



## Riesgos de Control

- Problemas con el Control de Flujo
- Saltos en el MIPS R3000
- Métodos de Resolución
  - ✓ Reducción de la Latencia
  - ✓ Predicción de Salto
  - ✓ Salto Retardado

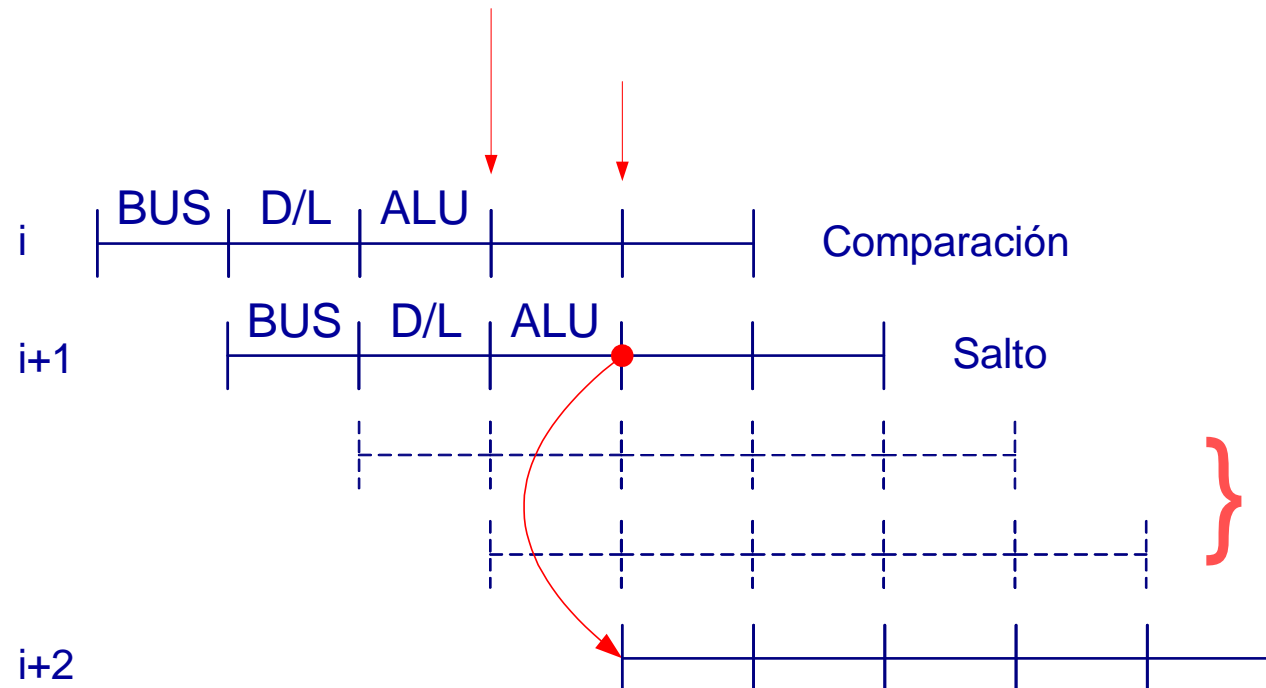


## Problemas con el Control de Flujo

- Provocados por las rupturas en la ejecución secuencial de los programas:
  - ✓
  - ✓
- Se traducen en una indeterminación a la hora de emitir las siguientes Instrucciones a ejecutar mientras no se conoce el nuevo punto de ejecución (el resultado del salto)



## Saltos en el MIPS R3000



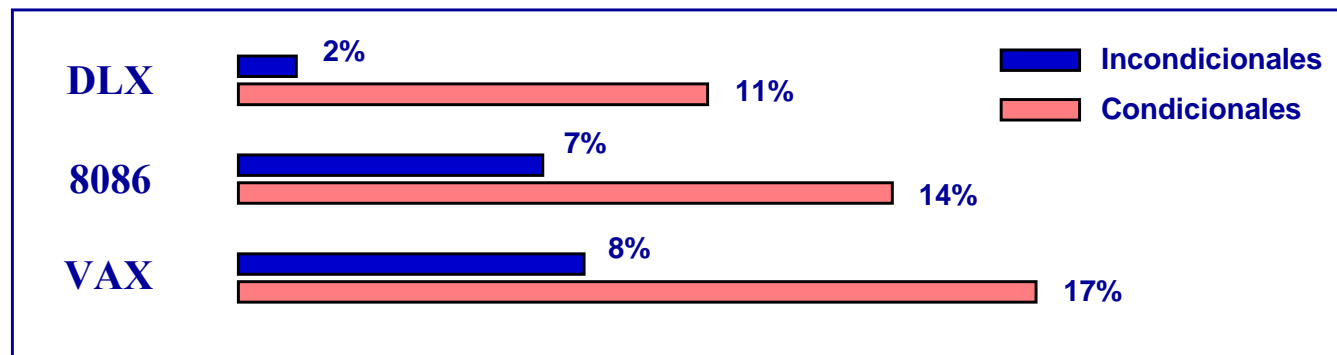
✓ La dirección efectiva del salto no se conoce hasta después de dos ciclos

(

✓ Las instrucciones de salto representan entre el 15 y el 30% del total de instrucciones



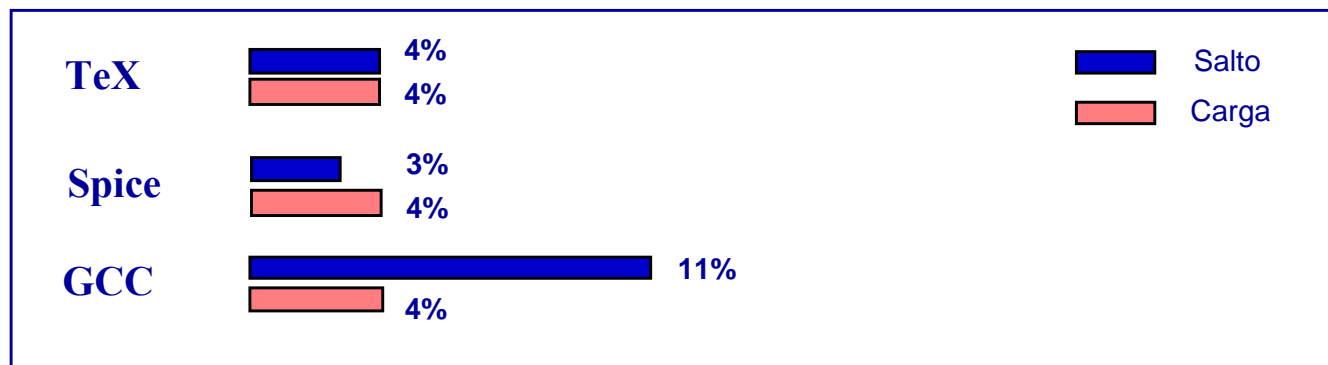
## Instrucciones de Control



*Frecuencia de instrucciones de control en varias arquitecturas  
(53% de saltos condicionales efectivos en DLX)*



## Retardos de Salto



*Porcentaje de los ciclos de reloj en los retardos (de carga y salto) frente a los consumidos por ejecución de Instrucciones*



## Métodos de Resolución

- Parada del cauce (
  - Reducción de la Latencia
    - ✓ Anticipar las operaciones que se puedan dentro del proceso de ejecución de las instrucciones de salto
  - Efectuar la búsqueda de Instrucciones por las dos ramas
    - ✓ Ahorro del ciclo de búsqueda de instrucción una vez resuelto el salto
  - Predicción de Salto
    - ✓ Estática
    - ✓ Dinámica
  - Salto Retardado
    - ✓ Hacer visible la latencia al nivel de lenguaje máquina
- (

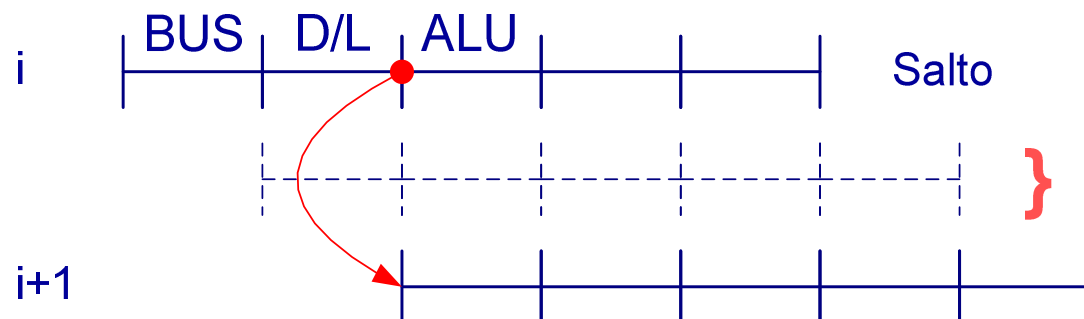


## Reducción de la Latencia

- Averiguar antes si el salto es o no efectivo
- Calcular antes la dirección de destino del salto

Ejemplo:

- ✓ Instrucción de salto que chequea únicamente si el valor de un registro es 0  
⇒
- ✓ Adición de un sumador a la etapa D/L  
⇒
- ✓ **Resultado:** reducción de la latencia (HUECO DE RETARDO DE SALTO) a





## Predicción de Salto

- Tipos de predicción:
  - ✓ Estática (
  - ✓ Dinámica (
- Predicción SALTO NO EFECTIVO
  - ✓ Se continúa la búsqueda de instrucciones normalmente
  - ✓ Si el salto NO es finalmente efectivo ⇒
  - ✓ Si el salto es finalmente efectivo ⇒
- Predicción SALTO EFECTIVO
  - ✓ Se calcula la dirección destino y se comienza la búsqueda a partir de ella
  - ✓ Si el salto es finalmente efectivo ⇒
  - ✓ Si el salto NO es finalmente efectivo ⇒
  - ✓ Inútil en nuestro cauce (
  - ✓ De aplicación en caso de tener condiciones de salto complejas





# Predicción SALTO NO EFECTIVO

Ciclo de Reloj

1 2 3 4 5 6 7 8 9 ...

**Salto NO Efec.**

BUS	D-L	EJE	MEM	ESC					
	BUS	D-L	EJE	MEM	ESC				
		BUS	D-L	EJE	MEM	ESC			
			BUS	D-L	EJE	MEM	ESC		
				BUS	D-L	EJE	MEM	ESC	

Instrucción i+1

Instrucción i+2

Instrucción i+3

Instrucción i+4

Ciclo de Reloj

1 2 3 4 5 6 7 8 9 ...

**Salto Efectivo**

BUS	D-L	EJE	MEM	ESC					
	BUS	BUS	D-L	EJE	MEM	ESC			
			BUS	D-L	EJE	MEM	ESC		
				BUS	D-L	EJE	MEM	ESC	
					BUS	D-L	EJE	MEM	ESC

Instrucción i+1

Instrucción i+2

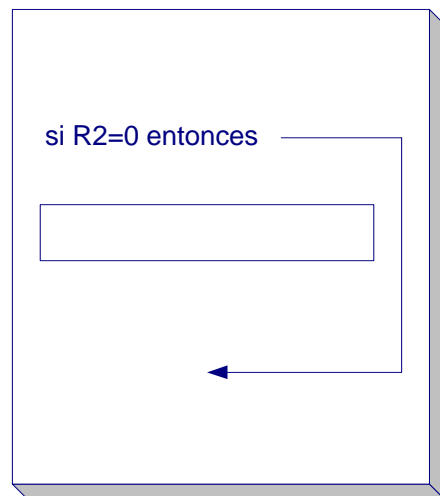
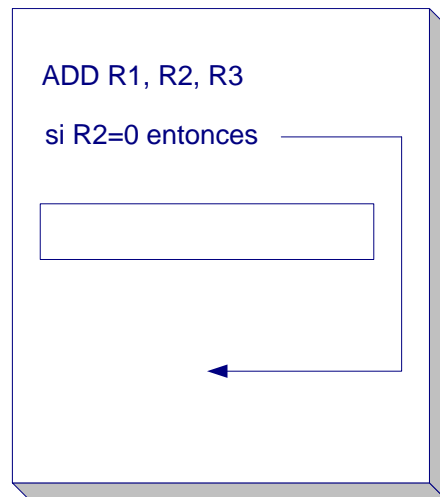
Instrucción i+3

Instrucción i+4



## Salto Retardado

### (1) Desde Antes



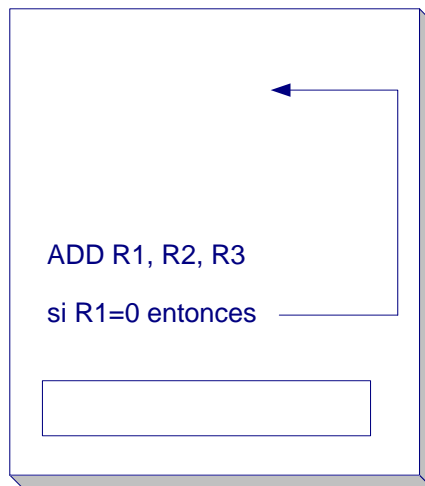
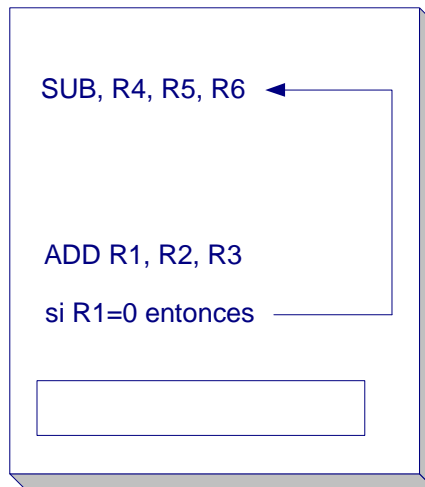
- ✓ El hueco se rellena con instrucciones anteriores a la de salto
- ✓ Los saltos NO deben depender de las instrucciones replanificadas





## Salto Retardado

### (2) Desde Destino



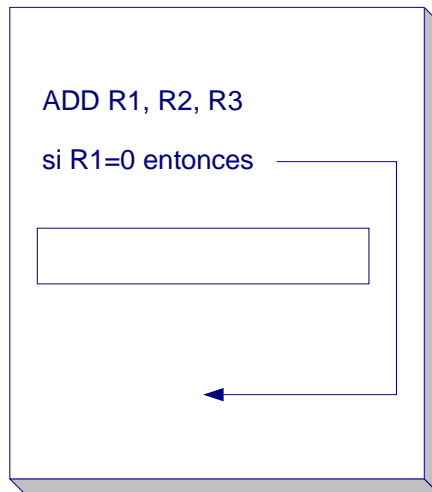
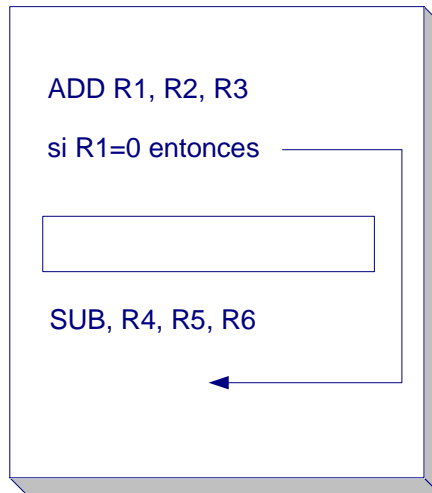
- ✓ El hueco se rellena con instrucciones a partir de la dirección de salto
- ✓ La ejecución de las instrucciones replanificadas cuando NO es efectivo el salto solo debe implicar trabajo desperdiciado, NUNCA resultados incorrectos
- ✓ Puede ser necesario duplicar instrucciones





## Salto Retardado

### (3) Desde Siguiente

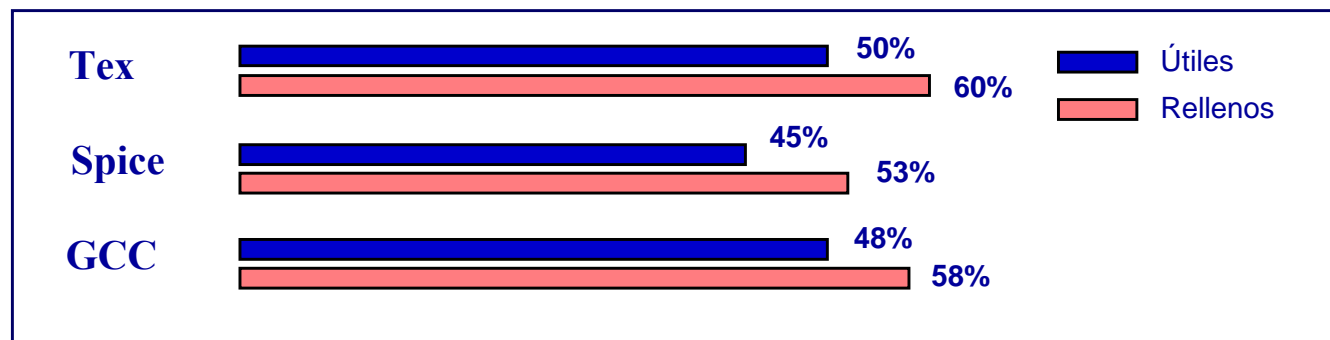


- ✓ El hueco se rellena con instrucciones que siguen a la de salto
- ✓ La ejecución de las instrucciones replanificadas cuando es efectivo el salto solo debe implicar trabajo desperdiciado, **NUNCA** resultados incorrectos





## Salto Retardado



*Frecuencia con la que se rellenan los huecos de retardo de salto y frecuencia con la que son útiles para la computación*



## Problema de Riesgos de Control

A partir de las gráficas y datos relativos a riesgos de control para la arquitectura del procesador segmentado visto en clase, y considerando un CPI ideal sin riesgos de 1, se pide:

- 1) ¿Cuál es la frecuencia total de instrucciones de control?
- 2) ¿Qué porcentaje de las instrucciones de control cambia realmente el contador de programa?
- 3) ¿Qué porcentaje medio de saltos retardados hace realmente trabajo útil?
- 4) Rellenar la tabla siguiente relativa a los costes asociados a varios esquemas de planificación de salto:

<b>Esquema de planificación</b>	<b>Penalización de salto</b>	<b>CPI efectivo o real</b>	<b>Aceleración de la segmentación</b>	<b>Aceleración respecto a la estrategia de detención</b>
1 - Detención	2			
2 - Predicción "salto efectivo"	1			
3 - Predicción "salto no efectivo"	1			
4 - Salto retardado	1			



## Solución:

1) (11% de saltos condicionales + 2% de saltos incondicionales)

2)

53% de saltos condicionales efectivos  $\rightarrow$  total de saltos efectivos =  $(11 * 0.53 + 2)/13 = 0.6$

3)

frecuencia media de saltos retardados que realizan trabajo util =  $(0.50 + 0.45 + 0.48) / 3 = 0.48$

4) Aceleración =  $CPI\ Ideal * Prof.\ Seg. / (CPI\ Ideal + frecuencia\ saltos * penalización\ salto)$   
=  $1 * 5 / CPI\ real$   
=  $5 / CPI\ real$

Detención  $\rightarrow CPI\ real = 1 + 0.13 * 2 = 1.26 \rightarrow Aceleración = 5 / 1.26 =$   
Pred. efectivo  $\rightarrow CPI\ real = 1 + 0.13 * 1 = 1.13 \rightarrow Aceleración =$   
Pred. no efectivo  $\rightarrow CPI\ real = 1 + (0.13 * 0.60) * 1 = 1.08 \rightarrow Aceleración =$

Salto retardado:

Frecuencia media de saltos retardados que provocan penalización =  
1 - frecuencia media de saltos retardados que realizan trabajo util =  $1 - 0.48 = 0.52$

$\rightarrow CPI\ real = 1 + (0.13 * 0.52) * 1 = 1.07 \rightarrow Aceleración =$

Aceleración de la estrategia n respecto a la 1 = Aceleración de n / Aceleración de 1:

$\rightarrow Aceleración\ 1/1 = 3.97 / 3.97 =$   
Aceleración 2/1 =  $4.43 / 3.97 =$   
Aceleración 3/1 =  
Aceleración 4/1 =