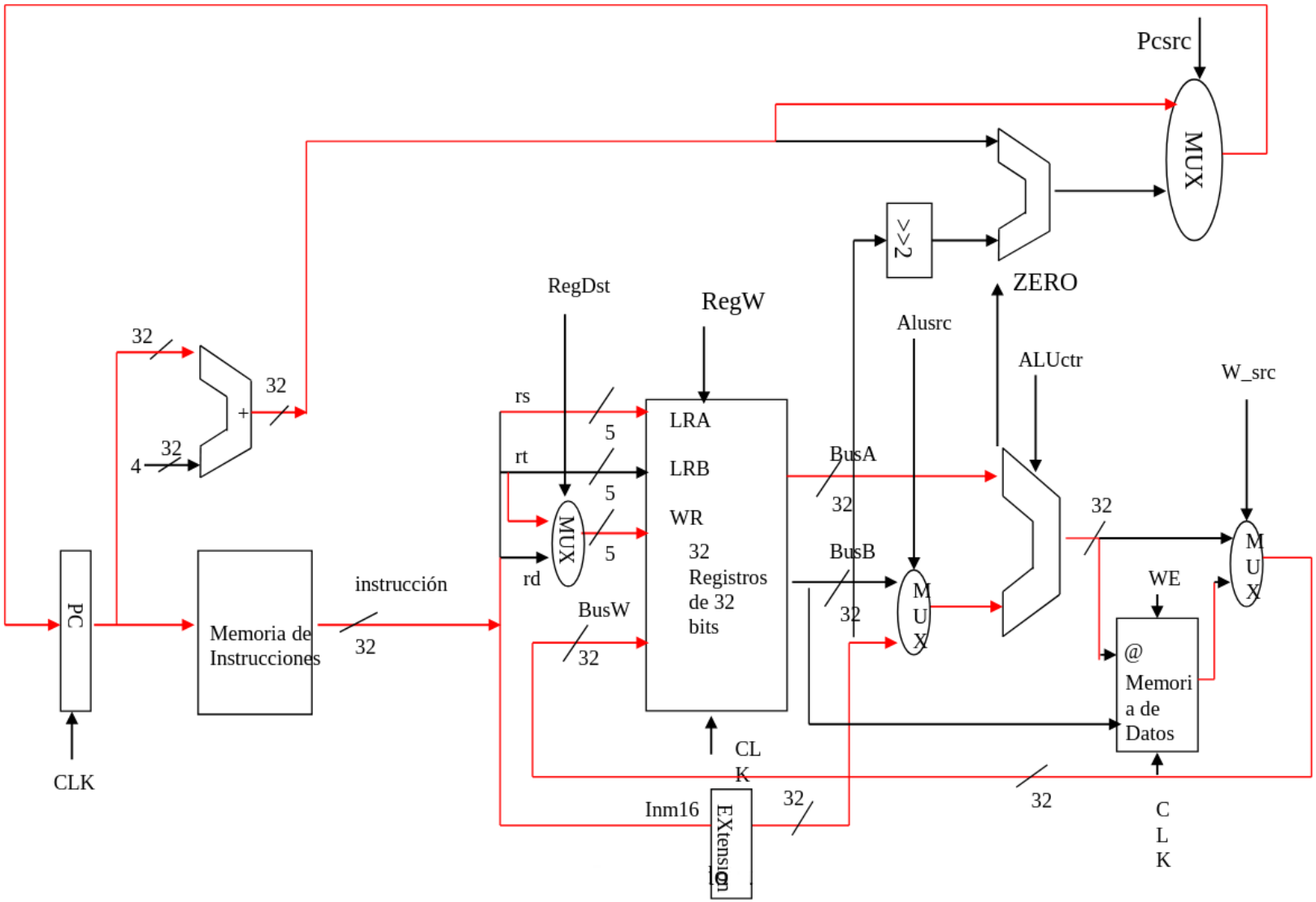
- Add y sub
  - addu rd ,rs ,rt
  - subu rd, rs ,rt
- OR inmediato
  - ori rt, rs, inm16
- Load y Store
  - lw rt, rs, inm16
  - sw rt, rs, inm16
- Branch
  - beq rs, rt, inm16


Branch
CONTROL
WE
W_src
RegDst
RegW
Alusrc
ALUctr
ZERO
Pcsrc
MUX
<<2
32
32
4
+
PC
CLK
Memoria de Instrucciones
instrucción
32
rs
rt
rd
5
5
5
MUX
LRA
LRB
WR
32 Registros de 32 bits
BusW
32
CLK
BusA
BusB
32
32
M U X
32
@ Memoria de Datos
M U X
32
C L K
Inm1
6
EXtension
32

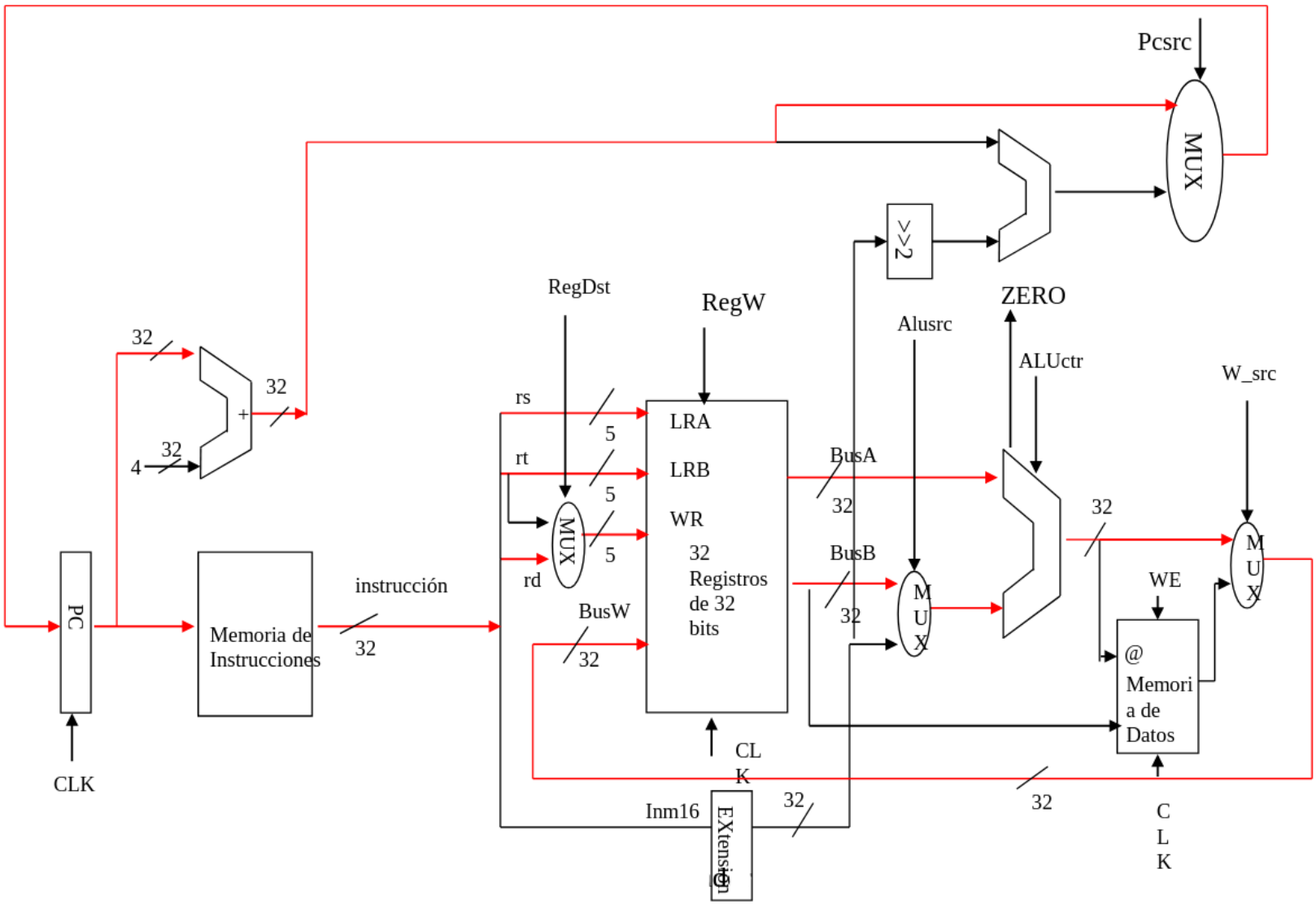
Señales de control

	No activa	Activa
RegDst	Destino es rt	Destino es rd
ALUsrc	Operando proviene del banco del registros	Operando son los 16 bits de la instrucción con el signo extendido
W_src	El valor que se escribe en el banco de registros provienen de la ALU	El valor que se escribe en el banco de registros proviene de memoria
RegW	Nada	Se escribe el valor en el registro especificado
We	Nada	Se escribe en la memoria el valor que esta en su entrada
PC_src	$PC = PC + 4$	PC=destino de salto

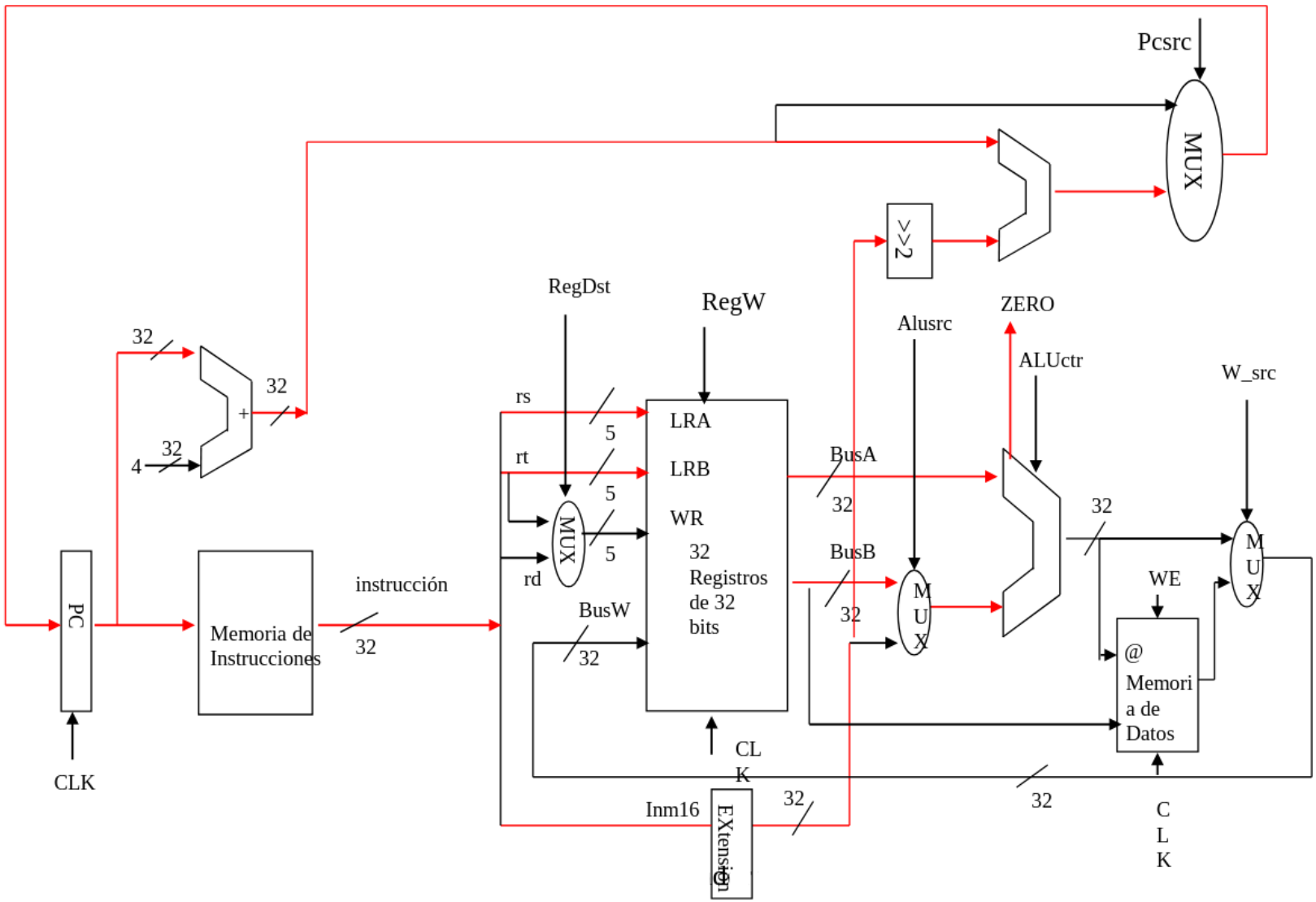
Ejemplo de Load



Ejemplo de Addu



Ejemplo de Beq (realizado)



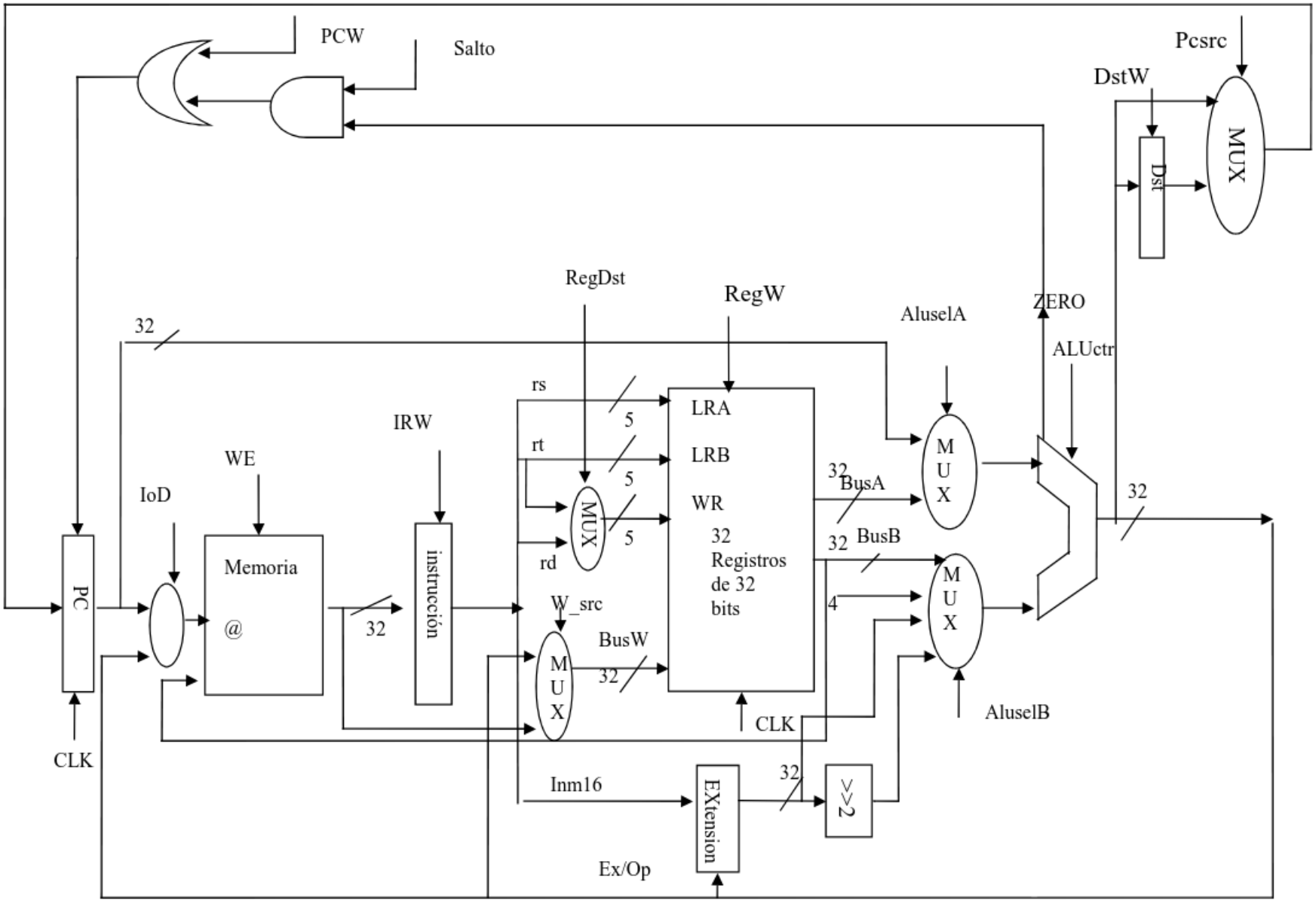
Monociclo vs Multiciclo

- Diseño monociclo
  - Todas las instrucciones tardan tanto como la más lenta
  - Ciclo de reloj es largo, pero CPI=1
  - No hay reuso de unidades funcionales
- Diseño multiciclo
  - Reduce el tiempo de ciclo
  - Un ciclo de reloj=una etapa de ejecución
  - Reuso de Unidades Funcionales
  - Cada instrucción puede utilizar o no todas las etapas, el CPI > 1.

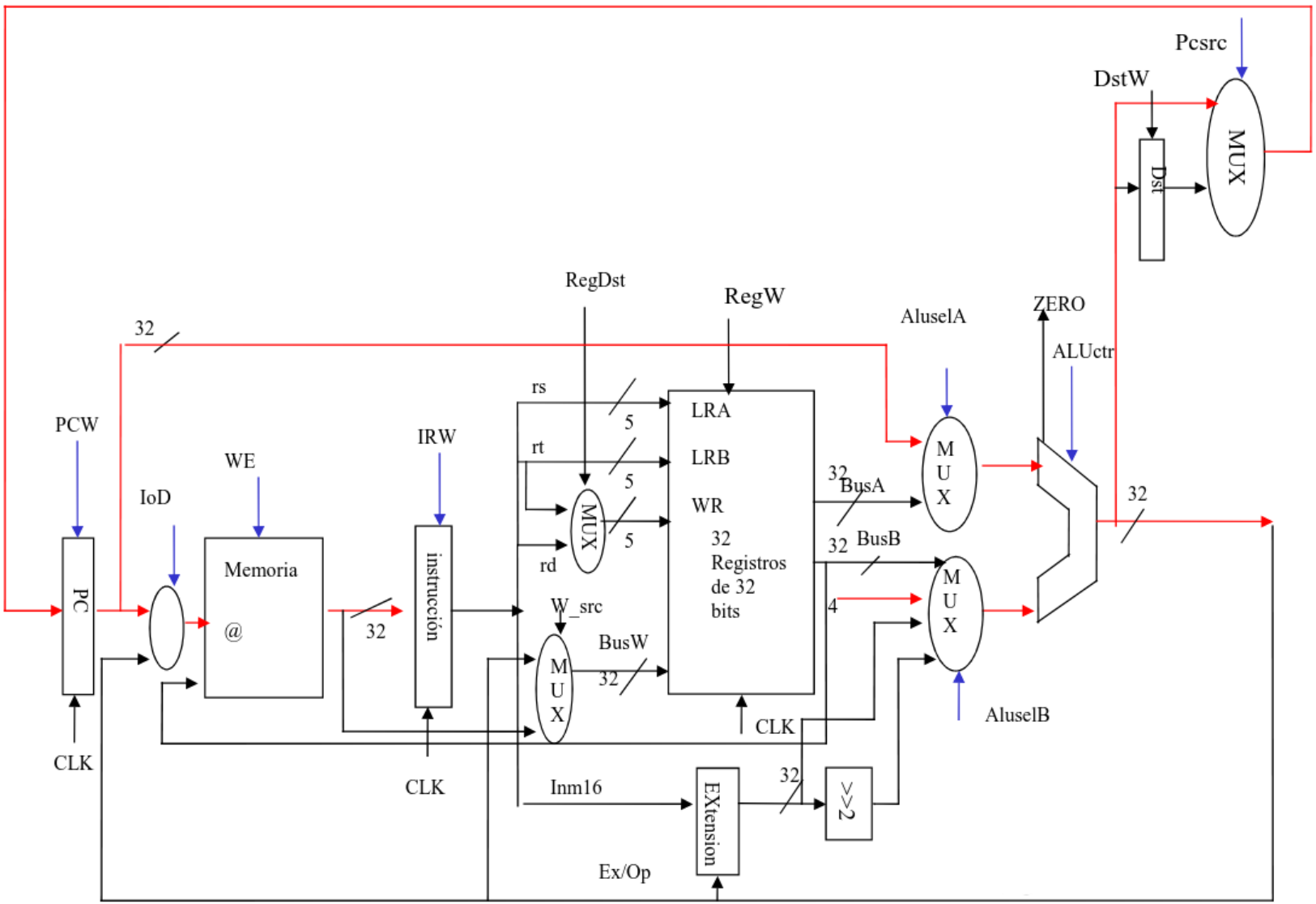
Cinco Etapas de Ejecución

- Búsqueda de instrucción (IF)
- Decodificación de instrucción y lectura de registros (ID)
- Ejecución, calculo de dirección de memoria o realización de salto (EXE)
- Acceso a memoria (MEM)
- Paso de escritura en registros (WB)

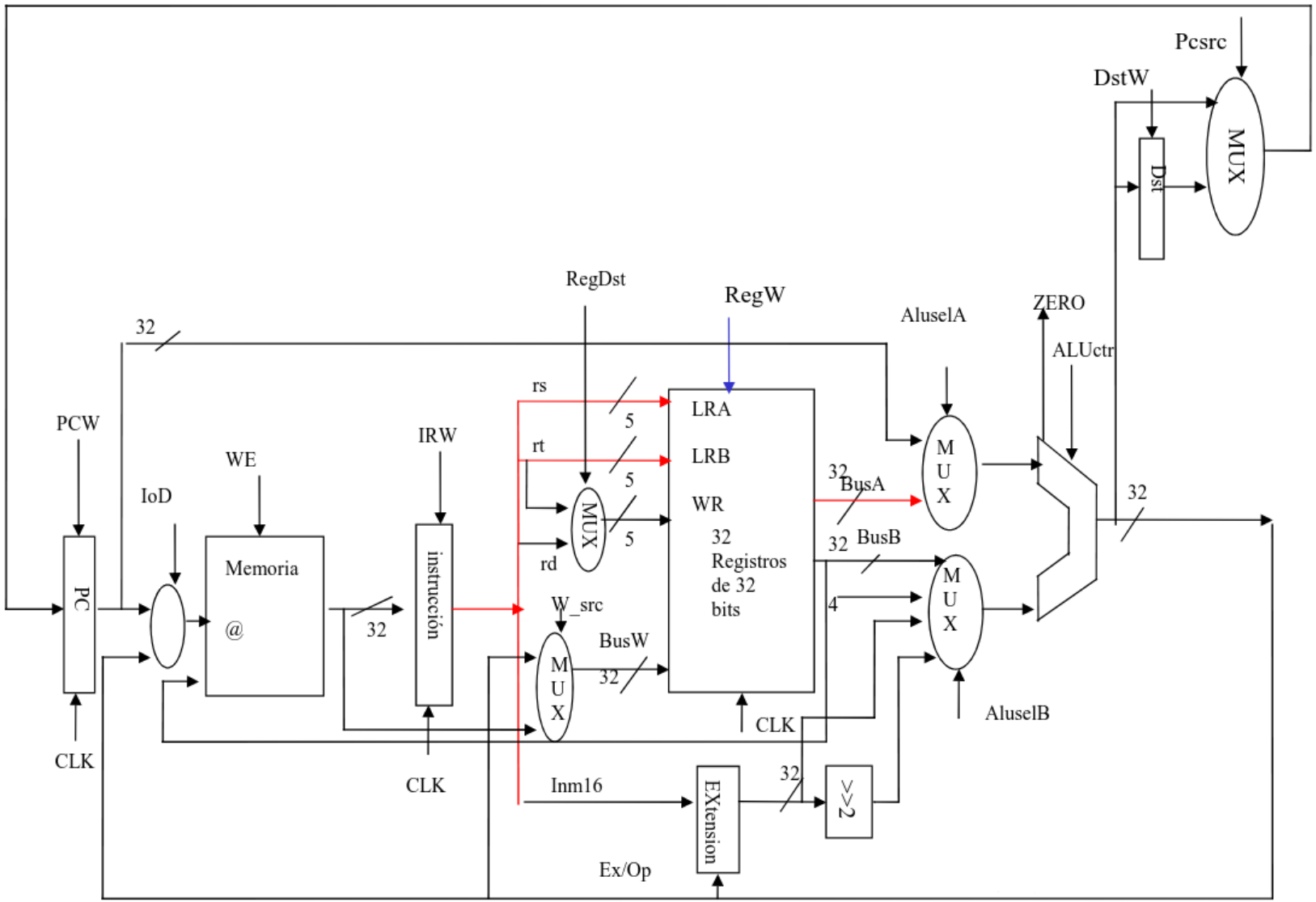
Las instrucciones tardan de 3 a 5 ciclos


PCW
Salto
Pcsrc
DstW
MUX
Dst
RegDst
RegW
AluselA
ZERO
ALUctr
32
rs
5
LRA
IRW
rt
5
LRB
WE
IoD
MUX
5
WR
32
BusA
MUX
rd
32 Registros de 32 bits
32
BusB
32
Memoria
@
instrucción
32
W_src
4
MUX
BusW
32
MUX
PC
CLK
CLK
AluselB
Inm16
EXtension
32
<<2
Ex/Op

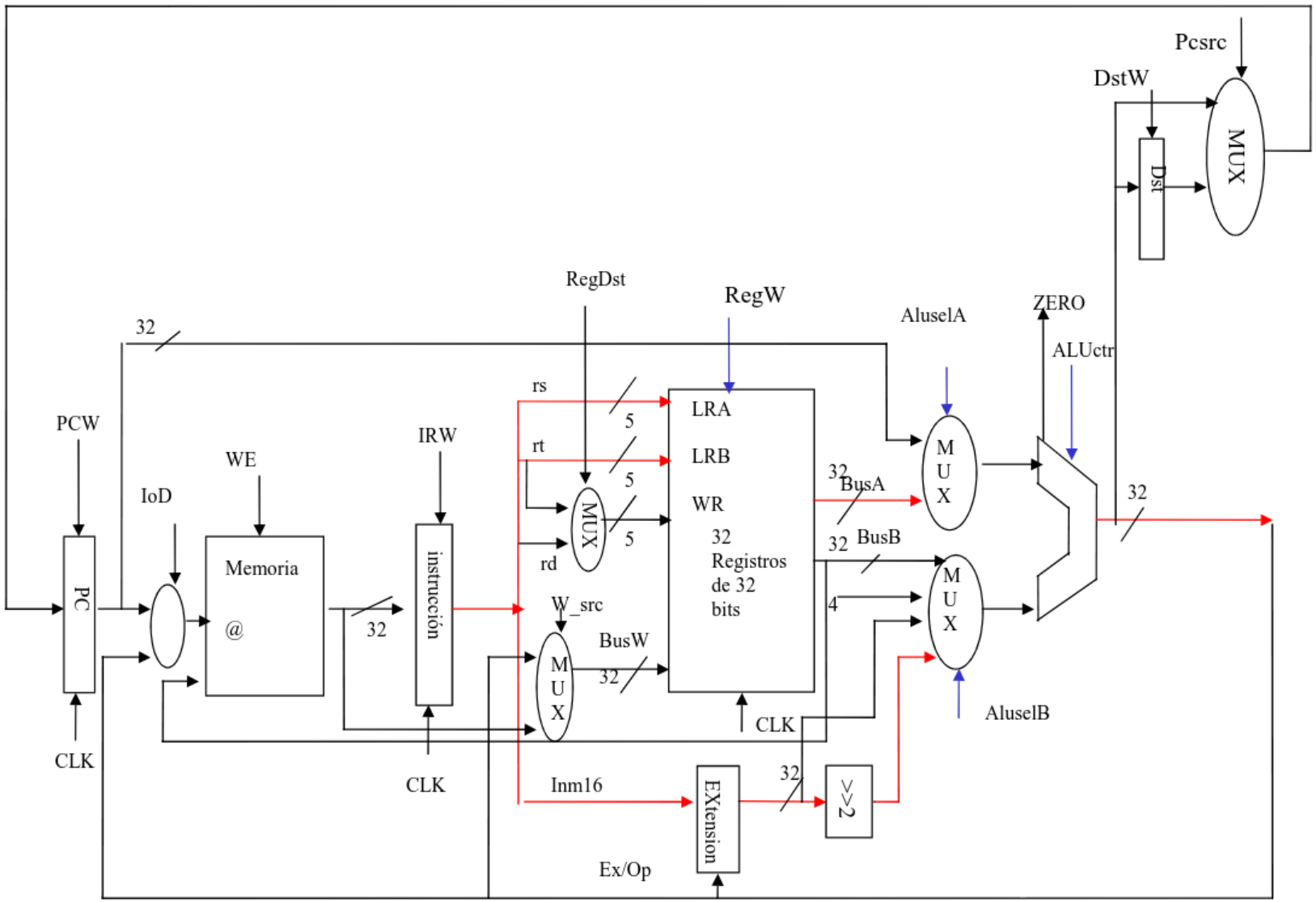
Instrucción Load: búsqueda



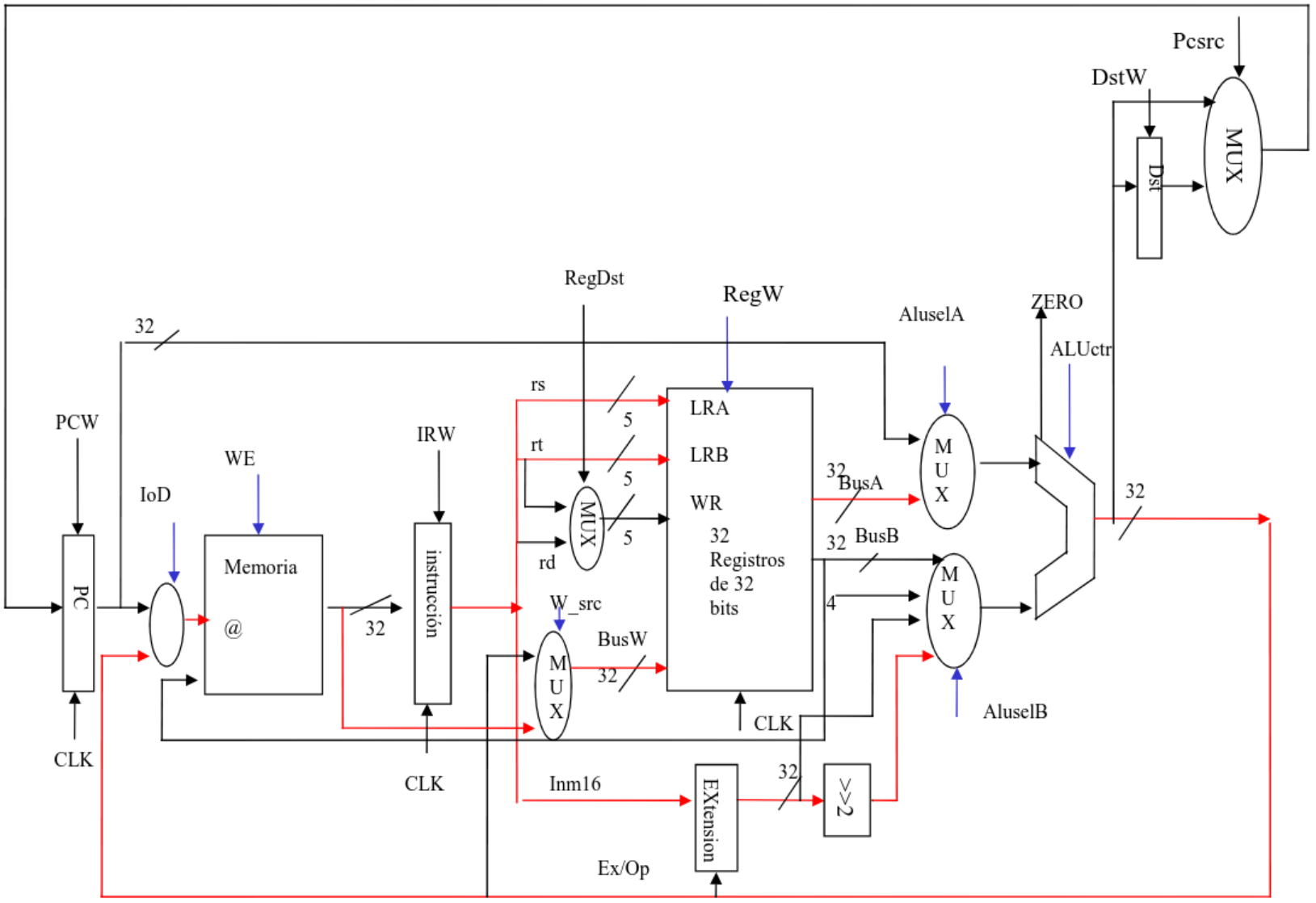
Instrucción Load: decodificación



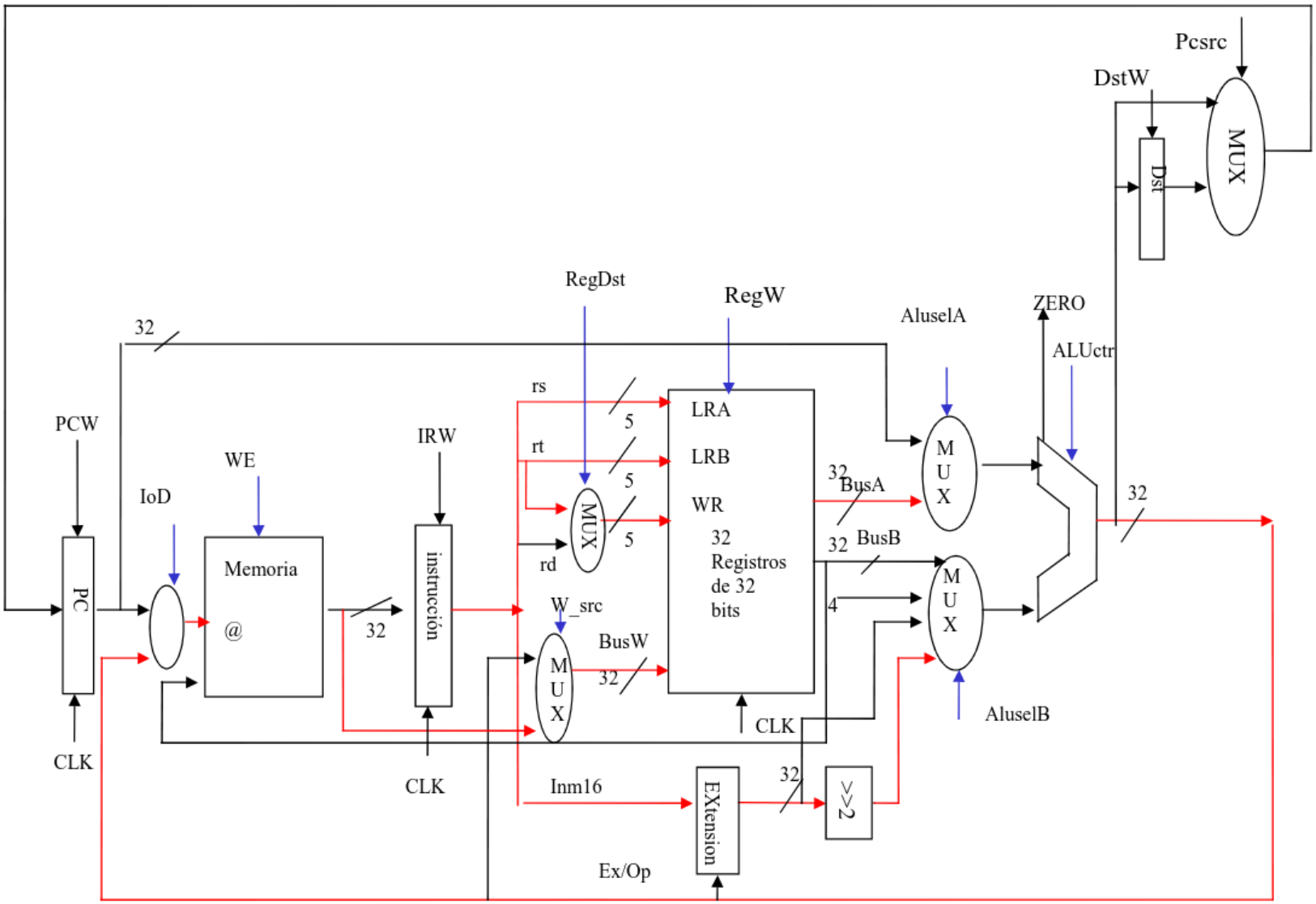
Instrucción Load: ejecución



Instrucción Load: memoria



Instrucción Load: write back



Etapas utilizadas en cada instrucción

	IF	ID	EXE	MEM	WB
Add	✓	✓	✓		✓
Ori	✓	✓	✓		✓
Sw	✓	✓	✓	✓	
Lw	✓	✓	✓	✓	✓
Beq	✓	✓	✓		

Rendimiento

Tipo	CPI	Frecuencia (%)	CPI x frecuencia
ALU	4	40	1,6
Load	5	30	1,5
Store	4	10	0,4
Branch	3	20	0,6
		CPI medio = 4,1	

Problema

Comparar el rendimiento de una máquina de 500Mhz, con el rendimiento de otra máquina de 750 Mhz pero en la cual el acceso a memoria de datos tarda dos ciclos.

Tipo	CPI_1	CPI_2	Frecuencia (%)	CPI_1 x frecuencia	CPI_2 x frecuencia
ALU	4	4	40	1,6	1,6
Load	5	6	30	1,5	1,8
Store	4	5	10	0,4	0,5
Branch	3	3	20	0,6	0,6

CPI $medio_1 = 4{,}1$

CPI $medio_2 = 4{,}5$

Comparando las dos máquinas

$$\frac{R_2}{R_1} = \frac{T_1}{T_2} = \frac{N \times CPI_1 \times t_1}{N \times CPI_2 \times t_2} = \frac{4,1 \times 100/50}{4,5 \times 100/75} = \frac{8,2}{6} = 1,36$$

La máquina de 750 Mhz es 36% más rápida que la máquina de 500 Mhz.

Segmentación

Descompone una determinada operación en n suboperaciones a realizar en etapas distintas, de manera que se puedan realizar n operaciones simultáneas, cada una en una etapa distinta.

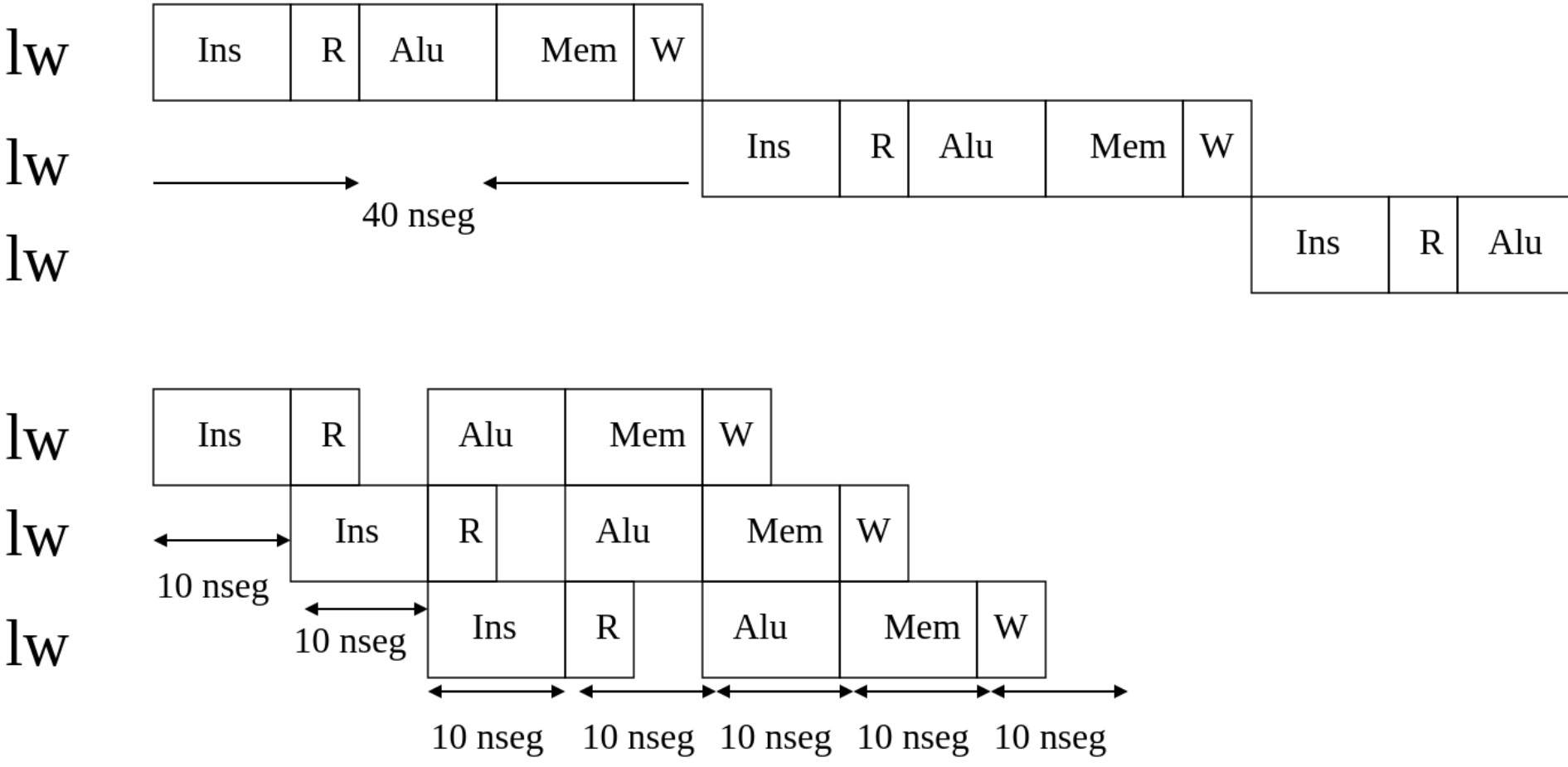
- Divide una operación en suboperaciones
- Desacopla las suboperaciones
- No reduce la latencia de una instrucción, sino que ayuda a incrementar la productividad de toda la tarea.
- Permite que los recursos se utilizen óptimamente, sin ciclos ociosos.
- La frecuencia con que una instrucción sale del pipeline (cauce) está limitada por el tiempo de proceso de la etapa más lenta.
- Idealmente, la mejora en velocidad debida a la segmentación es igual al número de etapas

Etapas de la segmentación

- Búsqueda de instrucción (IF)
- Decodificación de instrucción y lectura de registros (ID)
- Ejecución, calculo de dirección de memoria o realización de salto (EXE)
- Acceso a memoria (MEM)
- Paso de escritura en registros (WB)

Una etapa, un ciclo, debe acomodar la operación más lenta.

Máquina monociclo vs segmentada



Mejora de la segmentación

$$\frac{R_{seg}}{R_{mono}} = \frac{T_{mono}}{T_{seg}} = \frac{120}{70} = 1,7$$

- Donde están los problemas?
  - Hay que considerar el tiempo de llenado y vaciado del pipeline (20nseg+20nseg)
  - Las etapas tienen la misma duración y no realizan la misma cantidad de trabajo
  - La eficiencia del pipeline es mayor al considerar muchas instrucciones.

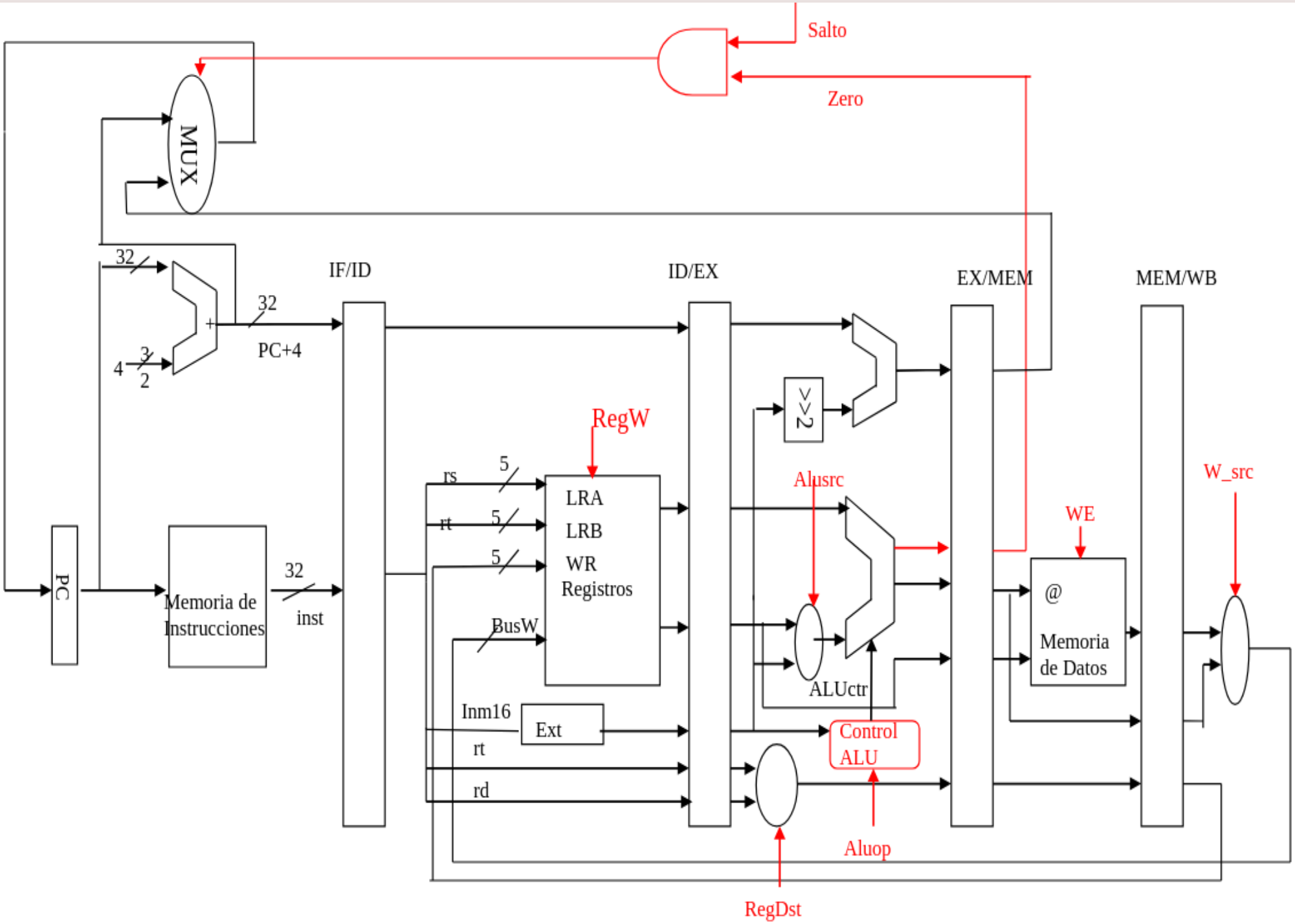
Mejora de la segmentación

Suponemos que ejecutamos 100 instrucciones

- Monociclo: $40nseg \times 1CPI \times 100inst = 4000nseg$
- Multiciclo: $10nseg \times 4,5CPI \times 100inst = 4500nseg$
- Segmentada: $10nseg \times (1CPI \times 100inst + 4ciclos) = 1040nseg$

$$\frac{R_{seg}}{R_{mono}} = \frac{T_{mono}}{T_{seg}} = \frac{4000}{1040} = 3,8$$

$$\frac{R_{seg}}{R_{mult}} = \frac{T_{mult}}{T_{seg}} = \frac{4500}{1040} = 4,3$$

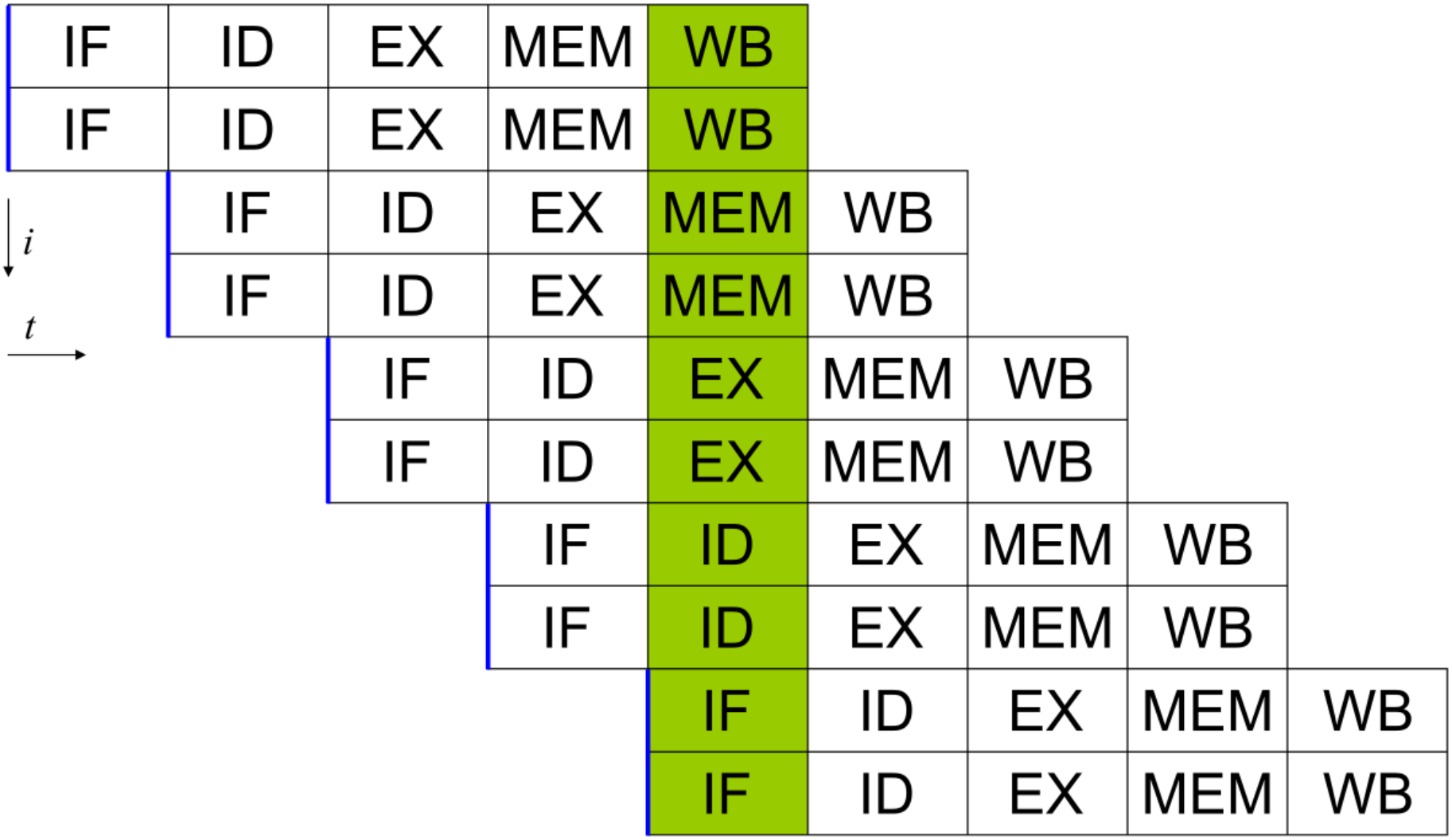

Salto
Zero
MUX
32
4
3
2
+
32
PC+4
IF/ID
ID/EX
EX/MEM
MEM/WB
RegW
rs
rt
5
5
5
LRA
LRB
WR
Registros
BusW
<<2
Alusrc
ALUctr
WE
@
Memoria
de Datos
W_src
PC
Memoria de
Instrucciones
32
inst
Inm16
Ext
rt
rd
Control
ALU
Aluop
RegDst

Limitaciones

- Conflictos en los recursos (dos instrucciones en ejecución no pueden usar el mismo recurso al mismo tiempo).
- Dependencia de control. Los saltos reducen la eficiencia.
- Dependencia de datos. Existe cuando dos instrucciones utilizan el mismo registro. Impiden que las instrucciones puedan reordenarse.

Arquitecturas superescalares

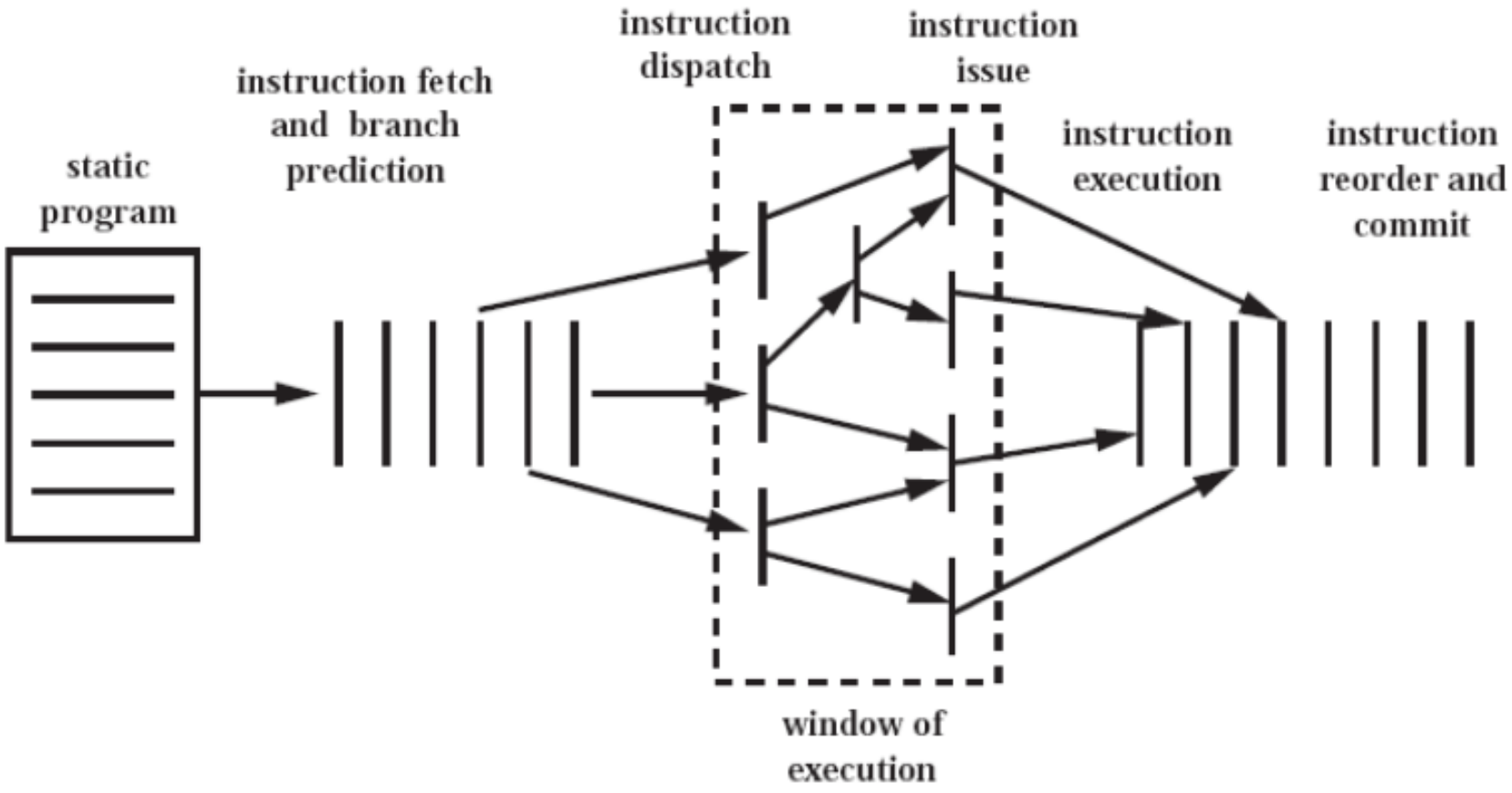
- La ejecución segmentada permite instrucciones simultáneas, pero sólo una instrucción puede estar en cada etapa del cauce.
- En una arquitectura superescalar se dispone de múltiples cauces de instrucciones independientes, y a su vez segmentados, de modo que puede iniciarse la ejecución de varias instrucciones en forma simultánea.
- Una arquitectura superescalar tiene las prestaciones de la segmentación, permitiendo además la existencia simultánea de varias instrucciones en la misma etapa. Para ello es necesaria la duplicación de recursos y la utilización de diversas técnicas que permitan optimizar su utilización.
- Como puede iniciarse la ejecución de varias instrucciones en el mismo ciclo, puede alcanzarse una productividad mayor que una instrucción por ciclo de reloj.


IF ID EX MEM WB
IF ID EX MEM WB
IF ID EX MEM WB
IF ID EX MEM WB
IF ID EX MEM WB
IF ID EX MEM WB
IF ID EX MEM WB
IF ID EX MEM WB
IF ID EX MEM WB
IF ID EX MEM WB
i
t

Paralelismo a nivel de instrucciones (ILP)

- Existe ILP cuando las instrucciones que componen un programa son independientes. En ese caso el código puede ser reordenado sin alterar el resultado. Por lo tanto existe la posibilidad de ejecutar las instrucciones en paralelo, rompiendo la secuencialidad implícita en un programa estático.
- Las dependencias de datos son críticas en este contexto.

Planificación dinámica



El ciclo de instrucción

- CAPTACION (fetch): múltiples instrucciones son captadas simultáneamente, utilizando técnicas de predicción de saltos y ejecución especulativa.
- DECODIFICACION (decode): en dos pasos, i) Predecodificación entre la memoria y el cache para identificación de saltos, y ii) Determinación de la operación, localización de operandos y localización del resultado.
- VENTANA DE EJECUCION = ENCOLADO (dispatch) y EMISION (issue): identificación de las instrucciones de la cola que están listas para comenzar su ejecución.
- EJECUCION (execute): en paralelo, en diferentes unidades funcionales.
- FINALIZACION (commit): El resultado es confirmado en su destino.