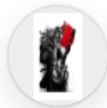


# WUOLAH



Bigbounze

[www.wuolah.com/student/Bigbounze](http://www.wuolah.com/student/Bigbounze)



24319

## Tema 4.pdf

*Resúmenes Por Temas*



2º Arquitectura de Computadores



Grado en Ingeniería Informática



Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación  
UGR - Universidad de Granada



## MÁSTER EN FINANZAS

¿Quieres alcanzar el **éxito profesional**?

■ Título Oficial

■ Prácticas Profesionales



Semana de Formación en Londres

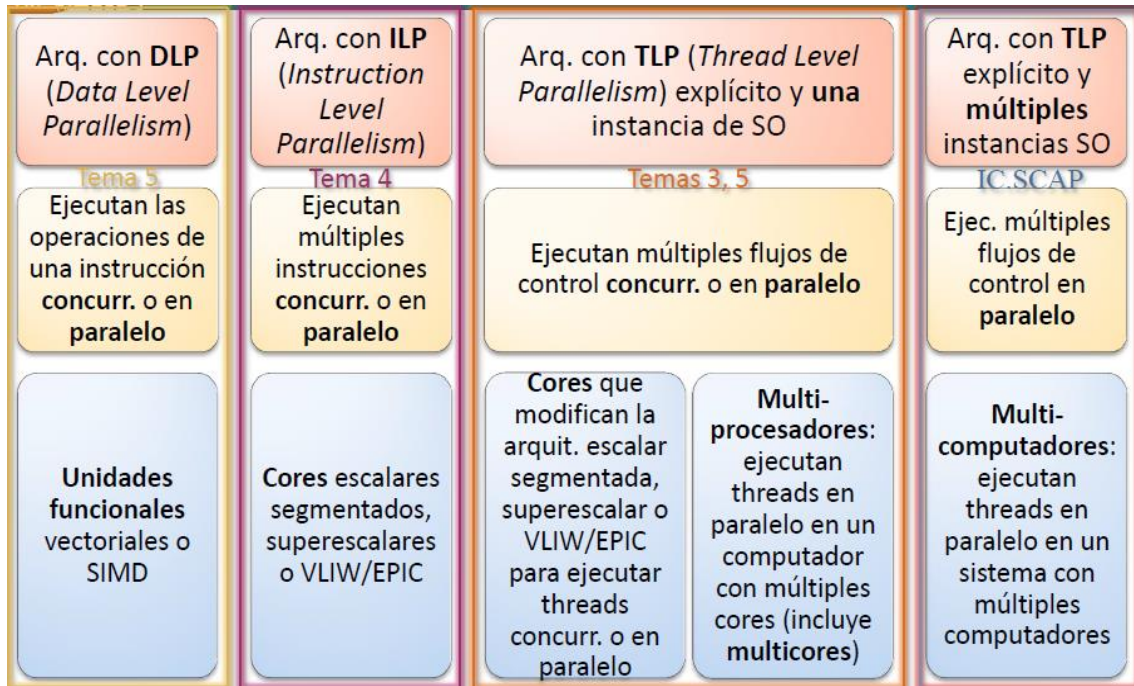


[www.cunef.edu](http://www.cunef.edu)

## Tema 4

### - Lección 11

#### . Arquitecturas con DLP, ILP y TLP (thread=flujo de control)



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

## Encuentra tu nota para este examen

Una carrera académica sobresaliente se merece la mejor salida profesional. Elige tu Master en Escuela Internacional de Gerencia e impulsa tu futuro.

Hacemos gente de empresa.

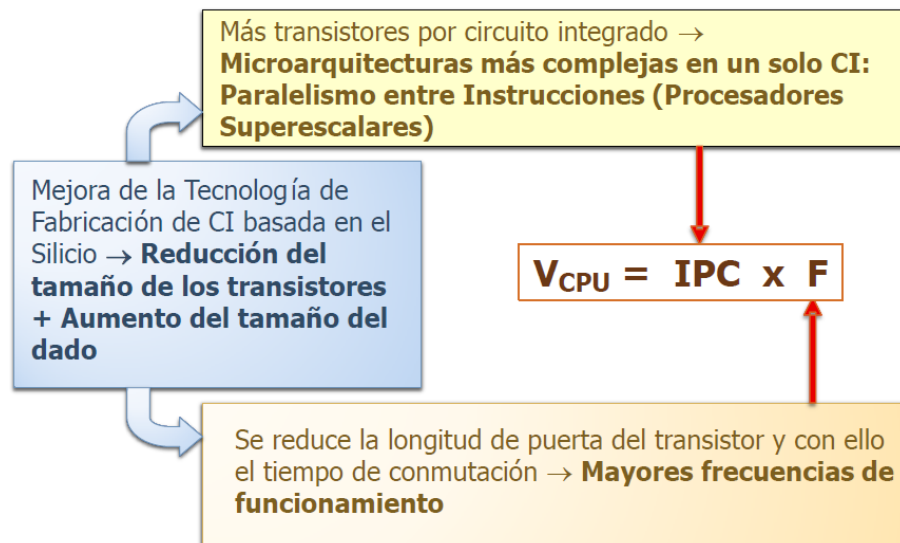
Programas Master | 95% Alumnos en prácticas | 80% Alumnos trabajando

     | C/ Eduardo Molina Fajardo, 38, Granada | Infórmate: 958 222 194 | [esgerencia.com](http://esgerencia.com)

  
BUSINESS & MARKETING SCHOOL

ESCUELA  
INTERNACIONAL  
DE GERENCIA 

## . Mejora de las Prestaciones de los Procesadores



## . Paralelismo entre instrucciones (ILP) y paralelismo de la máquina

- Paralelismo entre instrucciones: 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 esté disponible).
- Paralelismo de la máquina: determinado por el número de instrucciones que pueden captarse y ejecutarse al mismo tiempo (número de cauces paralelos) y por la velocidad y los mecanismos que usa el procesador para encontrar las dependencias entre instrucciones.

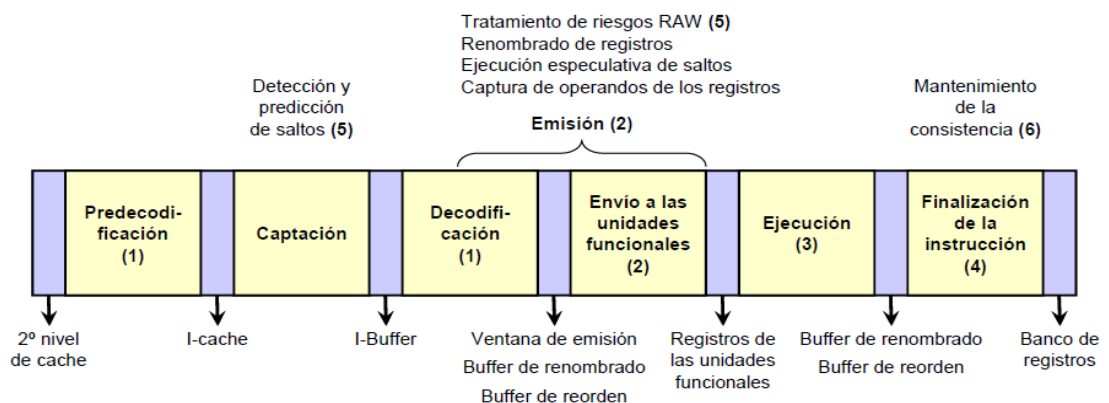
## . Ordenaciones en una secuencia de instrucciones

- En una secuencia de instrucciones se pueden distinguir tres tipos de ordenaciones:
  - o El orden en que se captan las instrucciones.
  - o El orden en que se ejecutan.
  - o 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 eficazmente los recursos existentes (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 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.

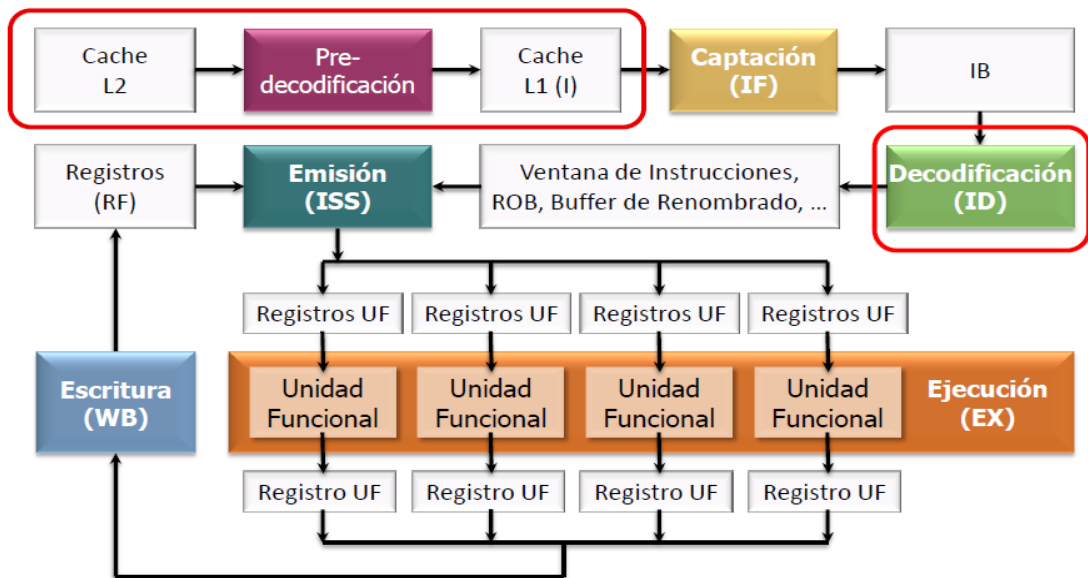


## . Aspectos del Procesamiento Superscalar y Etapas del Cauce

- 1- Decodificación Paralela (Decodificación a mayor velocidad → Precodificación)
  - a. 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)
  - a. Una vez captadas las instrucciones, se almacenan en una cola en el mismo orden en el que se han captado
- 3- Ejecución Paralela de Instrucciones a 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 Superscalar: Precodificación



Los bits que se añaden en la etapa de precodificación (bits de precodificación) suelen indicar:

- Si es una instrucción de salto o no (se puede empezar su procesamiento antes)



- El tipo de unidad funcional que va a utilizar (se puede emitir más rápidamente si hay cauces para enteros o como flotante...)
- Si hace referencia a memoria o no

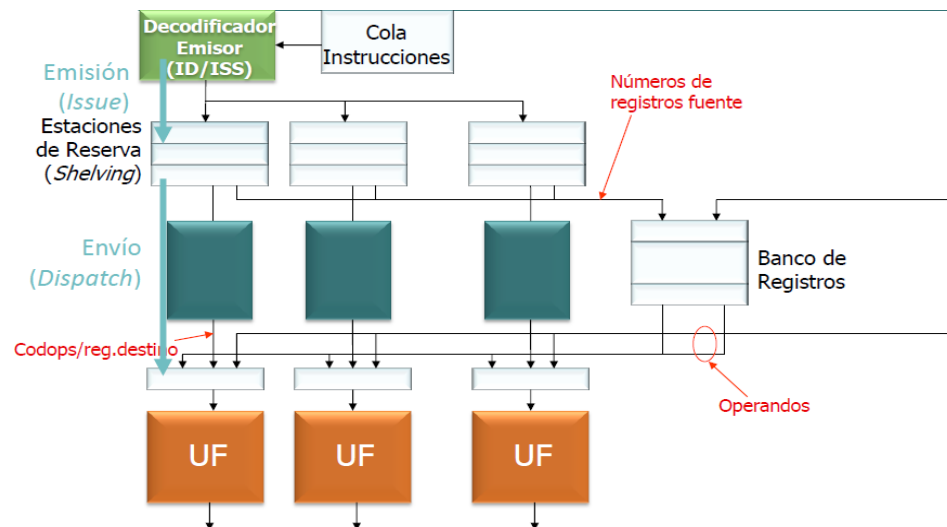
## . Ventana de Instrucciones

La ventana de instrucciones almacena las instrucciones pendientes. Todas si la ventana es centralizada, o las de un tipo, si en distribuida.

Las instrucciones se cargan en la ventana una vez decodificadas. Se utiliza un bit para indicar si un operando está disponible (se almacena la unidad funcional desde donde llegará el operando).

Una instrucción puede ser emitida cuando tiene todos sus operandos disponibles y la unidad funcional donde se procesará. Hay diversas posibilidades para el caso en el que varias instrucciones estén disponibles.

## . Estación de Reserva



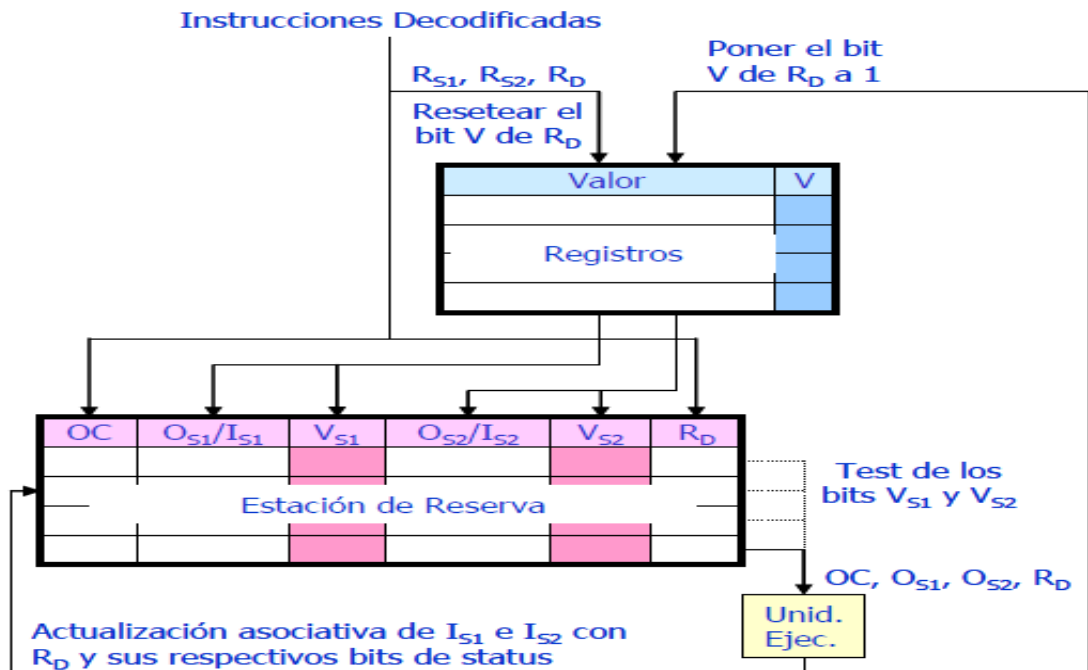
Si no existen limitaciones en el hardware las instrucciones se emiten a las estaciones de reserva independientemente de las dependencias. Las instrucciones esperan hasta que se resuelvan las dependencias y se envían a las unidades funcionales una vez comprobada la disponibilidad de la unidad, cuando sea su turno.

## . Alternativas para el Envío a las Unidades Funcionales

- Reglas de Selección: Se determina las instrucciones que pueden enviarse (instrucciones ejecutables).

- Reglas de Arbitraje: Instrucción que se envía si hay varias ejecutables. La más antigua entre las ejecutables tendrá prioridad.
- Orden de Envío: Ordenadas, Desordenadas o Parcialmente ordenadas.
- Velocidad de Envío: Número de instrucciones que se envían por ciclo. Hay dos alternativas, una por ciclo o varias por ciclo.

## . Comprobación de los Operandos

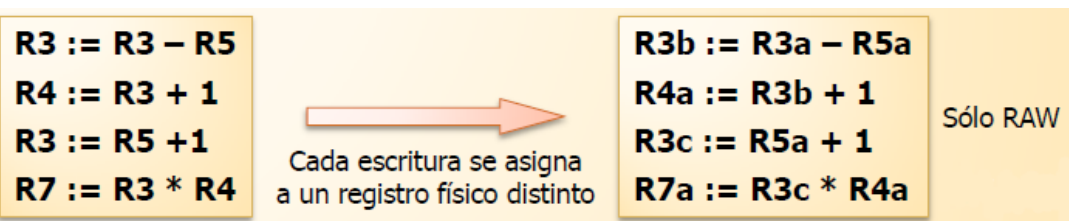


Comprobación de los bits de validez en la ventana o estación de reserva.

- $O_C$ : 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 Registros

Técnica para evitar el efecto de las dependencias WAR y WAW.



Su implementación estática se da durante la compilación, pero también se puede implementar dinámicamente durante la ejecución con circuitería adicional y registros extra.

Características de los Buffers de Renombrado son:

- Tipos de Buffers: separados o mezclados con los registros de la arquitectura.
- Número de Buffers de Renombrado.
- Mecanismos para acceder a los Buffers: asociativos o indexados.

Se denomina velocidad de renombrado al número de nombres asignados por ciclo que admite el procesador.

Los Buffers de Renombramiento permiten varias escrituras pendientes de un mismo registro.

## - Lección 12

### . Consistencia

En el Procesamiento de una Instrucción se puede distinguir entre:

- El final de la ejecución de la operación codificada en las instrucciones: Se dispone de los resultados generados por las UF pero no se han modificado los registros de la arquitectura
- El final del procesamiento del procesamiento de la Instrucción: se escriben los resultados de la Operación en los registros de la Arquitectura.

La consistencia de un programa se refiere al orden en que las instrucciones se completan y el orden en que se accede a memoria para leer o escribir.

Cuando se ejecutan instrucciones en paralelo, el orden en que termina esa ejecución puede variar según el orden que las correspondientes instrucciones tenían en el programa, pero debe existir consistencia entre el orden en que se completan las instrucciones y el orden secuencial que tienen en el código de programa.

<b>Consistencia de Procesador</b>	Débil: Las instrucciones se pueden completar desordenadamente siempre que no se vean afectadas las dependencias	Deben detectarse y resolverse las dependencias	Power1 (90) PowerPC 601 (93) Alpha R8000 (94) MC88110 (93)
	Fuerte: Las instrucciones deben completarse estrictamente en el orden en que están en el programa	Se consigue mediante el uso de ROB	PowerPC 620 PentiumPro (95) UltraSparc (95) K5 (95) R10000 (96)
<b>Consistencia de Memoria</b>	Débil: Los accesos a memoria (Load/Stores) pueden realizarse desordenadamente siempre que no afecten a las dependencias	Deben detectarse y resolverse las dependencias de acceso a memoria	MC88110 (93) PowerPC 620 UltraSparc (95) R10000 (96)
	Fuerte: Los accesos a memoria deben realizarse estrictamente en el orden en que están en el programa	Se consigue mediante el uso del ROB	PowerPC 601 (93) E/S 9000 (92)

Tendencia



Tendencia / Prestaciones



## . Reordenamiento Load/Store

Las instrucciones LOAD y STORE implican cambios en el Procesador y en Memoria

**LOAD:**

- Cálculo de Dirección en ALU o Unidad de Direcciones
- Acceso a Cache
- Escritura del Dato en Registro

**STORE:**

- Cálculo de Dirección en ALU o Unidad de Direcciones
- Esperar que esté disponible el dato a almacenar (en ese momento acaba)

**La Consistencia de Memoria Débil** (reordenación de los accesos a memoria):

- **'Bypass' de Loads/Stores:**

Los Loads pueden adelantarse a los Stores pendientes y viceversa (siempre que no se violen dependencias)

- **Permite los Loads y Stores Especulativos:**

Cuando un Load se adelanta a un Store que le precede antes de que se haya determinado la dirección se habla de Load especulativo. Igual para un Store que se adelanta a un Load o a un Store.

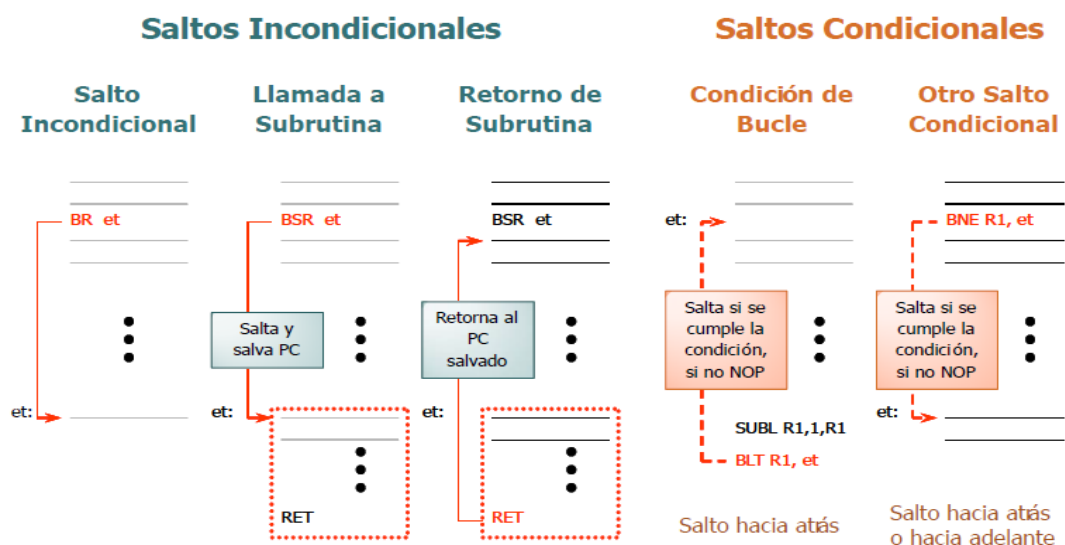
- **Permite ocultar las Faltas de Cache:**

Si se adelanta un acceso a memoria a otro que dio lugar a una falta de cache y accede a Memoria Principal.

## . Buffer de Reordenamiento (ROB)

- El puntero de cabecera apunta a la siguiente posición libre y el puntero de cola a la siguiente instrucción a retirar.
- Las instrucciones se introducen en el ROB en orden de programa estricto y pueden estar marcadas como emitidas, en ejecución o finalizada.
- Las instrucciones sólo se pueden retirar si han finalizado, y todas las que les preceden también.
- La consistencia se mantiene porque sólo las instrucciones que se retiran del ROB se completan y se retiran en el orden estricto de programa.

## . Clasificación de los saltos





## . Alternativas para la condición de salto

- Estado del resultado
  - o Existen bits de Estado que se modifican al realizar operaciones o mediante operaciones que comprueban específicamente el valor de los registros
- Comprobación directa
  - o Los resultados de las operaciones se comprueban directamente respecto a las condiciones específicas mediante instrucciones específicas.

## . Aspectos del Procesamiento de Saltos en un procesador Superescalar

- Detección de la Instrucción de salto: cuanto antes se detecte que una instrucción es de salto menor será la posible penalización. Los saltos se detectan usualmente en la fase de decodificación e incluso en la captación si hay predecodificación.
- Gestión de los Saltos Condicionales no Resueltos: si en el momento en que la instrucción de salto evalúa la condición de salto ésta no se haya disponible se dice que el salto o la condición no se ha resuelto. Para resolver este problema se suele utilizar el procesamiento especulativo de salto.
- Acceso a las instrucciones destino del salto: hay que determinar la forma de acceder a la secuencia a la que se produce el salto.

El efecto de los saltos en los procesadores superescalares es más pernicioso ya que, al emitirse varias instrucciones por ciclo, prácticamente en cada ciclo puede haber una instrucción de salto.

## . Gestión de Saltos Condicionales no resueltos

Uso de los ciclos que siguen a la inst. de salto condicional	Salto Retardado	Se utilizan los ciclos que siguen a la captación de una instrucción de salto para insertar instrucciones que deben ejecutarse independientemente del resultado del salto  (Primeras arquitecturas RISC y posteriores)
<b>Gestión de Saltos Condicionales no Resueltos</b>  (Una condición de salto no se puede comprobar si no se ha terminado de evaluar)	Bloqueo del Procesamiento del Salto	Se bloquea la instrucción de salto hasta que la condición esté disponible  (68020, 68030, 80386)
	Procesamiento Especulativo de los Saltos	La ejecución prosigue por el camino más probable (se especula sobre las instrucciones que se ejecutarán). Si se ha errado en la predicción hay que recuperar el camino correcto.  (Típica en los procesadores superescalares actuales)
	Múltiples Caminos	Se ejecutan los dos caminos posibles después de un salto hasta que la condición de salto se evalúa. En ese momento se cancela el camino incorrecto.  (Máquinas VLIW experimentales: Trace/500, URPR2)
Evitar saltos condicionales	Ejecución Vigilada ( <i>Guarded Exec.</i> )	Se evitan los saltos condicionales incluyendo en la arquitectura instrucciones con operaciones condicionales  (IBM VLIW, Cydra-5, Pentium, HP PA, Dec Alpha)

## . Esquema de Predicción de Salto

- Predicción Fija: se toma siempre la misma decisión: el salto siempre se realiza. “taken”, o no, “not taken”.
- Predicción Verdadera: la decisión de si se realiza o no se realiza el salto se toma mediante:
  - Predicción Estática: según los atributos de la instrucción de salto.
    - Basada en el Código de Operación: para ciertos códigos de operación se predice que el salto se toma y para otros que no.
    - Basada en el Desplazamiento del Salto: si el desplazamiento es positivo se predice que no se toma el salto en caso contrario se predice que se toma.
    - Dirigida por el compilador: el compilador es el que establece la predicción fijando, para cada instrucción, el valor de un bit específico que existe en la instrucción de salto.
  - Predicción dinámica: según el resultado de ejecuciones pasadas de la instrucción. La predicción para cada instrucción de salto puede cambiar cada vez que va a ejecutar ésta según la historia previa de saltos tomados/no-tomados para dicha instrucción. El presupuesto básico de la predicción dinámica es que es más probable que el resultado de una instrucción de salto sea similar al que se tuvo en la última. Presenta mejores prestaciones de predicción, aunque su implementación es más costosa.
    - Predicción Dinámica Implícita: no hay bits de historia propiamente dichos sino que se almacena la dirección de la instrucción que se ejecutó después de la instrucción de salto en cuestión
    - Predicción Dinámica Explícita: para cada instrucción de salto existen unos bits específicos que codifican la información de historia de dicha instrucción de salto

## . Extensión del procesamiento Especulativo

Tas la predicción, el procesador continúa ejecutando instrucciones especulativamente hasta que se resuelve la condición. El intervalo de tiempo entre el comienzo de la ejecución especulativa y la resolución de la condición puede variar considerablemente y ser bastante largo. En los procesadores superescalares, pueden emitir varias instrucciones por ciclo, pueden aparecer más instrucciones de salto condicional no resueltas durante la ejecución especulativa. Si el número de instrucciones que se ejecutan especulativamente es muy elevado y la predicción es incorrecta, la penalización es mayor. Así, cuanto mejor es el esquema de predicción mayor puede ser el número de instrucciones ejecutadas especulativamente.

- Nivel de Especulación: número de instrucciones de salto condicional sucesivas que pueden ejecutarse especulativamente.

- Grado de Especulación: hasta qué etapa se ejecutan las instrucciones que siguen en un camino especulativo, después de un salto.

## **. Recuperación de Predicción Incorrecta**

La recuperación de una predicción comprende:

- Descartar los resultados de la ejecución especulativa.
- Continuar la ejecución de la secuencia de instrucciones alternativa (la correcta)
- Recuperación desde un salto efectuado:
  - o El procesador debe guardar la dirección de la instrucción siguiente a la de salto para utilizarla si la predicción es incorrecta.
  - o La recuperación es más rápida si no se descartan las instrucciones que se habían precaptado junto con la de salto.
- Recuperación cuando no se ha saltado:
  - o Pre-calcular la dirección de salto y almacenarse para permitir la recuperación.
  - o La recuperación es más rápida si se precaptan instrucciones de la secuencia que empieza a partir de la dirección a la que se salta.

## **. Acceso a la secuencia de salto**

Si se detecta una instrucción de salto, se calcula su dirección de destino para acceder a la posición de memoria correspondiente si se produce el salto. Los saltos condicionales efectuados son más frecuentes que los no efectuados. Por ello, sería interesante reducir al máximo el tiempo de acceso a la secuencia de instrucciones a partir de la dirección de salto y reducir la penalización para las predicciones incorrectas de los saltos efectuados. La rapidez de acceso a la secuencia de instrucciones que empieza en la dirección a donde se salta es fundamental para mejorar las prestaciones del esquema de gestión de los saltos condicionales.

## **. Instrucciones de Ejecución Condicional**

Se pretende reducir el número de instrucciones de salto incluyendo en el repertorio máquina instrucciones con operaciones condicionales. Estas instrucciones tienen dos partes, la condición, denominada guardia, y la parte de operación.

## **- Lección 13**

### **. Características generales de los procesadores VLIW**

Cada palabra de instrucción larga codifica varias operaciones que se ejecutan simultáneamente en distintas Unidades Funcionales.

## . ILP Hardware vs ILP Software

- ILP intensivo en Hardware:
  - o Capaz de tener en cuenta los eventos que se producen dinámicamente durante la ejecución.
  - o Mayor portabilidad de los códigos entre plataformas y mejor aprovechamiento de la memoria.
- ILP intensivo en Software
  - o Capaz de aprovechar la mayor visibilidad del código que tiene el compilador.
  - o Mayor simplicidad en el hardware y menor consumo de energía.

## . El papel del compilador

- VLIW: Planificación estática. Necesita asistencia del compilador, que puede realizar renombrados, reorganizaciones de código, etc., para mejorar el uso de los recursos disponibles, el esquema de predicción de saltos.
- Superescalar: Planificación dinámica. Requiere menos asistencia del compilador, pero más coste hardware. Facilita la portabilidad del código entre la misma familia de procesadores.

El compilador construye paquetes de instrucciones sin dependencias, de forma que el procesador no necesita comprobarlas explícitamente.

## . Planificación estática local y global

- Planificación local: actúa sobre un bloque básico mediante desarrollo de bucles y planificación de las instrucciones del cuerpo aumentado del bucle.
- Planificación global: actúa considerando bloques de código entre instrucciones de salto
- Desenrollado de bucles:
  - o Al desenrollar un bucle se crean bloques básicos más largos, lo que facilita la planificación local de sus sentencias.
  - o Además de disponer de más sentencias, éstas suelen ser independientes, ya que operan sobre diferentes datos.
- Segmentación software:
  - o Se reorganizan los bucles de forma que cada iteración del código transformado contiene instrucciones tomadas de distintas iteraciones del bloque original.
  - o De esta forma se separan las instrucciones dependientes en el bucle original entre diferentes iteraciones del bucle nuevo.





## . Instrucciones con predicado

- Reducen el número de saltos condicionales. Esto es bastante importante sobre todo si no hay una opción más frecuente que otra en el salto.
- El uso general de predicados es muy útil en planificación global ya que puede eliminar todos los saltos condicionales que no sean de control de bucle.
- Las instrucciones de movimiento condicional de datos son las más utilizadas, aunque pueden ser ineficientes si se dispone sólo de ellas para transformar trozos de código largos que dependen de saltos.