

### Arquitecturas con ILP (instruction level parallelism):

Ejecutan múltiples instrucciones concurrentemente o en paralelo:

Cores escalares segmentados, superescalares o VLIW

No tienen buffer de renombramiento porque la planificación de instrucciones la realiza el compilador.

### Procesadores superescalares y VLIW:

Los procesadores superescalares y VLIW comparten las siguientes características además del hecho de ser procesadores segmentados:

- Disponen de varias unidades de ejecución.
- Pueden ejecutar varias instrucciones simultáneamente en esas unidades de ejecución.
- Pueden emitir múltiples instrucciones en paralelo a unidades de ejecución.

En los procesadores superescalares es el hardware el que debe descubrir el paralelismo que se puede aprovechar a partir de las instrucciones que se van captando.

En los procesadores VLIW el paralelismo es explícito → se captan juntas las instrucciones que se van a emitir juntas a unidades de ejecución.

La arquitectura de los procesadores VLIW es más sencilla ya que es el compilador el que debe detectar paralelismo al seleccionar las instrucciones que se captarán juntas en la misma palabra de instrucción.

## Paralelismo entre instrucciones (ILP):

Depende de la frecuencia de las dependencias de datos y control, y del retardo de la operación (tiempo hasta que el resultado de una operación está disponible)

## Paralelismo de la máquina:

Determinado por el número de instrucciones que puedan captarse y ejecutarse al mismo tiempo (número de canales paralelos) y por la velocidad y los mecanismos que usa el procesador para encontrar las dependencias entre instrucciones.

## Ordenaciones en una secuencia de instrucciones:

En una secuencia de instrucciones se pueden distinguir 3 tipos de ordenaciones:

- El orden en que se captan las instrucciones.
- El orden en que se ejecutan
- El orden en que las instrucciones cambian los registros y la memoria.

El procesador superescalar debe ser capaz de identificar el paralelismo entre instrucciones que exista en el programa y organizar la captación, decodificación y ejecución de instrucciones en paralelo, utilizando el paralelismo de la máquina.

Cuanto más sofisticado sea un procesador superescalar, menos tiene que ajustarse a la ordenación de las instrucciones según se captan para su ejecución y para la modificación de los registros. (Para mejorar los tiempos de ejecución).

La única restricción es que el resultado del programa sea correcto.

Mejora de las prestaciones de los procesadores:

Más transistores por circuito integrado → Microarquitecturas más complejas en un solo CI: Paralelismo entre Instrucciones (Procesadores Superscalares)

Mejora de la Tecnología de Fabricación de CI basada en el Silicio → Reducción del tamaño de los transistores + Aumento del tamaño del dado

Se reduce la longitud de puerta del transistor y con ello el tiempo de conmutación → Mayores frecuencias de funcionamiento

$$V_{CPU} = IPC \times F$$

## Aspectos del procesamiento superscalar y etapas del cauce:

se hace tan pronto como se evita penalización entre saltos

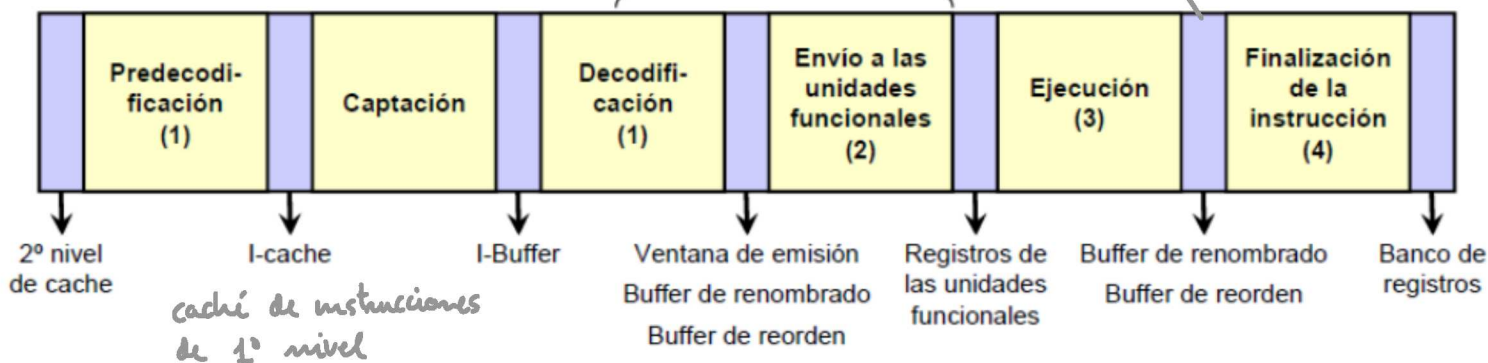
Detección y predicción de saltos (5)

Tratamiento de riesgos RAW (5)  
Renombrado de registros  
Ejecución especulativa de saltos  
Captura de operandos de los registros

Emisión (2)

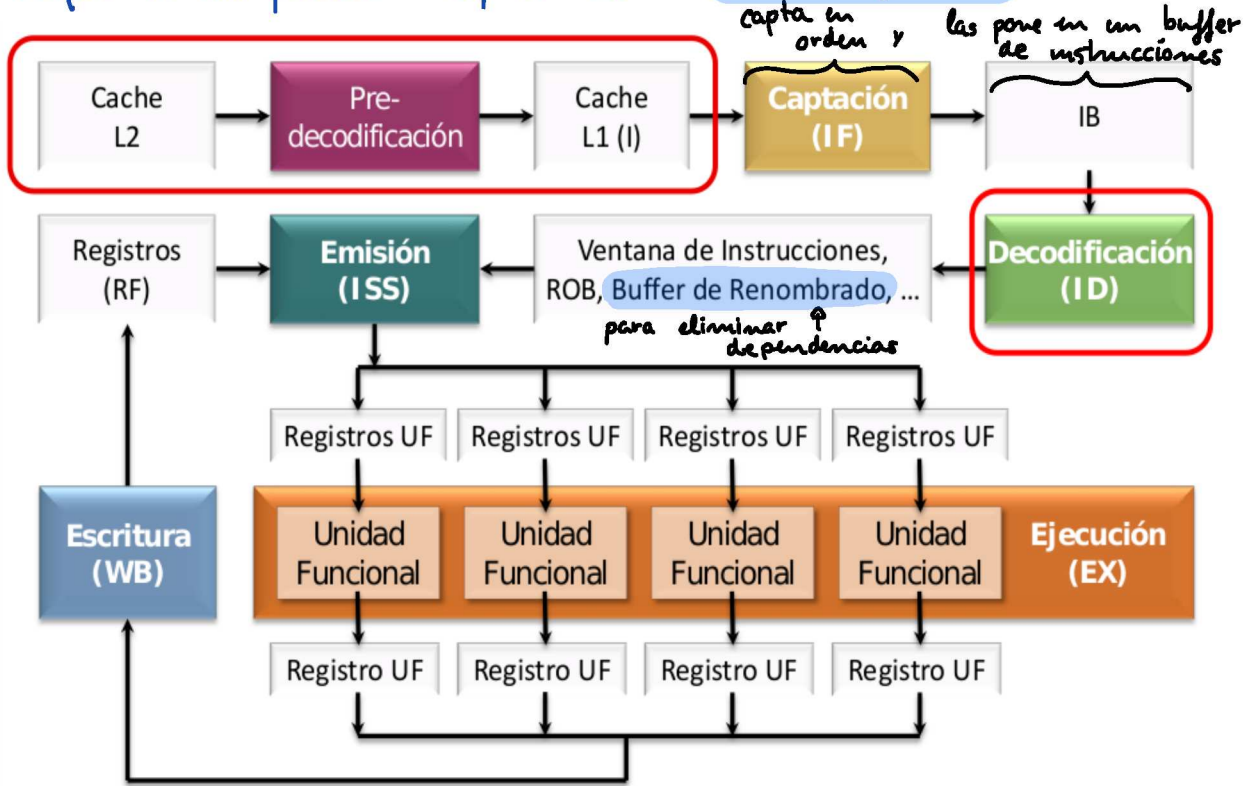
aquí se guardan los resultados

Mantenimiento de la consistencia (6)



- ① Decodificación paralela (decodificación a mayor velocidad → predecodificación)
  - La etapa de decodificación de un procesador superscalar debe ser capaz de decodificar varias instrucciones por ciclo.
- ② Emisión paralela de instrucciones a las unidades funcionales (Dependencias)
  - Una vez captadas las instrucciones, se almacenan en una cola en el mismo orden que se han captado.
- ③ Ejecución paralela de instrucciones en las unidades funcionales
- ④ Finalización del procesamiento de la instrucción.
- ⑤ Detección y predicción de saltos
- ⑥ Mantenimiento de la consistencia secuencial.

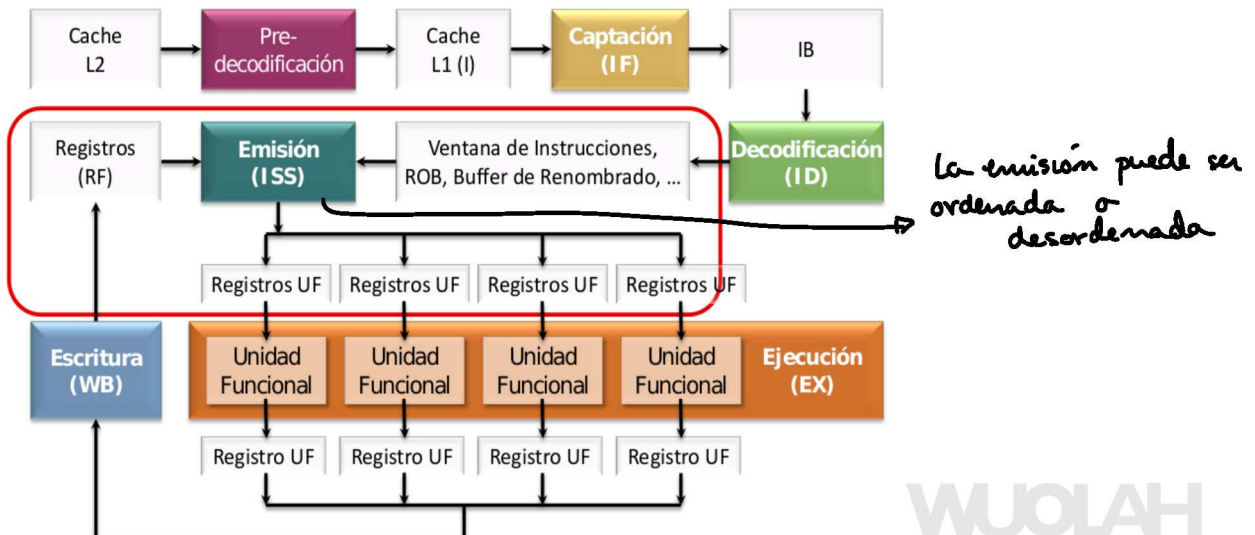
## Etapas de un procesador superescalar → **Predecodificación:**



Los bits que se añaden en la etapa de <sup>de</sup>predecodificación suelen indicar:

- Si es una instrucción de salto o no (se puede empujar su procesamiento antes)
- Si hace referencia a memoria o no
- El tipo de unidad funcional que va a utilizar (se puede emitir más rápidamente si hay chances para enteros o coma flotante...)

## Etapas de un procesador superescalar → **Emisión paralela de instrucciones:**





## Ventana de instrucciones:

la ventana de instrucciones almacena las instrucciones pendientes

Todas → ventana centralizada  
las de un tipo → ventana distribuida

las instrucciones se cargan en la ventana una vez decodificadas. Se utiliza un bit en función de si un operando está disponible o no:

- está disponible → se almacena el valor  
se indica el registro de donde se lee
- no está disponible → se almacena la UF desde donde llegará el operando.

Una instrucción puede ser emitida cuando tiene disponibles todos sus operandos y la unidad funcional donde se procesará.

## Ejemplo de Ventana de Instrucciones

#	opcode	address	rb_entry	operand1	ok1	operand2	ok2
2	MULTD	loop + 0x4	2	1	0	0	0
1	LD	loop	1	0	0	0	1

Dato no válido (indica desde dónde se recibirá el dato)

no está listo

está listo

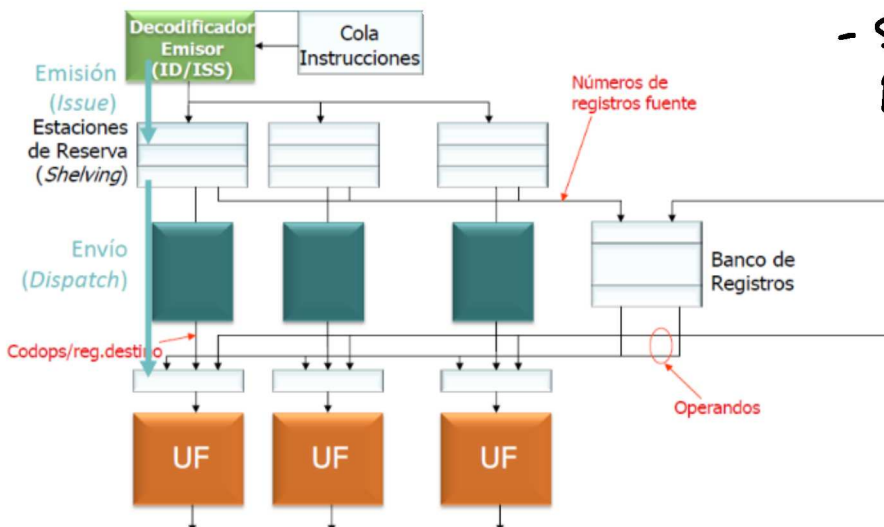
Lugar donde se almacenará el resultado

registro donde se guardará el dato cuando se calcule

Dato válido (en este caso es 0)

vamos a operar con un 0

## Estación de reserva:



- Si no existen limitaciones en el hardware las instrucciones se emiten a las estaciones de reserva independientemente de las dependencias.

- Las instrucciones esperan hasta que se resuelvan estas dependencias y se envían a

las UF una vez comprobada la disponibilidad de las mismas, cuando sea su turno.

# Alternativas para el envío a las unidades funcionales:

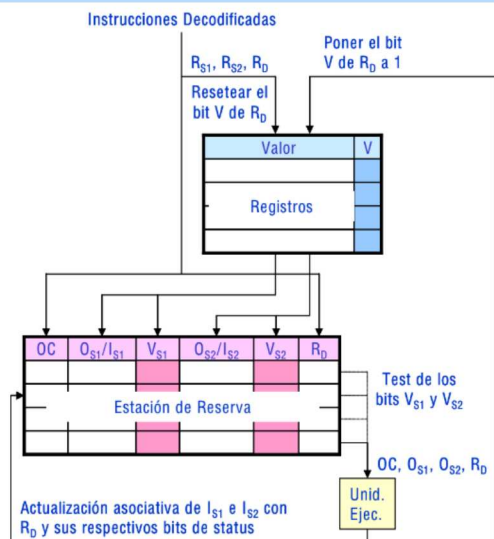
**Reglas de selección:** se determina las instrucciones que pueden enviarse  
LAS INSTRUCCIONES EJECUTABLES

**Reglas de arbitraje:** Instrucción que se envía si hay varias ejecutables.  
LA MÁS ANTIGUA ENTRE LAS INST. EJECUTABLES

**Orden de envío:** ciertas instrucciones no ejecutables bloquean instrucciones de un tipo, pero no de otros  
→ ORDENADAS, DESORDENADAS, PARCIALMENTE ORDENADAS

**Velocidad de envío:** Número de instrucciones que se envían por ciclo.  
(una por ciclo o varias por ciclo)

## Comprobación de los operandos:



Comprobación de los bits de validez en la estación de reserva.

$OC$  → Código de operación

$R_{S1}, R_{S2}$  → Registros fuente

$R_D$  → Registro de destino

$O_{S1}, O_{S2}$  → Operandos fuente

$I_{S1}, I_{S2}$  → Identificadores de los operandos fuente.

$V_{S1}, V_{S2}$  → Bits válidos de los operandos fuente.

## Renombramiento de los registros:

- Su implementación estática se da durante la compilación, pero también se puede implementar dinámicamente durante la ejecución con circuitos adicionales y registros extra

- Características de los buffer de renombrado:

- Tipos: Separados o mezclados con los registros de la arquitectura
- Número de buffer de renombrado
- Mecanismo de acceso:
  - indexado
  - asociativo.

- Velocidad de renombrado: Máximo número de nombres por ciclo que admite el procesador

- Los buffer de renombramiento permiten varias estructuras pendientes de un mismo

