

Arquitectura e Ingeniería de Computadores

Tema 2 – Segmentación y superescalares clase 3

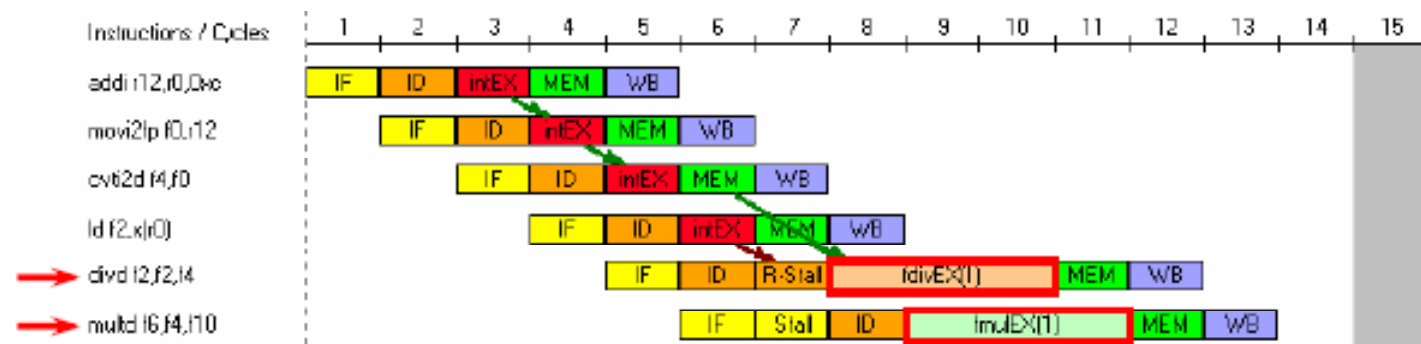
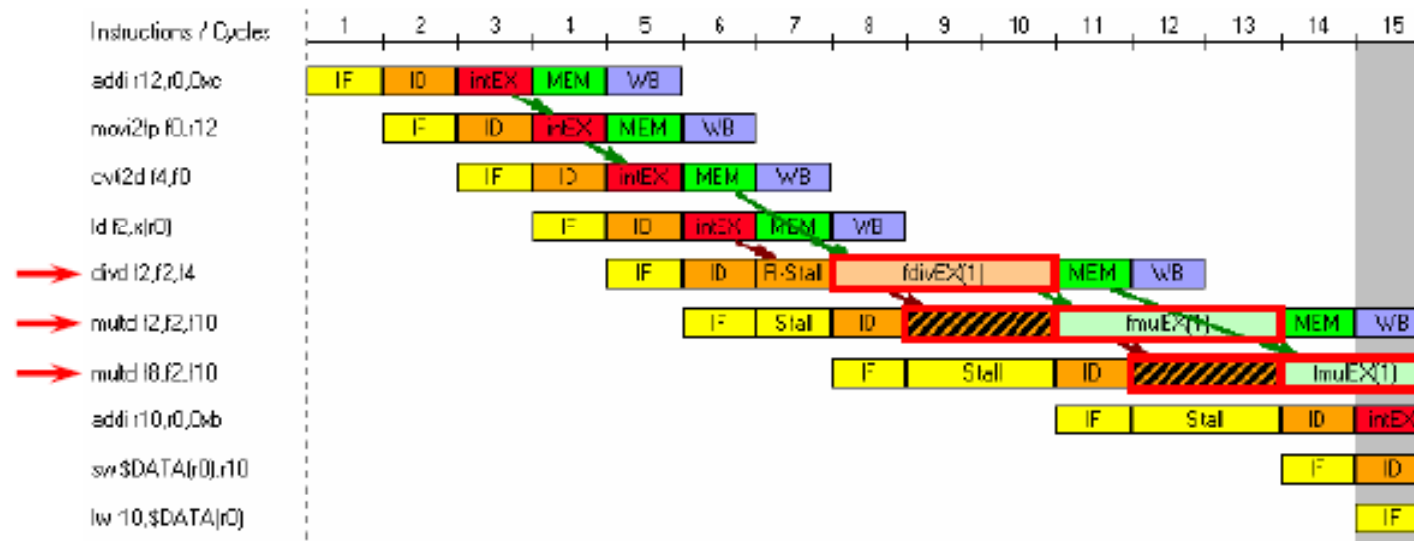
Ingeniería en Informática

Departamento de Tecnología Informática y Computación

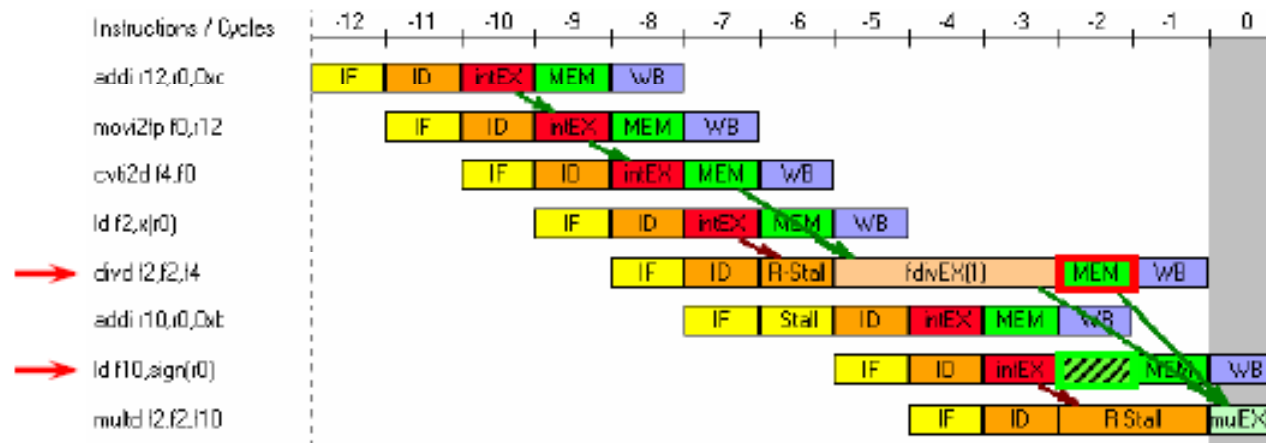
◆ Motivación del procesamiento superescalar

- Las dependencias estructurales provocan la pérdida de ciclos

Ej: una única unidad de FP vs varias unidades FP

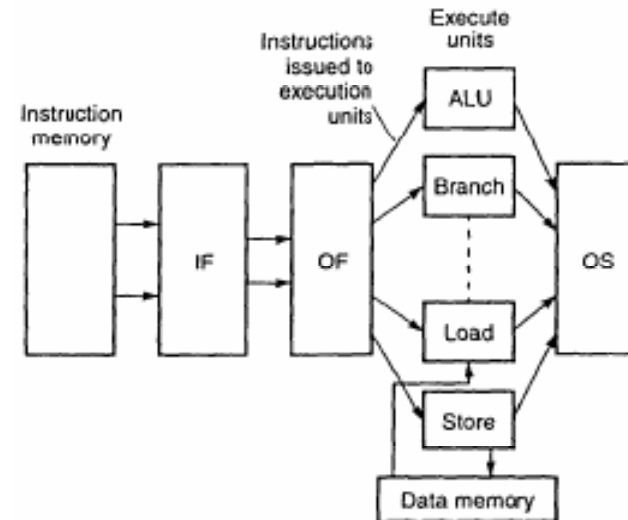
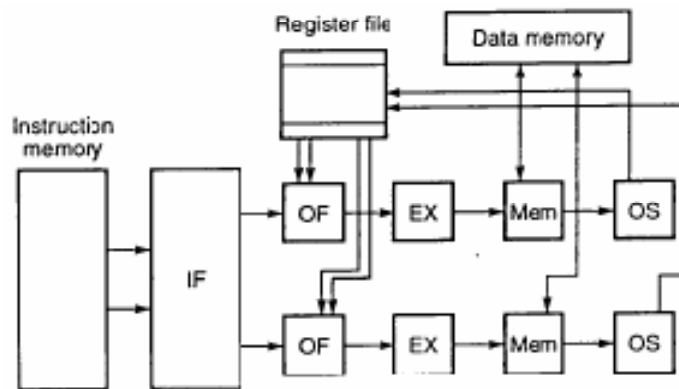


- La utilización de varias unidades funcionales permitiría la **ejecución de instrucciones fuera de orden** (habría que comprobar riegos WAR y WAW)



- Al disponer de una única unidad MEM y WB sólo se puede terminar 1 inst/ciclo
- Se podrían conseguir mejores prestaciones si se pudieran tener varias instrucciones en las distintas etapas (IF, ID, MEM y WB) => **procesamiento superescalar**

- ◆ **Definición:** un proc. Superescalar es un proc. Segmentado que puede iniciar varias instrucciones (que se ejecutarían simultáneamente) y terminar varias instrucciones por ciclo



◆ Ejemplos:

Familia	MIPS			AMD	Intel		
Microprocesador	R8000	R10000	RM7000	Athlon K7	Pentium II	Pentium III	Pentium 4
Año	1994	1995	1997	1999	1997	1999	2000
Frecuencia (MHz)	75	275	300	1200	300	450	1400
Superscalar (emisión)	4	4	2	3 (x86)	6 uop (3 x86)	6 uop (3 x86)	3 uop
Interrupciones	P/I	P (Hist.)	P	Prec.	Prec.	Prec.	Prec.
Salto (Predicción)	Dinam. (BTAC)	Dinam. (BTH)	Retard.	Dinam.	Dinam.	Dinam.	Dinam.
Segmentación (Etapas)	5	5	5	9/11 (FP 15)	12	12	22/24

◆ Paralelismo entre instrucciones (ILP)

Depende de:

- dependencias de datos
- dependencias de control
- retardo de las operaciones

```
load r1, r2(23) }  
add r3, #1      } (Paralelismo: 3)  
add r4, r5      }  
add r3, #1      }  
add r4, r3      }  
store (r4), r0  } (Paralelismo: 1)
```

◆ Paralelismo de la máquina

Depende de:

- nº de instrucciones que pueden captarse y ejecutarse simultánea
- velocidad del procesador para encontrar y resolver dependencias

Programa	Máquina Ideal (IPC)	Limitaciones (IPC)
gcc	54.8	2-35
espresso	62.6	2-41
li	17.9	2-16
fpppp	75.2	3-59
doducd	118.7	4-58
tomcatv	150.1	4-60

◆ Fases del procesamiento superescalar

- **Captación de Instrucciones (IF)**

Se llevan las instrucciones de la cache al procesador

- **Decodificación de Instrucciones (ID)**

Se identifica qué operación hay que realizar y con qué operandos

- **Emisión de Instrucciones (*Issue*, ISS)**

Las instrucciones esperan hasta que las unidades funcionales donde se van a ejecutar están libres y los operandos que necesitan estén disponibles

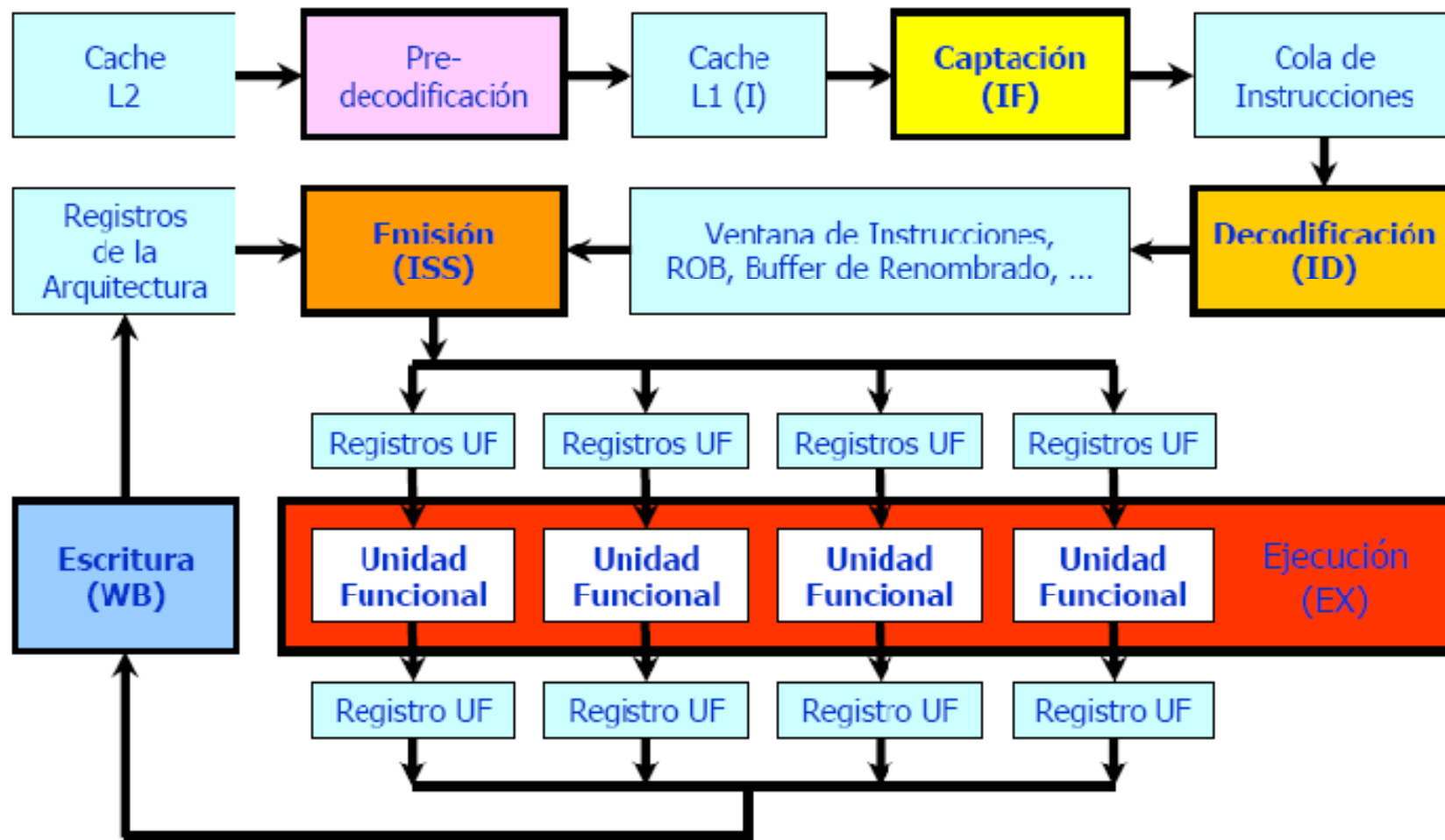
- **Ejecución de Instrucción (EX)**

Se realizan las operaciones codificadas en las instrucción (operaciones aritméticas, lectura o escritura de memoria, determinación de si se produce salto o no,...). Termina con la ejecución de los resultados en los registros intermedios (*Finish*)

- **Escritura (WB)**

Escritura de los resultados en los registros de la arquitectura (*Commit*, o *Complete*)

◆ Etapas de un cauce superescalar



◆ Características propias del procesamiento superescalar

- **En una secuencia de instrucciones se pueden distinguir tres tipos de ordenaciones:**
 - El orden en que se **captan** las instrucciones (el orden de las instrucciones en el código)
 - El orden en que las instrucciones se **ejecutan**
 - El orden en que las instrucciones **cambian los registros y la memoria.**
- El procesador debe ser capaz de **identificar el paralelismo entre instrucciones (ILP)** que exista en el programa y **organizar la captación, decodificación y ejecución de instrucciones en paralelo**, utilizando eficazmente los recursos existentes.
- Cuanto más sofisticado sea el procesador, **menos tiene que ajustarse a la ordenación de las instrucciones según se captan, para la ejecución y modificación de los registros** de cara a mejorar los tiempos de ejecución. La única restricción es que *el resultado del programa sea correcto*.

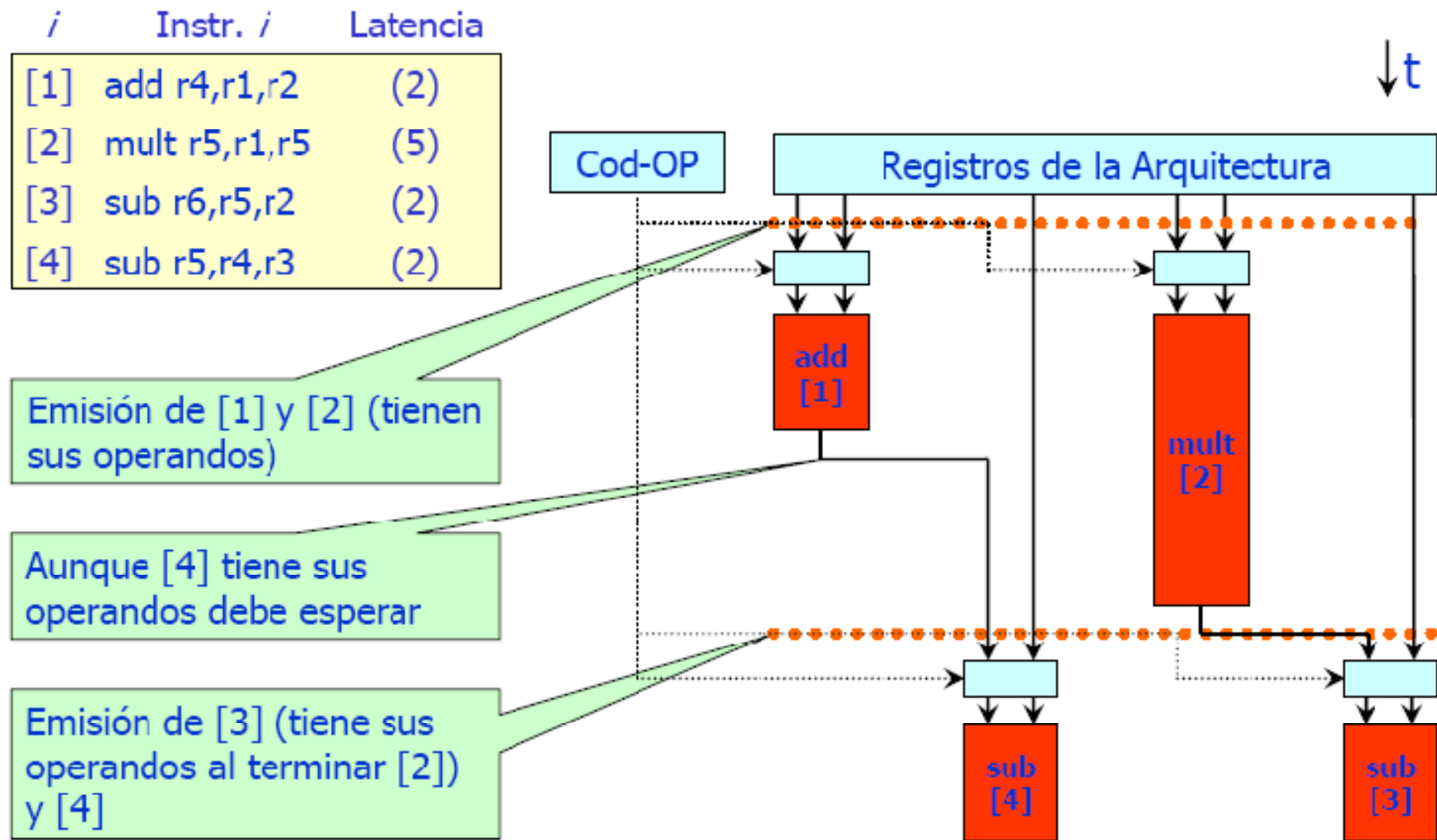
Orden de emisión

Orden de finalización

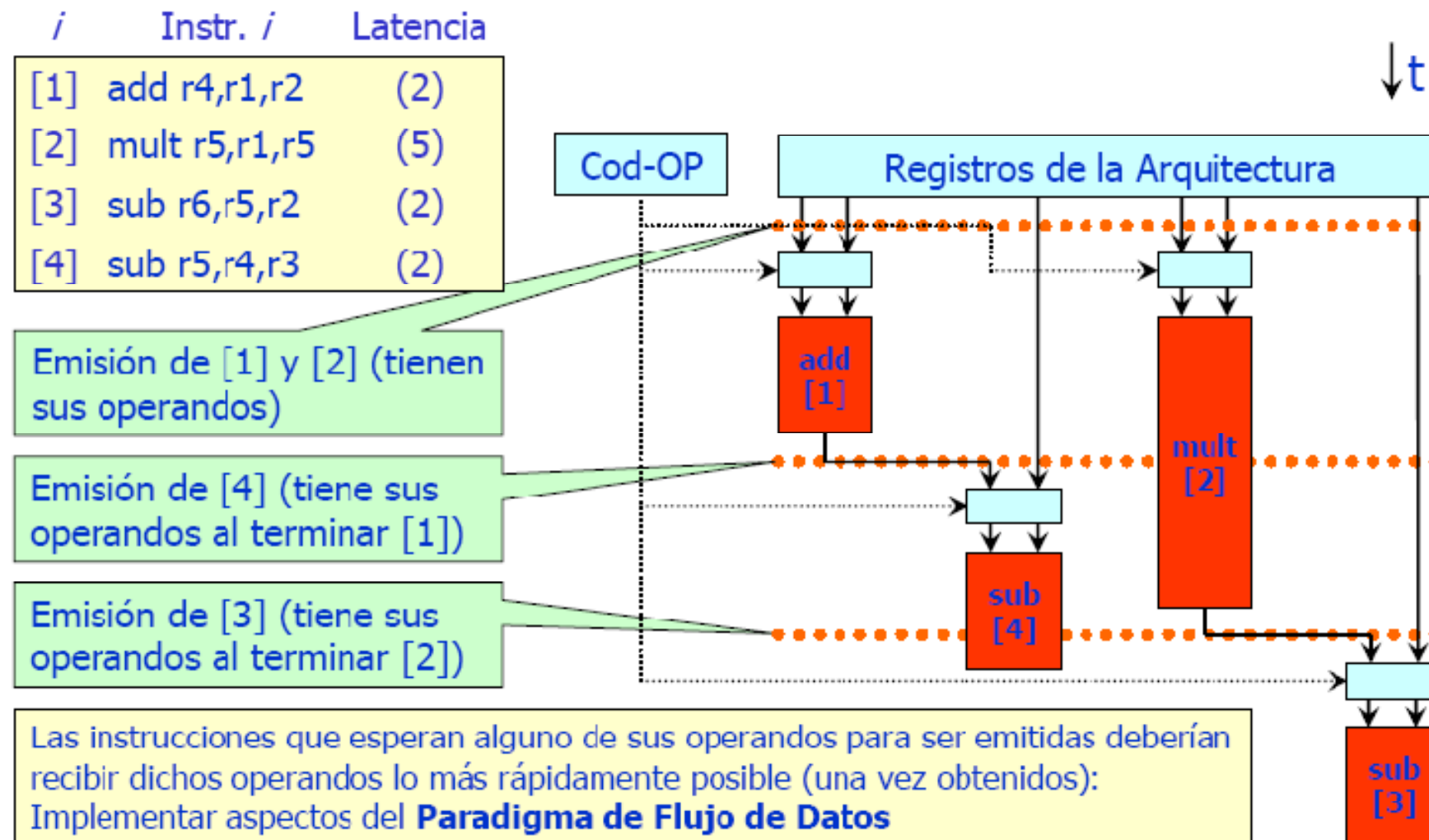
◆ Orden de emisión y finalización

- ▶ **Emisión Ordenada con Finalización Ordenada**
- ▶ **Emisión Ordenada con Finalización Desordenada**
- ▶ **Emisión Desordenada con Finalización Ordenada**
- ▶ **Emisión Desordenada con Finalización Desordenada**

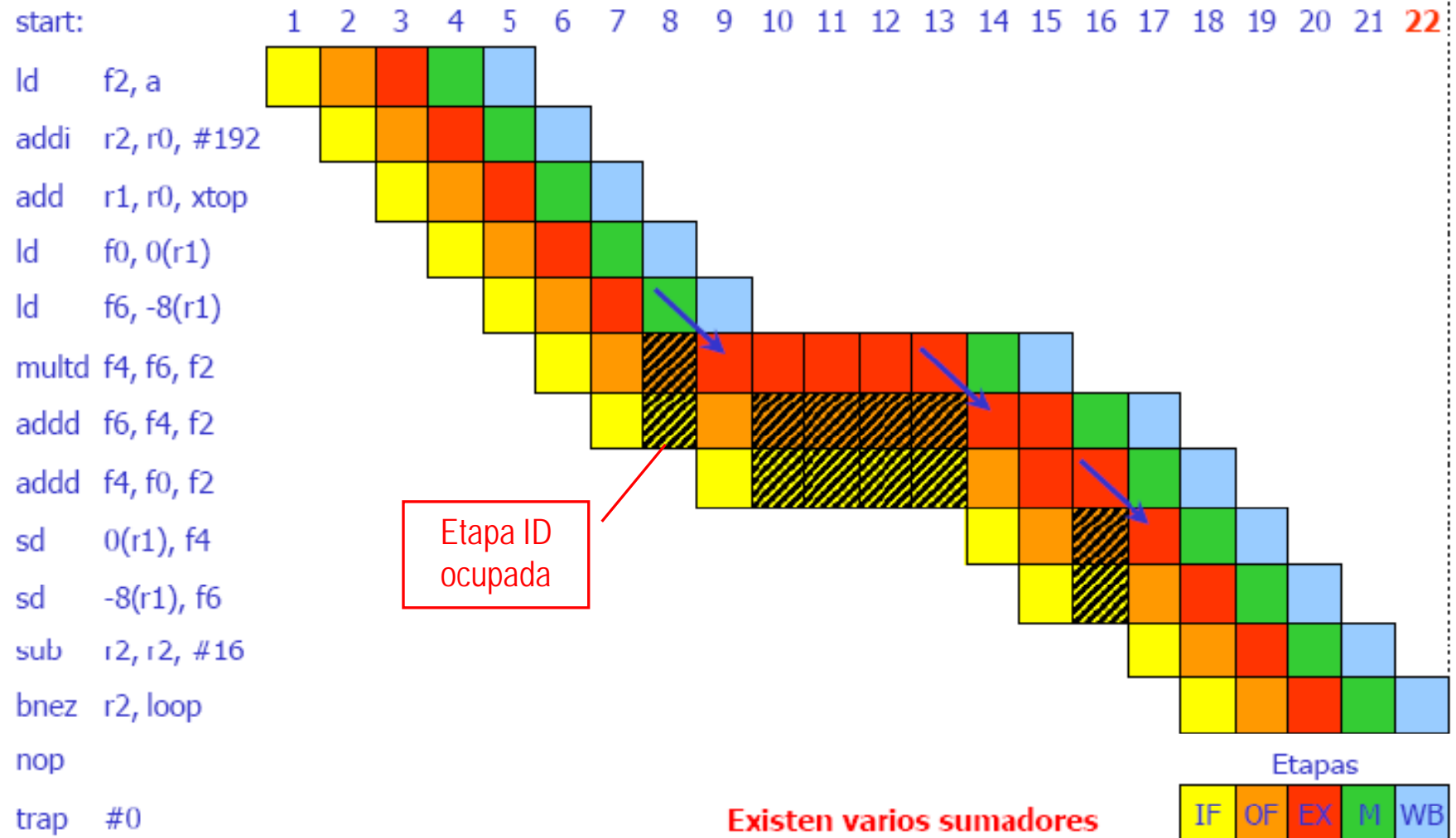
◆ Ejemplo emisión ordenada



◆ Ejemplo emisión desordenada

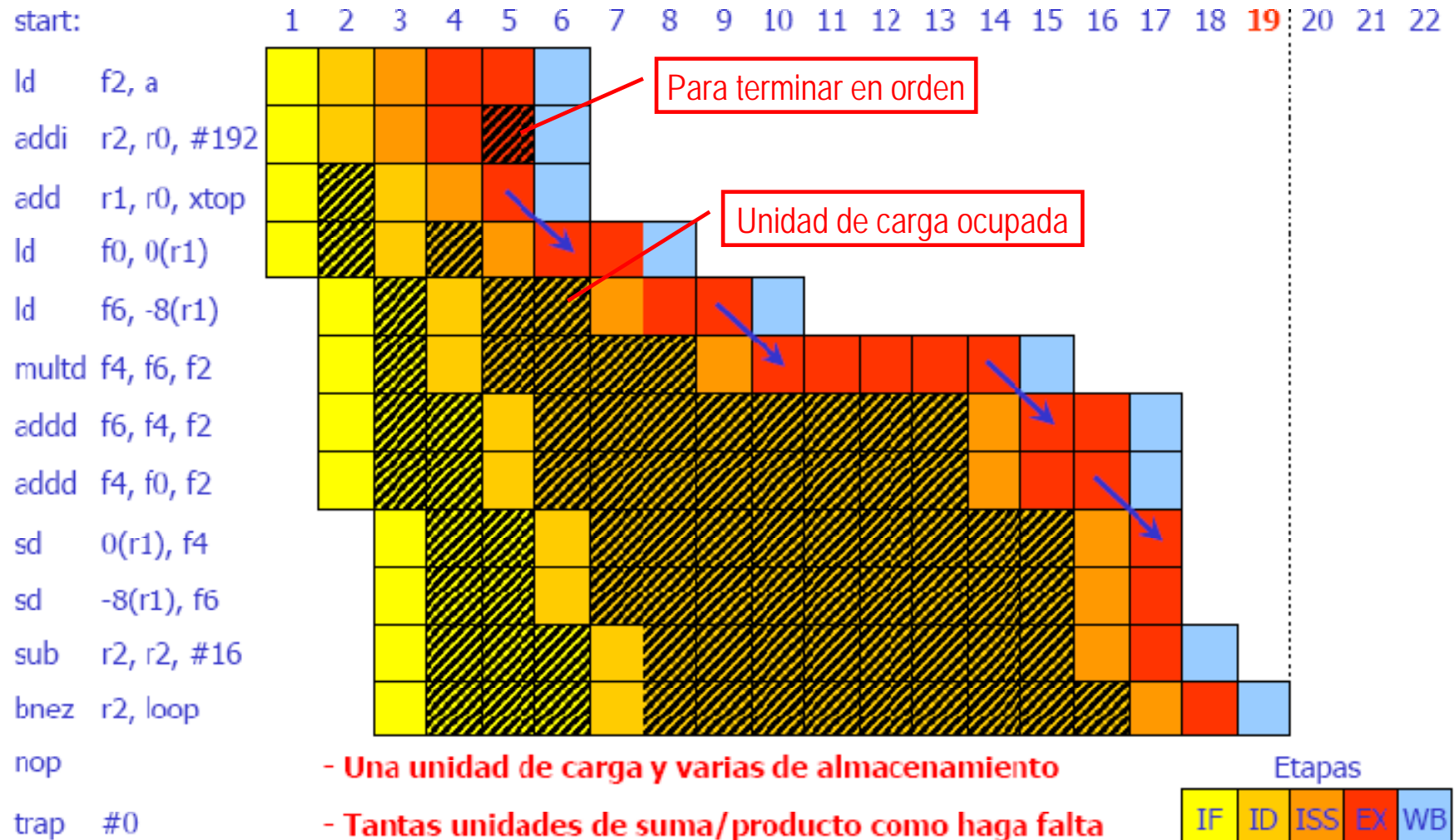


◆ Ejemplo: ejecución programa DLX en procesador segmentado

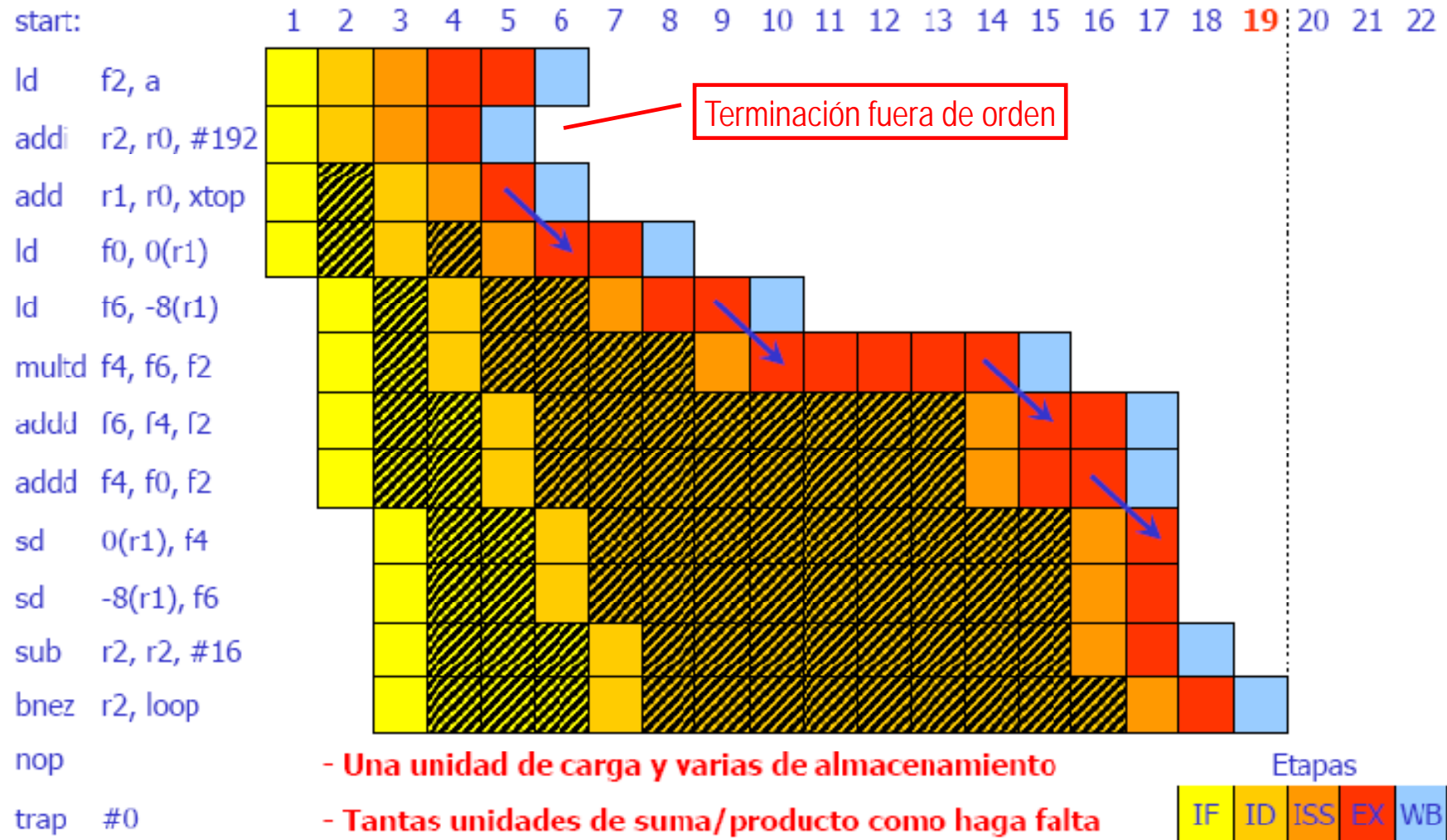


- ◆ Ejecución en superescalar: IF (4 inst/ciclo); ID (2 inst/ciclo); EX (tantas como hagan falta); WB (4)

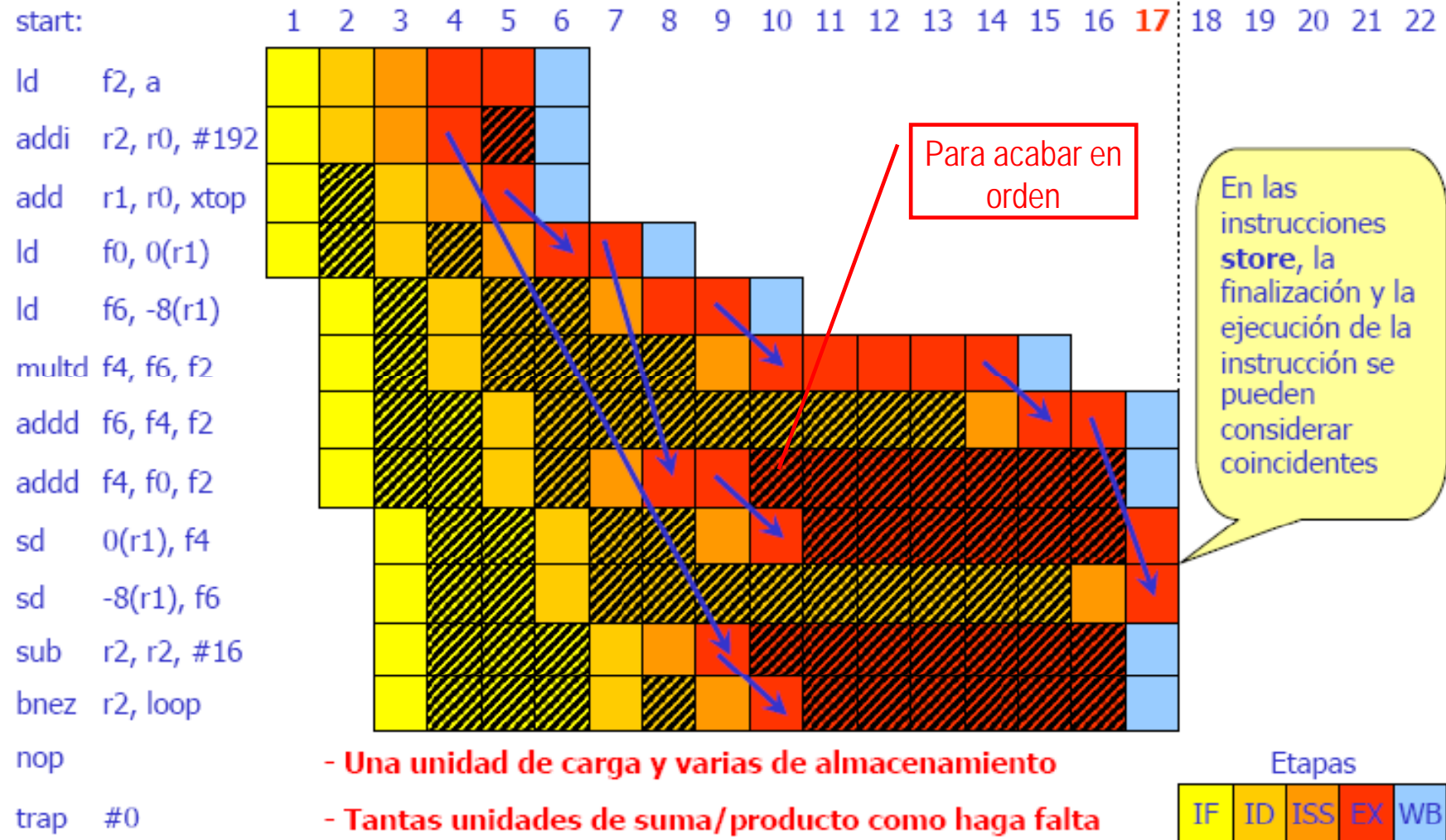
► Emisión ordenada/Finalización ordenada



► Emisión Ordenada/Finalización Desordenada



► Emisión Desordenada/Finalización ordenada



► Emisión Desordenada/Finalización Desordenada

