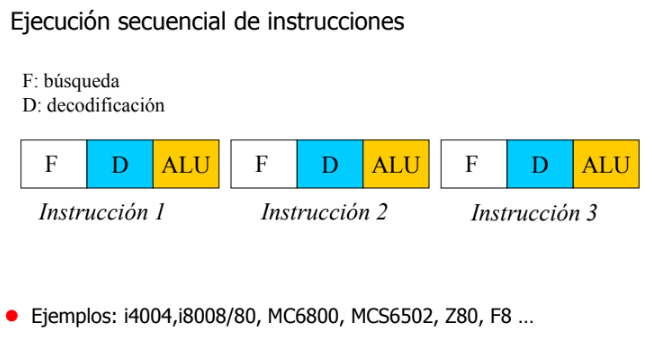
[Resumen clase 8](https://drive.google.com/file/d/1vItIsQ_lXxyBmId5hAvZE0ombofALnsz/view?usp=sharing)

Procesadores superescalares

# Procesador escalar

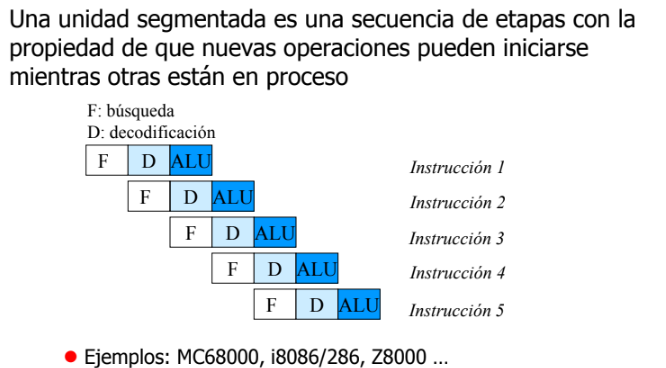
Ejecución secuencial de instrucciones



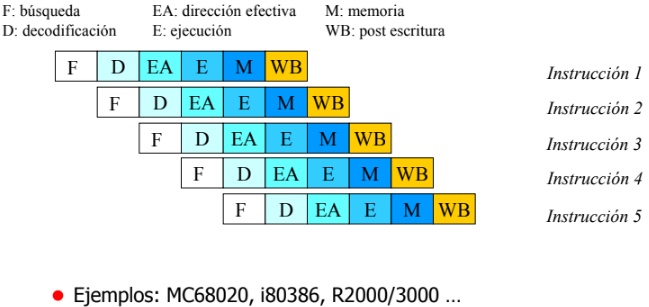
Ejemplos: i4004, i8008/80, MC6800, MCS6502, Z80, F8…

# 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



# Procesador escalar segmentado



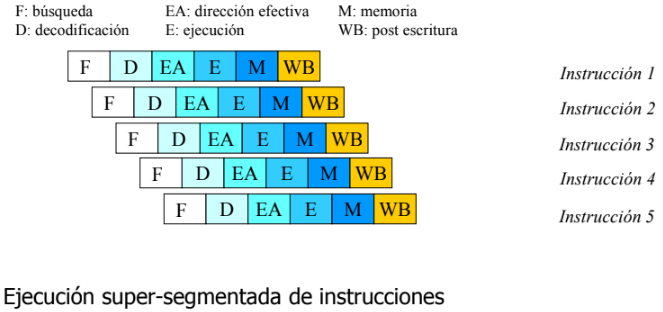
# Mayores prestaciones

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

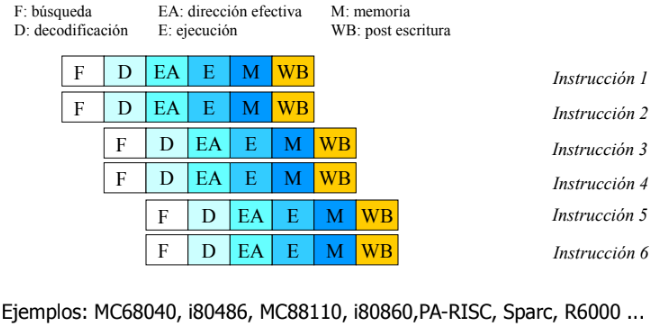
## Enfoque Supersegmentado

* 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 lo 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 de paralelismo
  + Incrementa la aceleración percibida

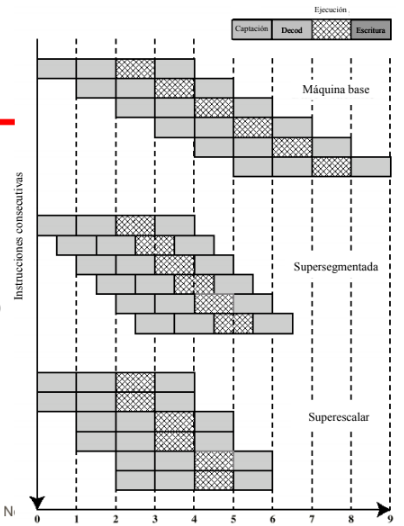


## Enfoque Superescalar

* 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



## Superescalar vs Supersegmentado

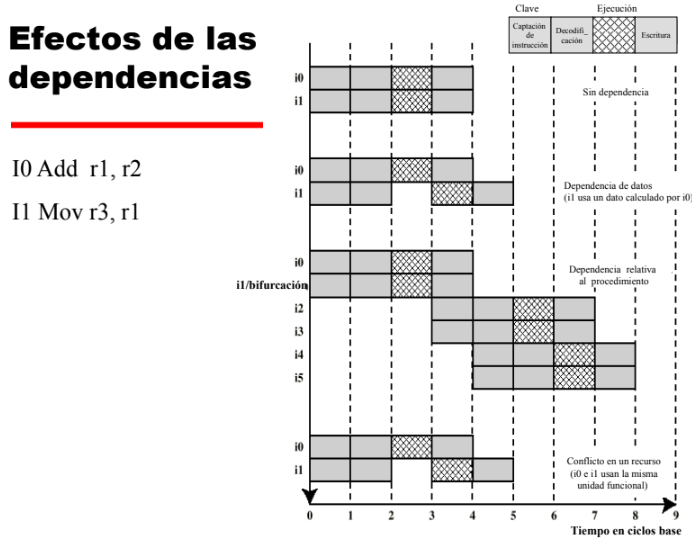


# Paralelismo de instrucciones

Existe cuando instrucciones de una secuencia son independientes y entonces pueden ejecutarse en paralelo solapandose

Limitaciones:

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



# 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 registro (dependencia de salida o antidependencia)
  + 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

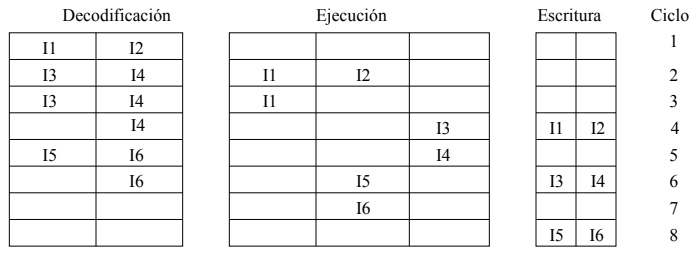
Supongase un cauce superescalar que:

* Capta y decodifica 2 instrucciones a la vez
* Posee 3 unidades funcionales independientes
* Posee 2 copias de la etapa de escritura

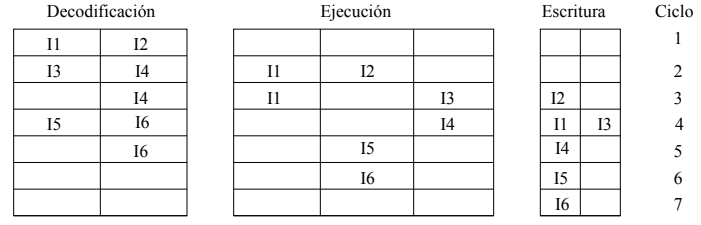
Supóngase un conjunto de 6 instrucciones con las siguientes características:

* 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

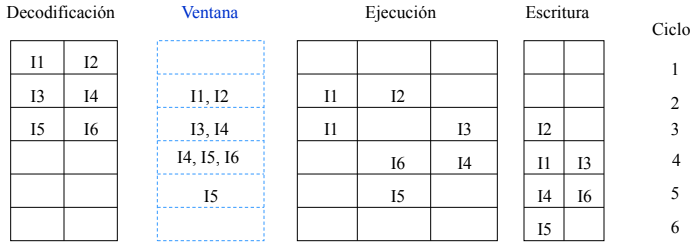
### Emisión y finalización en orden



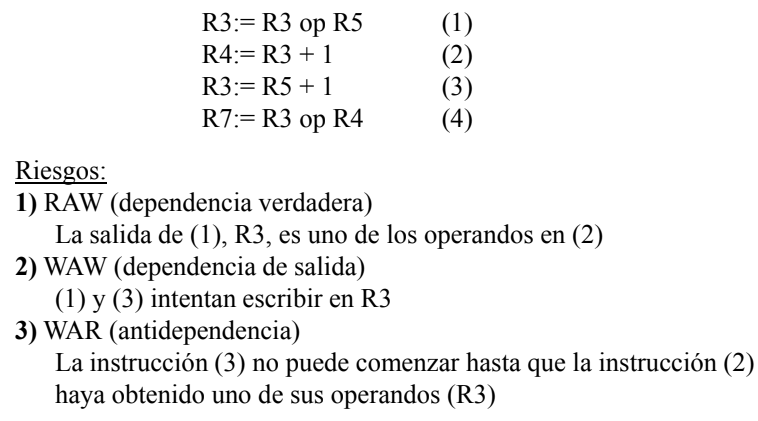
### Emisión en orden y finalización desordenada



### Emisión y finalización desordenada



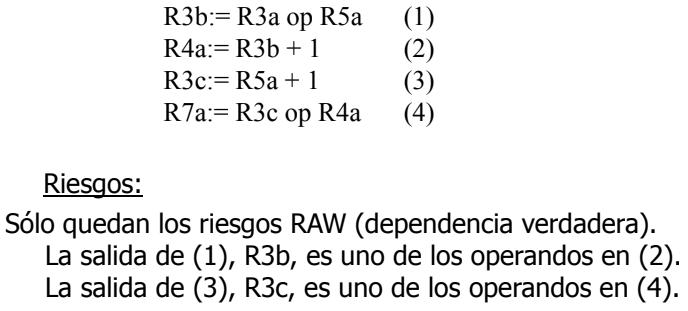
# Riesgos



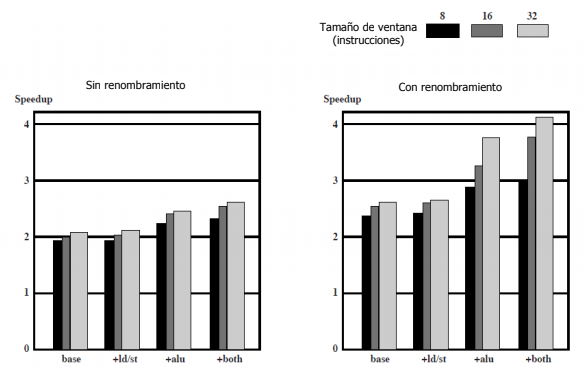
## Renombre de registros

* 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

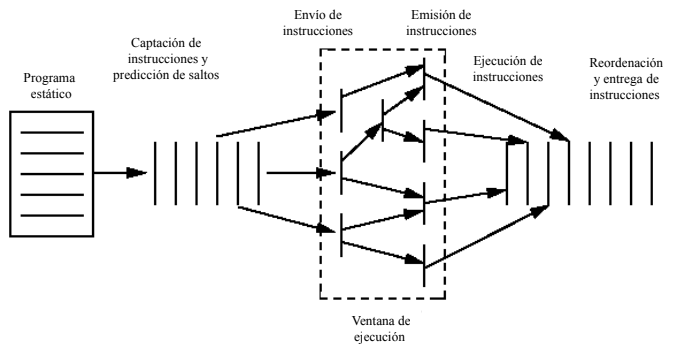
Ejemplo:



# Mejoras obtenibles en superescalar



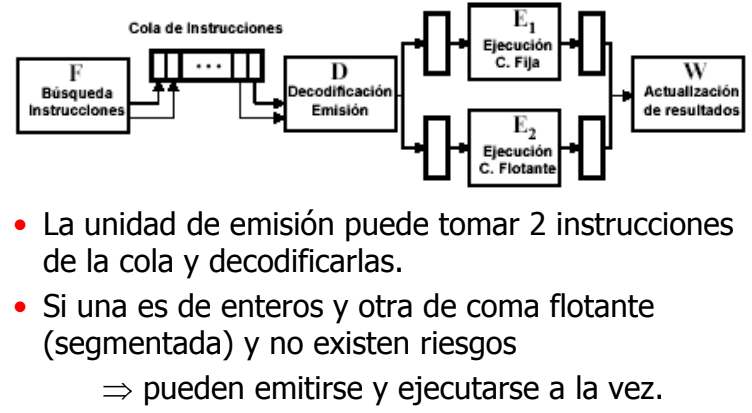
# Ejecución superescalar

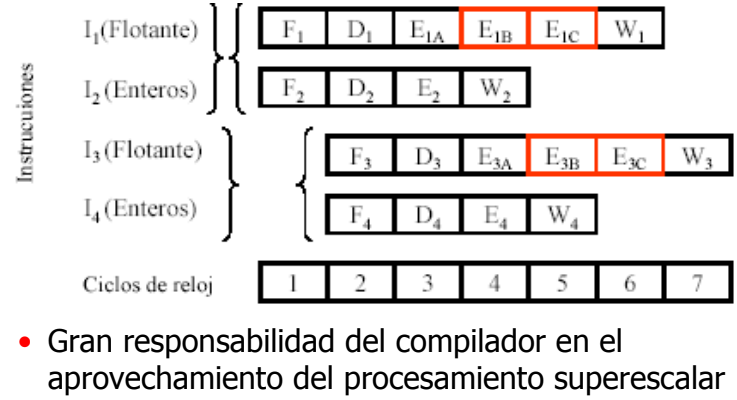


## Implementación superescalar

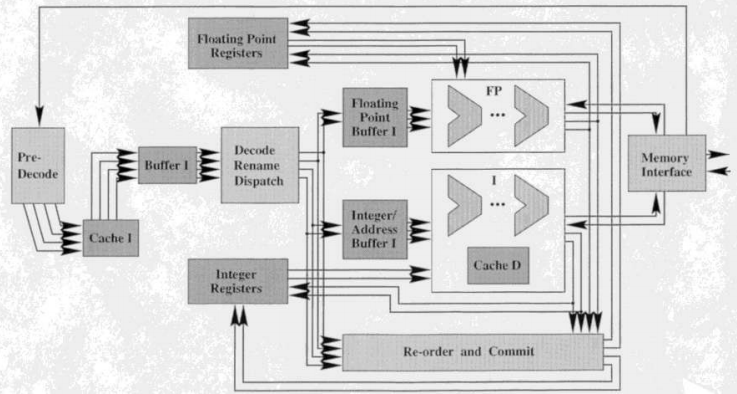
* Estrategias de captación simultánea de múltiples instrucciones
* Lógica para determinar dependencias verdaderas entre valores de registros y mecanismos para comunicar esos valores
* Mecanismos para iniciar o emitir múltiples instrucciones en paralelo
* Recursos para la ejecución en paralelo de múltiples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

## Ejemplo de procesador con 2 unidades de ejecución





## 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 I1 produce una excepción→Ha podido terminar I2?→ 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
* Excepciones precisas
  + Por ejemplo: los resultados se almacenan en el orden en que aparecen las instrucciones (En ejemplo anterior: retardar el W2 hasta el ciclo 6)
  + Con las interrupciones externas→ excepciones precisas
  + La unidad de emisión deja de emitir y se cancela la cola
  + Todas las instrucciones pendientes se completan→ comienza el procesamiento de la interrupción
* Compromiso entre
  + Ejecución desordenada ←→ Liberar las unidades de ejecución
  + Completar instrucciones en orden ←→ Excepciones precisas
  + Posible solución:
    - Emisión desordenada y finalización ordenada → Etapa de escritura temporal (TW) **Renombramiento de registros**.
    - El registro temporal TWi se usa solo por las instrucciones posteriores a i

