

# **Arquitectura de Computadoras**

---

## **Clase 8**

### **Procesadores superescalares**

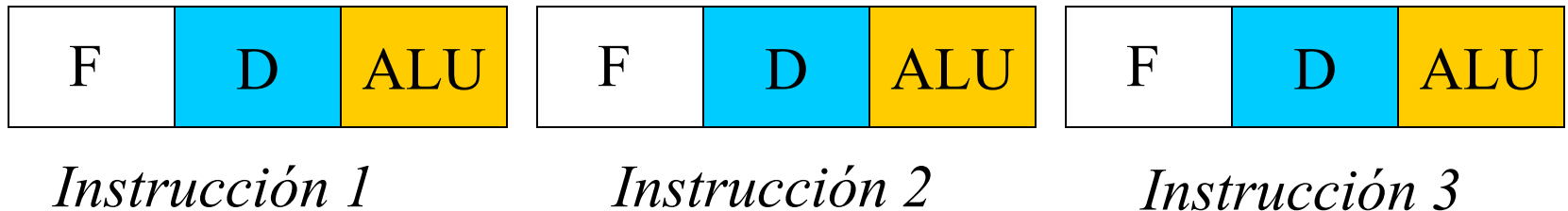
# Procesador Escalar

---

## Ejecución secuencial de instrucciones

F: búsqueda

D: decodificación



- Ejemplos: i4004, i8008/80, MC6800, MCS6502, Z80, F8 ...

# Segmentado de cauce

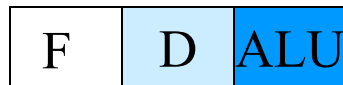
Una unidad segmentada es una secuencia de etapas con la propiedad de que nuevas operaciones pueden iniciarse mientras otras están en proceso

F: búsqueda

D: decodificación



*Instrucción 1*



*Instrucción 2*



*Instrucción 3*



*Instrucción 4*



*Instrucción 5*

- Ejemplos: MC68000, i8086/286, Z8000 ...

# Procesador Escalar segmentado

F: búsqueda  
D: decodificación  
EA: dirección efectiva  
E: ejecución  
M: memoria  
WB: post escritura



*Instrucción 1*



*Instrucción 2*



*Instrucción 3*



*Instrucción 4*



*Instrucción 5*

- Ejemplos: MC68020, i80386, R2000/3000 ...

# Mayores prestaciones

---

La evolución lógica de los diseños de cauces segmentados dio lugar a la aparición de dos técnicas de ejecución de mayores prestaciones.

- Procesadores Supersegmentados
- Procesadores Superescalares

# Enfoque Supersegmentado

---

- Muchas operaciones no necesitan todo un ciclo de reloj.
- Obtendremos mayores prestaciones subdividiendo el ciclo de reloj en sub-intervalos
  - Resulta una mayor frecuencia del ciclo de reloj.
- División de las etapas “macro” del cauce segmentado en sub-etapas más pequeñas (y, por tanto, más rápidas) y se transmiten los datos a la mayor velocidad del ciclo de reloj.
- El tiempo para las instrucciones individuales no varía.
  - Aumenta el grado del paralelismo.
  - Incrementa la aceleración percibida.

# Procesador supersegmentado

F: búsqueda

D: decodificación

EA: dirección efectiva

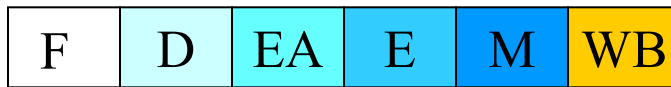
E: ejecución

M: memoria

WB: post escritura



*Instrucción 1*



*Instrucción 2*



*Instrucción 3*



*Instrucción 4*



*Instrucción 5*

Ejecución super-segmentada de instrucciones

# Enfoque Superescalar

---

- Se pueden llevar a cabo (completar) más de una instrucción simultáneamente.
- Conlleva la duplicación de algunas o todas las partes de la CPU/ALU.
  - Captar múltiples instrucciones al mismo tiempo.
  - Ejecutar sumas y multiplicaciones simultáneamente.
  - Ejecutar carga/almacenamiento, mientras se lleva a cabo una operación en ALU.
- El grado de paralelismo y, por tanto, la aceleración de la máquina aumentan, ya que se ejecutan más instrucciones en paralelo.



# Enfoque Superescalar (2)

F: búsqueda

D: decodificación

EA: dirección efectiva

E: ejecución

M: memoria

WB: post escritura

F	D	EA	E	M	WB
---	---	----	---	---	----

*Instrucción 1*

F	D	EA	E	M	WB
---	---	----	---	---	----

*Instrucción 2*

F	D	EA	E	M	WB
---	---	----	---	---	----

*Instrucción 3*

F	D	EA	E	M	WB
---	---	----	---	---	----

*Instrucción 4*

F	D	EA	E	M	WB
---	---	----	---	---	----

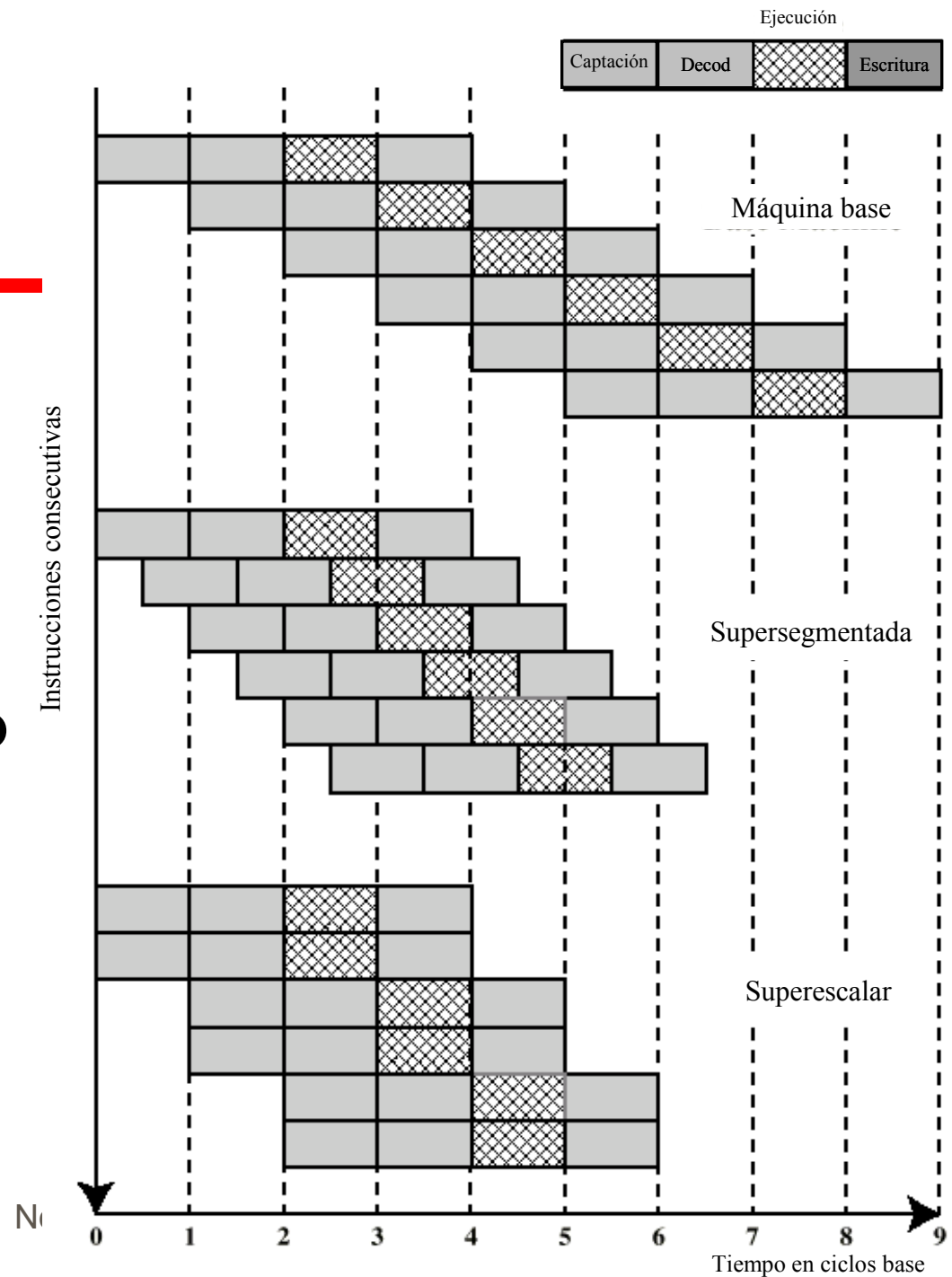
*Instrucción 5*

F	D	EA	E	M	WB
---	---	----	---	---	----

*Instrucción 6*

Ejemplos: MC68040, i80486, MC88110, i80860, PA-RISC, Sparc, R6000 ...

# Superescalar frente a supersegmentado



# Paralelismo de instrucciones

---

“existe cuando instrucciones de una secuencia son independientes y  $\Rightarrow$  pueden ejecutarse en paralelo solapándose”

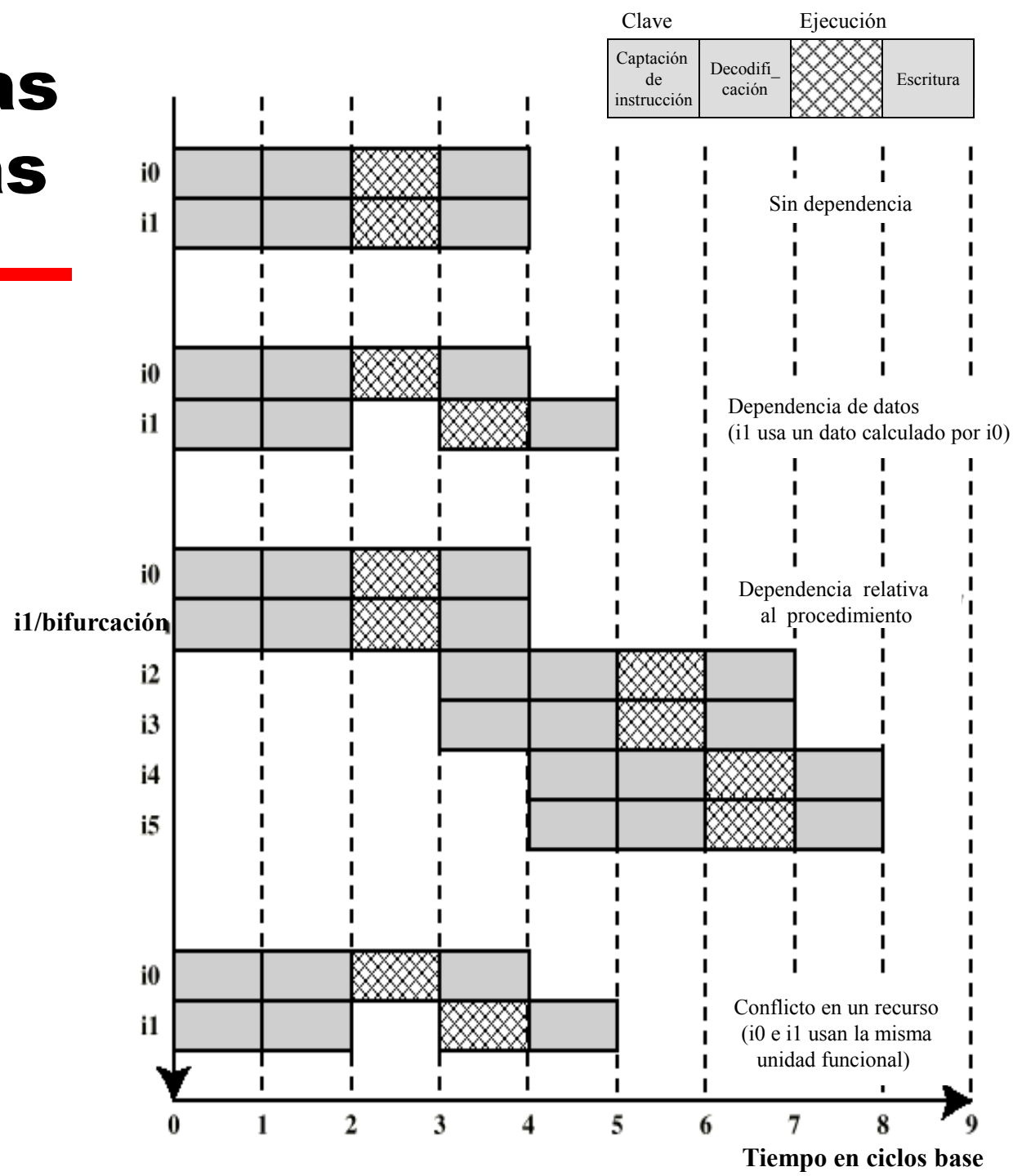
- Limitaciones

- Dependencia de datos verdadera.
- Dependencia relativa al procedimiento.
- Conflictos en los recursos.
- Dependencia de salida.
- Antidependencia.

# Efectos de las dependencias

I0 Add r1, r2

I1 Mov r3, r1



# Paralelismo de la máquina

---

“es una medida de la capacidad del procesador para sacar partido del paralelismo de instrucciones”

Depende de:

- Número de instrucciones captadas por ciclo
- Número de unidades funcionales
- Mecanismos para localizar instrucciones independientes
  - Identificar paralelismo y organizar F, D y E en paralelo.
  - Renombre de registros (dep. de salida o antidep.)
  - Ventana de instrucciones (emisión desordenada)

# Sobre las instrucciones

---

- En la localización de instrucciones independientes
  - Orden en que se captan las instrucciones.
  - Orden en que se ejecutan las instrucciones.
  - Orden en que las instrucciones actualizan los registros y las posiciones de memoria.
- Uso óptimo del cauce: atender dependencias
- Políticas de emisión en superescalares
  - Emisión y finalización en orden
  - Emisión en orden y finalización desordenada
  - Emisión y finalización desordenada

# Políticas de emisión: ejemplos

---

- Supóngase un cauce superescalar que:
  - Capta y decodifica 2 instrucciones a la vez
  - Posee 3 unidades funcionales independientes
  - Posee 2 copias de la etapa de escritura
- Supóngase un conjunto de 6 instrucciones con las siguientes características:
  - I1 tarda dos ciclos en ejecutarse
  - I3 e I4 requieren la misma unidad funcional
  - I5 depende del valor producido por I4
  - I5 e I6 requieren la misma unidad funcional

# Emisión y finalización en orden

---

Decodificación		Ejecución			Escritura		Ciclo
I1	I2						1
I3	I4	I1	I2				2
I3	I4	I1					3
	I4			I3	I1	I2	4
I5	I6			I4			5
	I6		I5		I3	I4	6
			I6				7
					I5	I6	8



# Emisión en orden y finalización desordenada

---

Decodificación		Ejecución			Escritura		Ciclo
I1	I2						1
I3	I4	I1	I2				2
	I4	I1		I3	I2		3
I5	I6			I4	I1	I3	4
	I6		I5		I4		5
			I6		I5		6
					I6		7

# Emisión y finalización desordenada

---

Decodificación	Ventana	Ejecución	Escritura	Ciclo																																																
<table><tr><td>I1</td><td>I2</td></tr><tr><td>I3</td><td>I4</td></tr><tr><td>I5</td><td>I6</td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td></td><td></td></tr></table>	I1	I2	I3	I4	I5	I6							<table><tr><td></td></tr><tr><td>I1, I2</td></tr><tr><td>I3, I4</td></tr><tr><td>I4, I5, I6</td></tr><tr><td>I5</td></tr><tr><td></td></tr></table>		I1, I2	I3, I4	I4, I5, I6	I5		<table><tr><td></td><td></td><td></td></tr><tr><td>I1</td><td>I2</td><td></td></tr><tr><td>I1</td><td></td><td>I3</td></tr><tr><td></td><td>I6</td><td>I4</td></tr><tr><td></td><td>I5</td><td></td></tr><tr><td></td><td></td><td></td></tr></table>				I1	I2		I1		I3		I6	I4		I5					<table><tr><td></td><td></td></tr><tr><td></td><td></td></tr><tr><td>I2</td><td></td></tr><tr><td>I1</td><td>I3</td></tr><tr><td>I4</td><td>I6</td></tr><tr><td>I5</td><td></td></tr></table>					I2		I1	I3	I4	I6	I5		<div>1</div> <div>2</div> <div>3</div> <div>4</div> <div>5</div> <div>6</div>
I1	I2																																																			
I3	I4																																																			
I5	I6																																																			
I1, I2																																																				
I3, I4																																																				
I4, I5, I6																																																				
I5																																																				
I1	I2																																																			
I1		I3																																																		
	I6	I4																																																		
	I5																																																			
I2																																																				
I1	I3																																																			
I4	I6																																																			
I5																																																				

# Riesgos

---

R3:= R3 op R5 (1)

R4:= R3 + 1 (2)

R3:= R5 + 1 (3)

R7:= R3 op R4 (4)

## Riesgos:

### 1) RAW (dependencia verdadera)

La salida de (1), R3, es uno de los operandos en (2)

### 2) WAW (dependencia de salida)

(1) y (3) intentan escribir en R3

### 3) WAR (antidependencia)

La instrucción (3) no puede comenzar hasta que la instrucción (2) haya obtenido uno de sus operandos (R3)

# Renombre de registros

---

- Las dependencias de salida y antidependencias surgen porque los valores de los registros no pueden reflejar la secuencia de valores dictada por el flujo del programa.
  - Conflictos de almacenamiento
- Esto puede detener alguna etapa del cauce.
- Cuando la ejecución de una instrucción guarda un resultado en Registro, se almacena en un registro nuevo

## Renombramiento de Registros

- Los registros se asignan dinámicamente
  - o sea, las referencias posteriores son a los registros nuevos.

# Riesgos y renombre: ejemplo

---

R3b:= R3a op R5a      (1)

R4a:= R3b + 1      (2)

R3c:= R5a + 1      (3)

R7a:= R3c op R4a      (4)

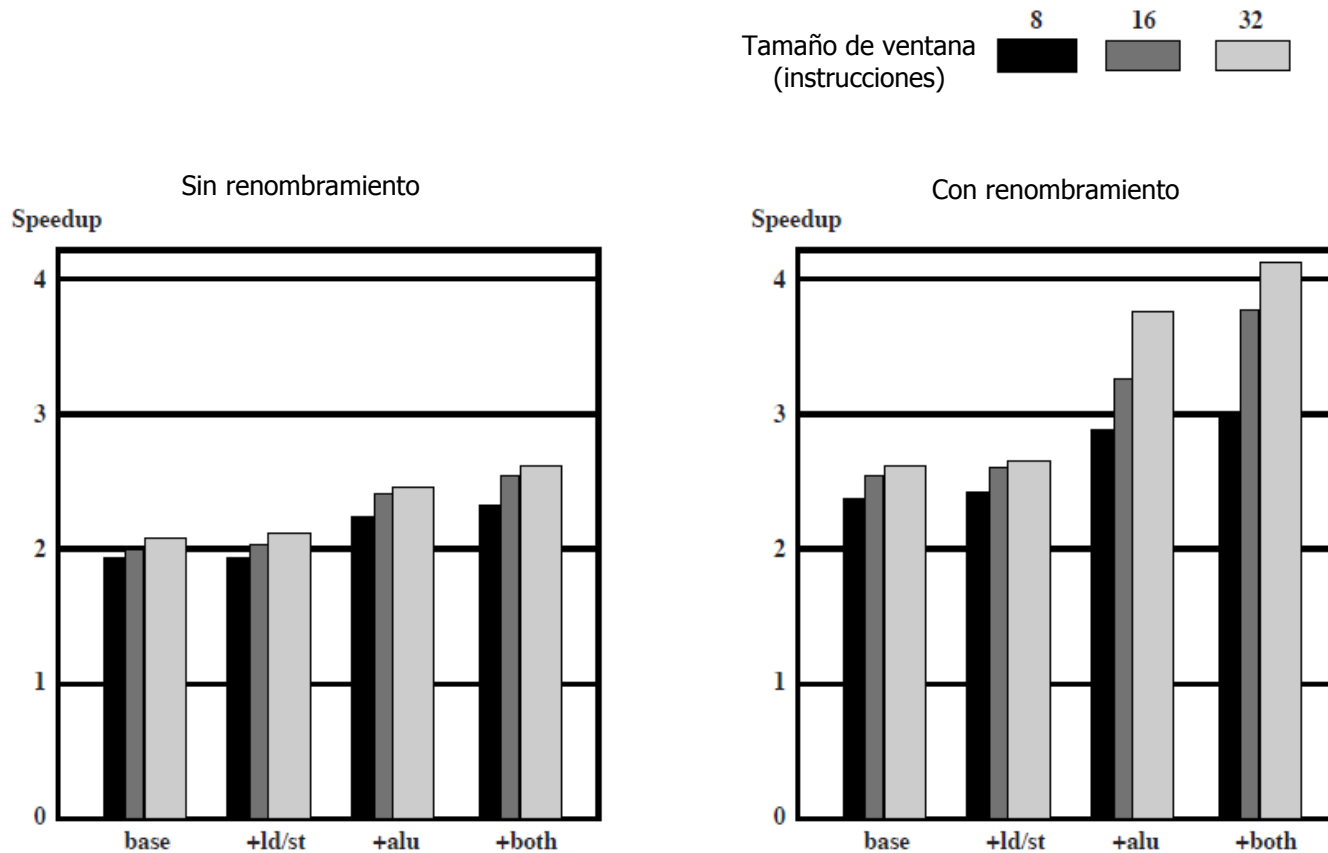
## Riesgos:

Sólo quedan los riesgos RAW (dependencia verdadera).

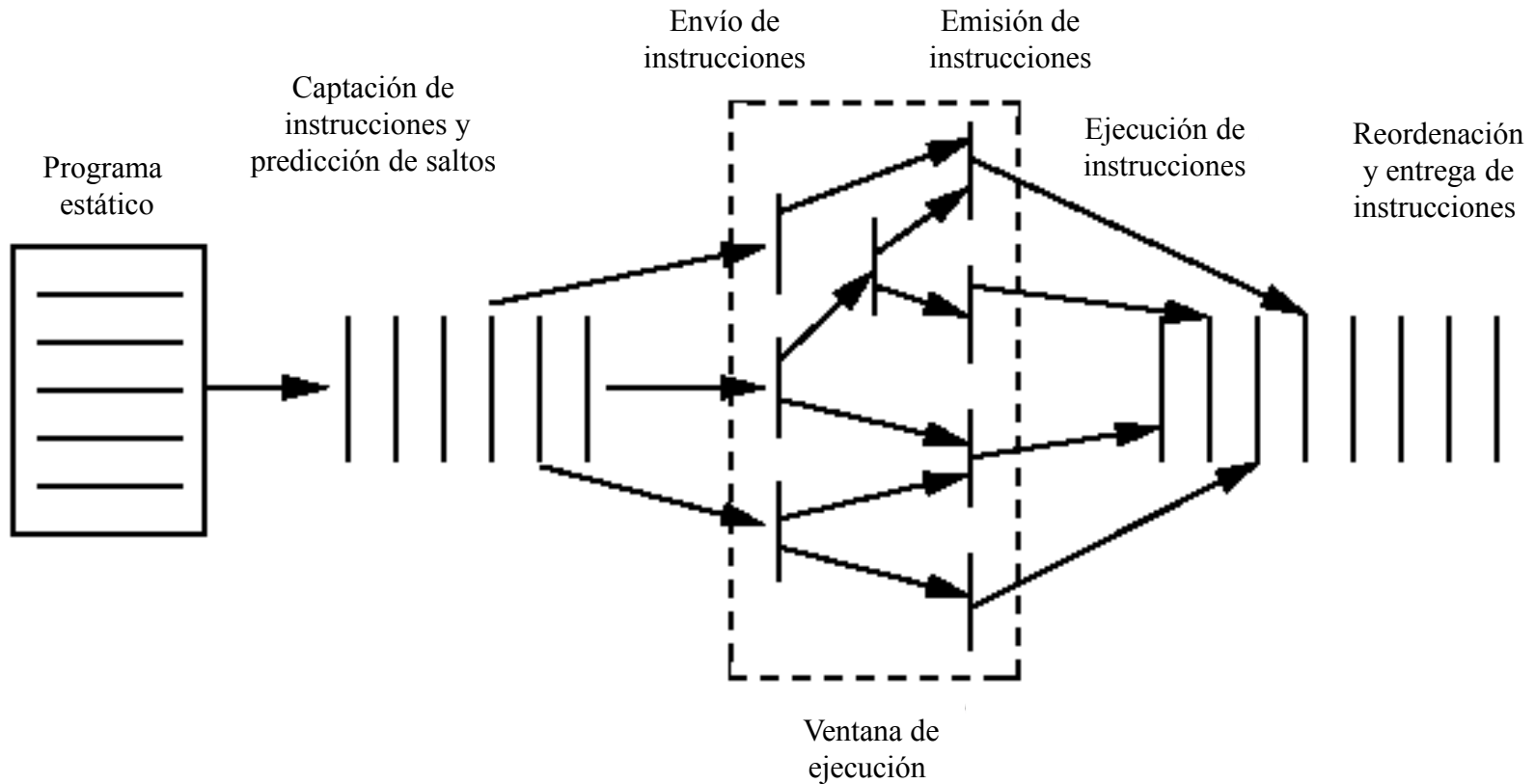
La salida de (1), R3b, es uno de los operandos en (2).

La salida de (3), R3c, es uno de los operandos en (4).

# Mejoras obtenibles en superescalar



# Ejecución superescalar



# Implementación superescalar

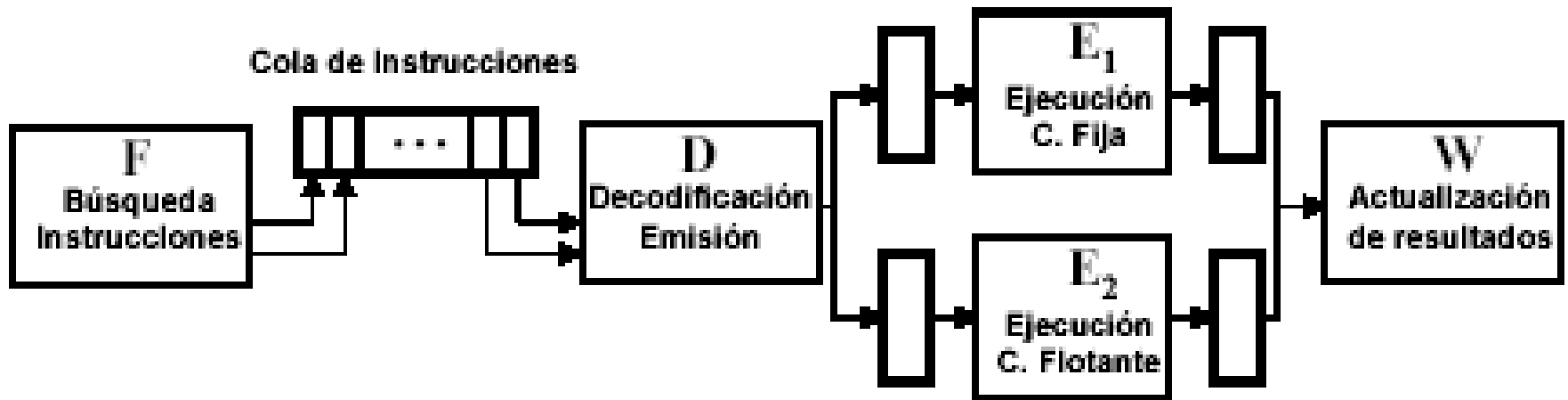
---

- Estrategias de captación simultánea de múltiples instrucciones.
- Lógica para determinar dependencias verdaderas entre valores de registros y mecanismos para comunicar esos valores.
- Mecanismos para iniciar o emitir múltiples instrucciones en paralelo.
- Recursos para la ejecución en paralelo de múltiples instrucciones.
- Mecanismos para entregar el estado del procesador en un orden correcto.



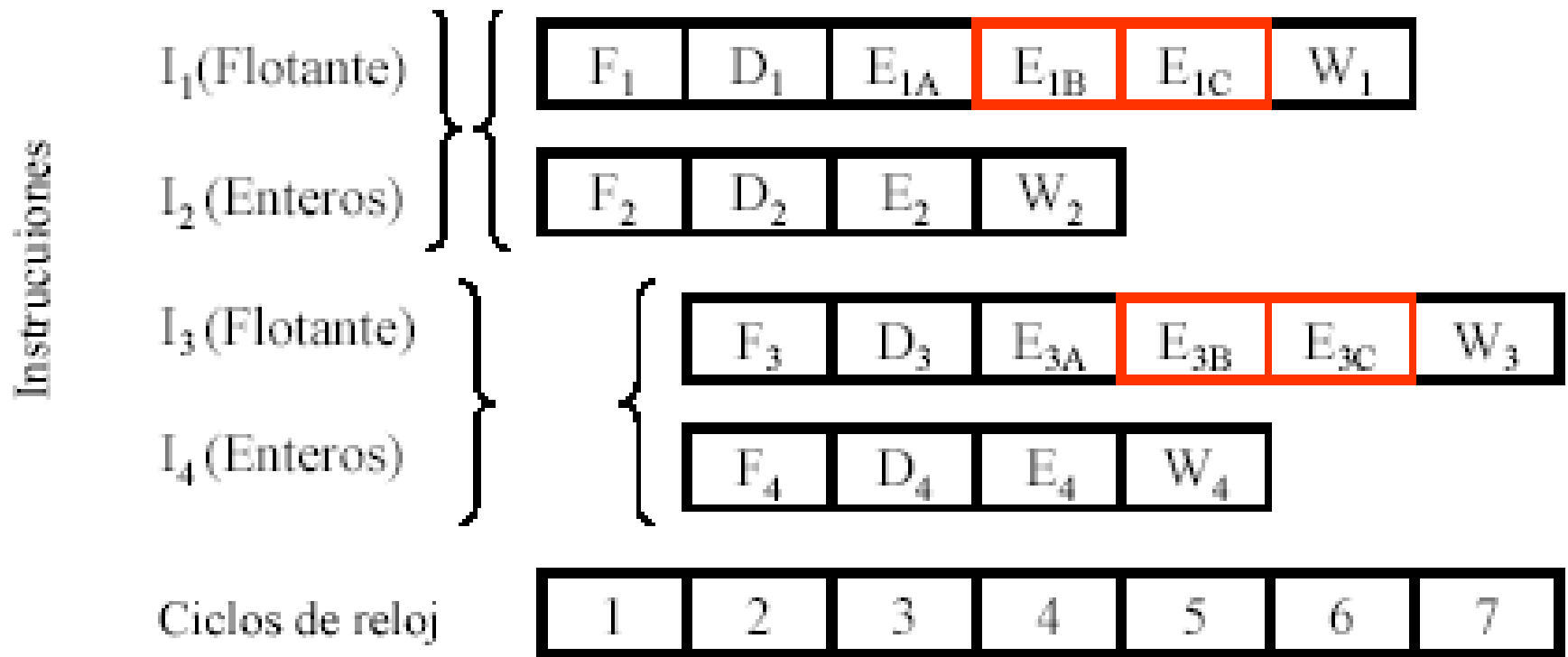
# Ejemplo de procesador con 2 unidades de ejecución

---



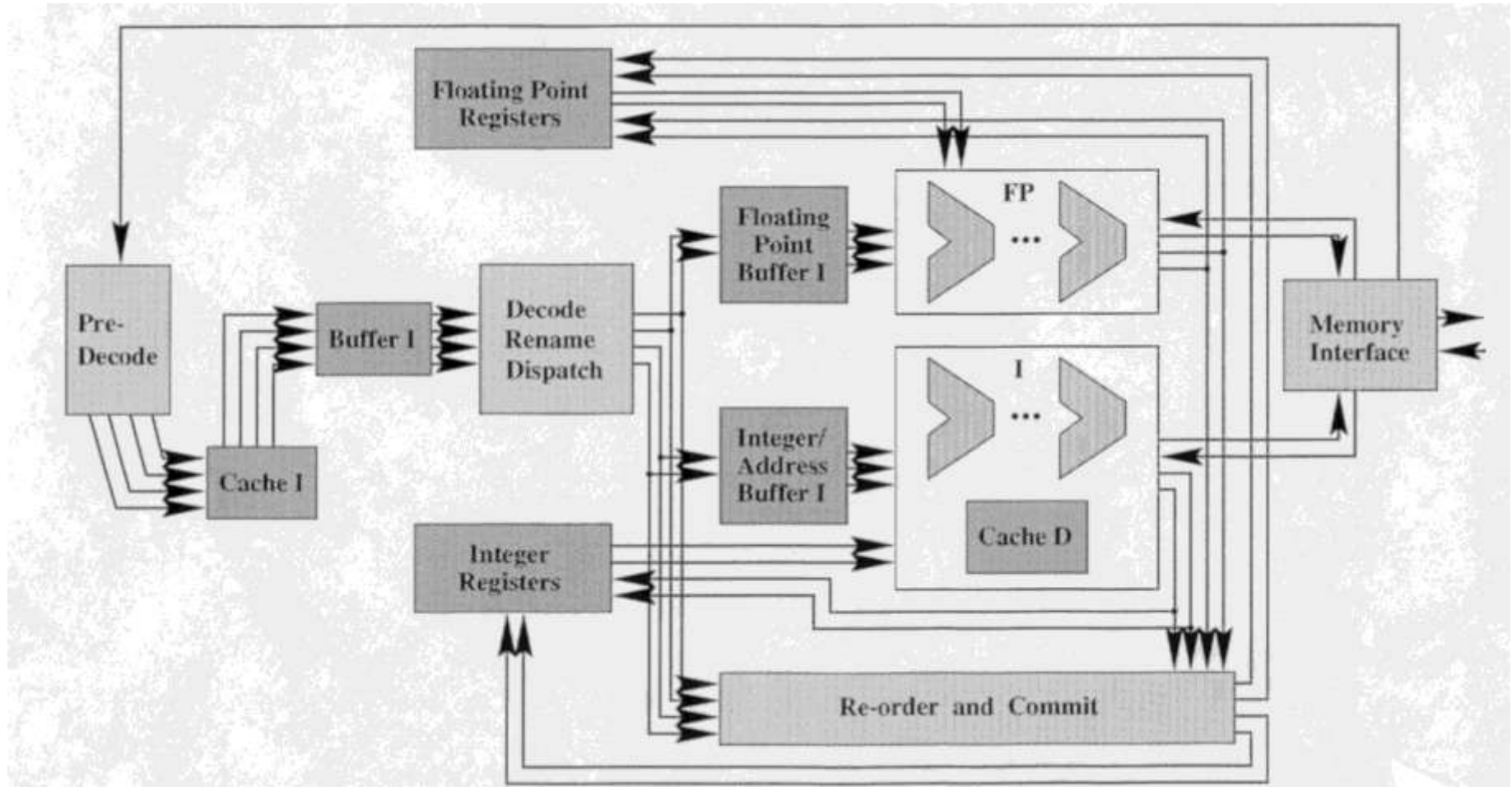
- La unidad de emisión puede tomar 2 instrucciones de la cola y decodificarlas.
- Si una es de enteros y otra de coma flotante (segmentada) y no existen riesgos  
⇒ pueden emitirse y ejecutarse a la vez.

# Ejemplo (cont.)



- Gran responsabilidad del compilador en el aprovechamiento del procesamiento superescalar

# Procesador superescalar



# Consideraciones destacables en el procesamiento superescalar

---

- Influencia de las excepciones

En el momento en que se produce una excepción hay varias instrucciones en ejecución

Si  $I_1$  produce una excepción  $\Rightarrow$  ¿Ha podido terminar  $I_2$ ?

$\Rightarrow$  Estado inconsistente (*Excepciones imprecisas*)

El comportamiento debería ser "idéntico" al que tendría la misma computadora no segmentada

Para garantizar un estado consistente (*preciso*):

- Instrucciones anteriores terminan correctamente
- La que origina la excepción y siguientes se abortan
- Tras la rutina de tratamiento se comienza por la que originó la excepción

# Consideraciones destacables en el procesamiento superescalar (2)

---

## *Excepciones precisas*

Por ejemplo: los resultados se almacenan en el orden en que aparecen las instrucciones.

- Retardar el  $W_2$  hasta el ciclo 6 (ver pag. 26)
- Con las interrupciones externas  $\Rightarrow$  *Excepciones precisas*
- La unidad de emisión deja de emitir y se cancela la cola.
- Todas las instrucciones pendientes se completan.  
 $\Rightarrow$  Comienza el procesamiento de la interrupción.

# Consideraciones destacables en el procesamiento superescalar (3)

---

## Compromiso entre

- Ejecución desordenada  $\leftrightarrow$  Liberar las unidades de ejecución
- Completar instrucciones en orden  $\leftrightarrow$  Excepciones precisas

## Possible solución

Emisión desordenada y finalización ordenada



Etapa de escritura temporal (TW)

## ***Renombramiento de registros***

El registro temporal  $TW_i$  se usa sólo por las instrucciones posteriores a  $i$ .

# Lecturas recomendadas

---

- *Organización y Arquitectura de Computadoras*, William Stallings, Capítulo 13 de 5<sup>ta</sup> edición ó Capítulo 14 de 7<sup>ma</sup> edición.
- *Diseño y evaluación de arquitecturas de computadores*, M. Pardo y A. Guzmán, Capítulo 3. 1<sup>o</sup> ed.