

### Apartado a)

Como la instrucción de salto conoce la dirección de destino al final de la etapa MEM hay un retardo de tres huecos por lo que hay que colocar tres instrucciones NOP (Figura 1.33 del libro de teoría).

```
i1: LD    R1, X(R7)
i2: ADDI R1, R1, #1
i3: BEQZ Rtest, i6
i*: NOP
i*: NOP
i*: NOP
i4: ADD  R2, R1, R2
i5: SUB  R2, R2, R3
i6: SD   0(R8), R5
```

Con independencia de que el salto sea o no efectivo, las instrucciones que se han colocado en el hueco de retardo siempre van a comenzar a ejecutarse e introducirse en la segmentación hasta que se conozca el destino del salto. Por ello, es importante colocar instrucciones que sean independientes del resultado del salto, o que se ejecuten con una alta probabilidad (por ejemplo, en un bucle se ejecutarán las n veces que itere el bucle pero habrá una vez que se ejecute de forma errónea, esto, al salir del bucle) de forma que el tener que deshacer los cambios (si los hubiese) que ha introducido la ejecución especulativa compense el tener que deshacerlos.

Si el bucle es efectivo la secuencia dinámica de ejecución es la siguiente:

```
i1: LD    R1, X(R7)
i2: ADDI R1, R1, #1
i3: BEQZ Rtest, i6
i*: NOP
i*: NOP
i*: NOP
i6: SD   0(R8), R5
```

La segmentación quedaría:

i1: LD	R1, X(R7)	IF	ID	EX	MEM	WB												
i2: ADDI	R1, R1, #1		IF	ID	---	EX	MEM	WB										
i3: BEQZ	Rtest, i6			IF	---	ID	EX	MEM	WB									
i*: NOP						IF	ID	EX	MEM	WB								
i*: NOP							IF	ID	EX	MEM	WB							
i*: NOP								IF	ID	EX	MEM	WB						
i6: SD	0(R8), R5								IF	ID	EX	MEM	WB					

Un ejemplo claro de aprovechamiento del hueco de retardo es imaginar que en lugar de saltar a i6 se saltase a i1 durante 1000 veces. Esas 3 NOPS se podrían reemplazar por instrucciones del cuerpo del bucle pero habría una vez en que se ejecutarían de forma especulativa, es decir, al salir del bucle.

## Apartado b)

Una posible solución muy optimizada es la siguiente:

i3: BEQZ Rtest, ix  
i1: LD R1, X(R7)  
i6: SD 0(R8), R5  
i2: ADDI R1, R1, #1  
i4: ADD R2, R1, R2  
i5: SUB R2, R2, R3  
ix: NOP

Efectivo:

i3: BEQZ R4, ix	IF	ID	EX	MEM	WB														
<b>i1: LD R1, X(R7)</b>		<b>IF</b>	<b>ID</b>	<b>EX</b>	<b>MEM</b>	<b>WB</b>													
<b>i6: SD 0(R8), R5</b>			<b>IF</b>	<b>ID</b>	<b>EX</b>	<b>MEM</b>	<b>WB</b>												
<b>i2: ADDI R1, R1, #1</b>				<b>IF</b>	<b>ID</b>	<b>EX</b>	<b>MEM</b>	<b>WB</b>											
ix: NOP					IF	ID	EX	MEM	WB										

No efectivo:

i3: BEQZ R4, ix	IF	ID	EX	MEM	WB														
<b>i1: LD R1, X(R7)</b>		<b>IF</b>	<b>ID</b>	<b>EX</b>	<b>MEM</b>	<b>WB</b>													
<b>i6: SD 0(R8), R5</b>			<b>IF</b>	<b>ID</b>	<b>EX</b>	<b>MEM</b>	<b>WB</b>												
<b>i2: ADDI R1, R1, #1</b>				<b>IF</b>	<b>ID</b>	<b>EX</b>	<b>MEM</b>	<b>WB</b>											
i4: ADD R2, R1, R2					IF	ID	EX	MEM	WB										
i5: SUB R2, R2, R3						IF	ID	EX	MEM	WB									
i7: NOP							IF	ID	EX	MEM	WB								

### **Apartado c)**

Debido a que R1 depende de la ejecución de las instrucciones antes del salto, no es posible realizar la optimización reorganizando el código. El resultado sería el del apartado a)