

Ampliación de Estructura de Computadores

Soluciones de los ejercicios del Tema 3

Aquí se muestra la solución final de algunos de los ejercicios propuestos en el boletín del tema 3. No se muestra el desarrollo completo en todos los casos. El objetivo es que el alumno pueda comprobar si ha obtenido el resultado correcto una vez que ha realizado él mismo el ejercicio.

Ejercicio 1

Apartado a

r1	r2	Salto	Último salto	Predicción	Comportamiento	Nueva Predicción
0	100	S1	NT	NT/NT → NT	T	T/NT
1	99	S3	T	NT/NT → NT	T	NT/T
1	99	S1	T	T/NT → NT	NT	T/NT
0	99	S2	NT	NT/NT → NT	T	T/NT
0	98	S3	T	NT/T → T	T	NT/T
0	98	S1	T	T/NT → NT	T	T/T
1	97	S3	T	NT/T → T	T	NT/T
1	97	S1	T	T/T → T	NT	T/NT
0	97	S2	NT	T/NT → T	T	T/NT
0	96	S3	T	NT/T → T	T	NT/T
0	96	S1	T	T/NT → NT	T	T/T
1	95	S3	T	NT/T → T	T	NT/T
1	95	S1	T	T/T → T	NT	T/NT
0	95	S2	NT	T/NT → T	T	T/NT
0	94	S3	T	NT/T → T	T	NT/T

Apartado b

r1	r2	Salto	Últimos saltos	Predicción	Comportamiento	Nueva Predicción
0	100	S1	NT,NT	NT/NT/NT/NT → NT	T	T/NT/NT/NT
1	99	S3	NT,T	NT/NT/NT/NT → NT	T	NT/T/NT/NT
1	99	S1	T,T	T/NT/NT/NT → NT	NT	T/NT/NT/NT
0	99	S2	T,NT	NT/NT/NT/NT → NT	T	NT/NT/T/NT
0	98	S3	NT,T	NT/T/NT/NT → T	T	NT/T/NT/NT
0	98	S1	T,T	T/NT/NT/NT → NT	T	T/NT/NT/T
1	97	S3	T,T	NT/T/NT/NT → NT	T	NT/T/NT/T
1	97	S1	T,T	T/NT/NT/T → T	NT	T/NT/NT/NT
0	97	S2	T,NT	NT/NT/T/NT → T	T	NT/NT/T/NT
0	96	S3	NT,T	NT/T/NT/T → T	T	NT/T/NT/T
0	96	S1	T,T	T/NT/NT/NT → NT	T	T/NT/NT/T
1	95	S3	T,T	NT/T/NT/T → T	T	NT/T/NT/T
1	95	S1	T,T	T/NT/NT/T → T	NT	T/NT/NT/NT
0	95	S2	T,NT	NT/NT/T/NT → T	T	NT/NT/T/NT
0	94	S3	NT,T	NT/T/NT/T → T	T	NT/T/NT/T

Ejercicio 2

Apartado a

Se obtiene un 27.6% de mejora.

Apartado b

Se obtiene un 16.7% de mejora.

Apartado c

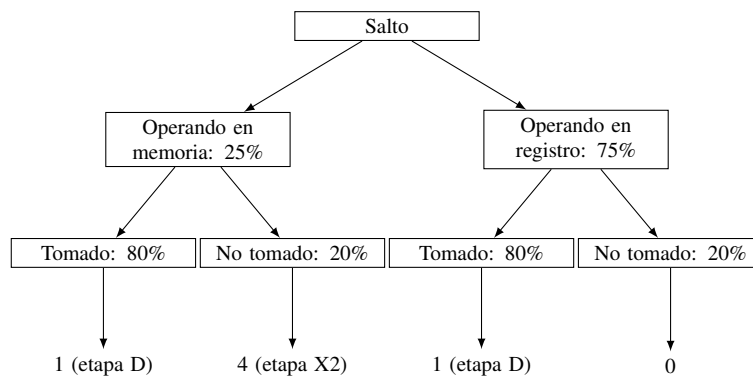
Se obtiene un 32.6% de mejora respecto al procesador original (3.9% respecto a la primera mejora y 13.6% respecto a la segunda).

Ejercicio 3

Apartado b

Suponemos que los saltos que acceden a memoria calculan la dirección de destino en la etapa D, al igual que los saltos que no acceden a memoria.

Realizamos un análisis por casos del resultado de la predicción estática:



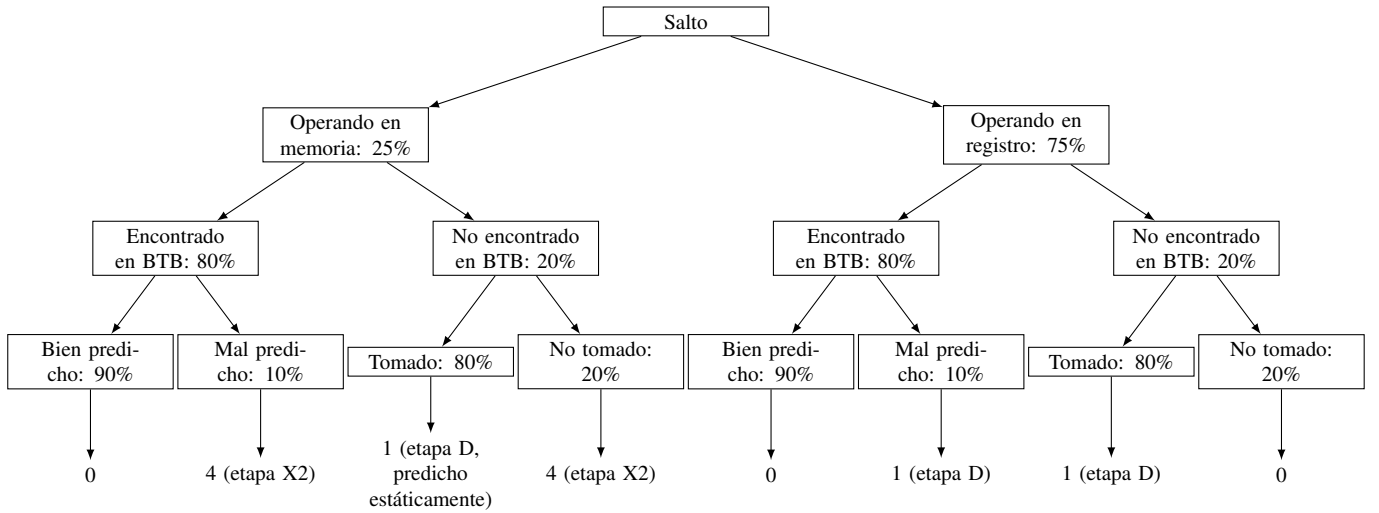
Por tanto:

$$pen_{branches} = 0.25 \times (0.80 \times 1 + 0.20 \times 4) + 0.75 \times 0.80 \times 1 = 1$$

$$CPI_{branches} = CPI_{base} + pen_{branches} = 1 + 1 = 2$$

Apartado c

Suponemos que en caso de no encontrar el salto en BTB se realiza predicción estática como en el apartado b. El análisis por casos es el siguiente:

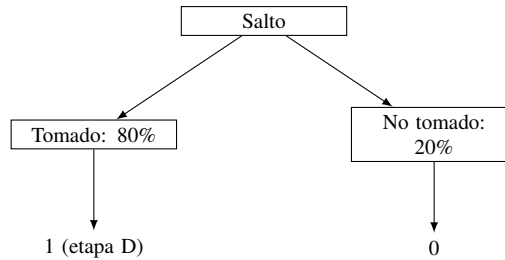


Por tanto:

$$\begin{aligned}
 pen_{branches} &= 0.25 \times ((0.80 \times 0.10 \times 4) + 0.20 \times (0.80 \times 1 + 0.20 \times 4)) \\
 &\quad + 0.75 \times (0.80 \times 0.10 \times 1 + 0.20 \times 0.80 \times 1) = 0.340 \\
 CPI_{branches} &= CPI_{base} + pen_{branches} = 1 + 0.340 = 1.340
 \end{aligned}$$

Apartado d

El análisis de casos queda reducido a:



Por lo que:

$$\begin{aligned}
 pen_{branches} &= 0.80 \times 1 = 0.80 \\
 CPI_{branches} &= CPI_{base} + pen_{branches} = 1 + 0.80 = 1.80
 \end{aligned}$$

Apartado e

Para poder comparar el rendimiento es necesario calcular el tiempo de ejecución para cada uno de los procesadores, teniendo en cuenta que el número de instrucciones del programa y su distribución cambia en algunos casos.

Sea NI_a el número de instrucciones del programa inicial (usado por los procesadores de los apartados b y d), sea $frac_{branches-a}$ la fracción en el programa original de instrucciones de salto condicional sobre el total de instrucciones, sea $frac_{branches a memoria-a}$ la fracción en el programa original de instrucciones de salto condicional que acceden a memoria sobre el total de instrucciones de salto condicional, y sea f la frecuencia de todos los procesadores.

- Procesador del apartado b:

$$\begin{aligned}
 CPI &= CPI_{base} + frac_{branches} \times pen_{branches} = 1 + 0.30 \times 1 \\
 T &= NI \times CPI \times 1/f = 1.30 \times NI_a / f
 \end{aligned}$$

- Procesador del apartado c:

$$CPI = CPI_{base} + frac_{branches} \times pen_{branches} = 1 + 0.30 \times 0.340 = 1.102$$

$$T = NI \times CPI \times 1/f = 1.102 \times NI_a / f$$

- Procesador del apartado d:

Con este pipeline, a la hora de comparar el rendimiento hay que tener en cuenta que cada instrucción de salto condicional con acceso a memoria (25% de los saltos condicionales, que a su vez son el 30% de las instrucciones totales) del programa escrito para el procesador original se transformará en dos instrucciones: un acceso a memoria para leer el operando a comparar y un salto condicional normal. Además, entre estas dos instrucciones existirá un riesgo de datos que provocará una detención de dos ciclos (de la etapa M del acceso a memoria a la etapa D del salto, $pen_{loads\ extra}$). Por tanto:

$$NI = NI_a \times (1 + frac_{branches-a} \times frac_{branches\ a\ memoria-a})$$

$$= NI_a \times (1 + 0.30 \times 0.25) = 1.075 \times NI_a$$

$$frac_{branches} = \frac{frac_{branches-a} \times NI_a}{NI}$$

$$= \frac{0.30 \times NI_a}{1.075 \times NI_a} = 0.2790697674418605$$

$$frac_{loads\ extra} = \frac{frac_{branches-a} \times frac_{branches\ a\ memoria-a} \times NI_a}{NI}$$

$$= \frac{0.30 \times 0.25 \times NI_a}{1.075 \times NI_a} = 0.06976744186046512$$

$$pen_{loads\ extra} = 2$$

$$CPI = CPI_{base} + frac_{branches} \times pen_{branches} + frac_{loads\ extra} \times pen_{loads\ extra}$$

$$= 1 + 0.2790697674418605 \times 0.80 + 0.06976744186046512 \times 2 = 1.362790697674419$$

$$T = NI \times CPI \times 1/f = 1.12 \times NI_a \times 1.01125 \times 1/f =$$

$$= 1.075 \times NI_a \times 1.362790697674419 \times 1/f = 1.465000000 \times NI_a / f$$

Por tanto, el procesador del apartado c es el más rápido.

Ejercicio 4

Apartado a

La temporización de la secuencia de instrucciones (primera iteración + primera instrucción de la segunda) es:

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
LD F0,0(R2)	IF	ID	EX	M	WB														
LD F2,8(R2)		IF	ID	EX	M	WB													
MULTD F0,F0,F2			ID	ID	ID	ID	M1	M2	M3	WB									
ADDI R2,R2,#8			IF	IF	IF	IF	ID	EX	M	WB									
ADDD F4,F0,F4							IF	ID	ID	ID	A1	A2	WB						
SUB R4,R3,R2								IF	IF	IF	ID	EX	M	WB					
BNEZ R4,Loop											IF	ID	ID	ID	EX	M	WB		
LD F0,0(R2)															IF	ID	EX	M	WB

De IF a IF de la primera instrucción de cada iteración se tarda 14 ciclos. El bucle se ejecuta 10 veces. Además la instrucción de salto tarda en terminar 3 ciclos más a partir del final de la iteración que hemos contado. Por tanto el tiempo total es $14 \times 10 + 3 = 143$.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
LD F0,0(R2)	IF	ID	EX	M	WB														
LD F2,8(R2)		IF	ID	EX	M	WB													
MULTD F0,F0,F2			ID	ID	ID	M1	M2	M3	WB										
ADDI R2,R2,#8				ID	ID	ID	EX	M	WB										
ADDD F4,F0,F4						ID	ID	ID	A1	A2	WB								
SUB R4,R3,R2							ID	ID	ID	EX	M	WB							
BNEZ R4,Loop									ID	ID	ID	EX	M	WB					
LD F0,0(R2)												IF	ID	EX	M	WB			

Apartado b

La temporización de la secuencia de instrucciones (primera iteración + primera instrucción de la segunda) es:

De IF a IF de la primera instrucción de cada iteración se tarda 11 ciclos. El bucle se ejecuta 10 veces. Además la instrucción de salto tarda en terminar 3 ciclos más a partir del final de la iteración que hemos contado. Por tanto el tiempo total es $11 \times 10 + 3 = 113$.

Apartado c

La mejora será el tiempo de ejecución del apartado a) dividido por el b). El tiempo de ejecución es $NI \times CPI \times T_{c_c}$. Por tanto:

$$SpeedUp = \frac{Teja}{Tejb} = \frac{NI \times CPI_a \times T_{c_a}}{NI \times CPI_b \times 1.1 T_{c_a}} = \frac{143}{113 \times 1.1} = 1.1504$$

La mejora de b) respecto a a) es de un 15.04%.

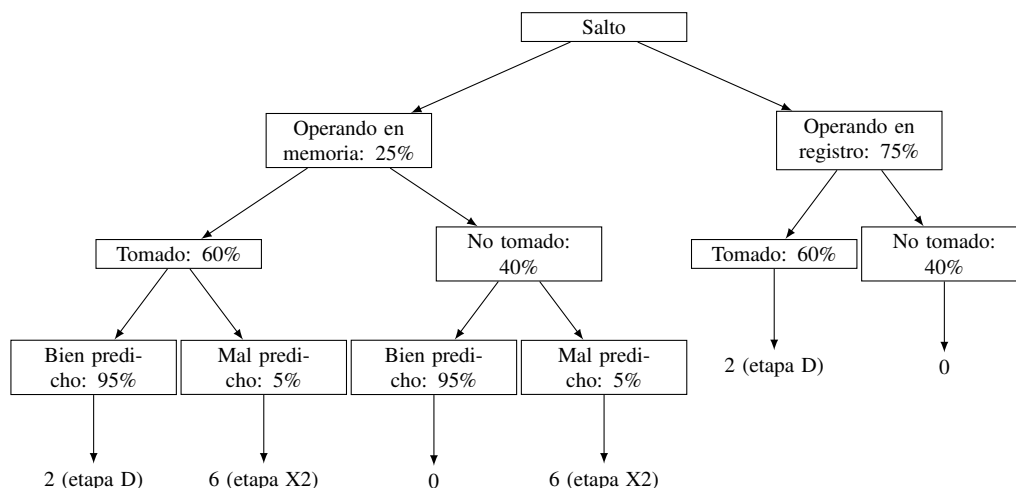
Ejercicio 5

Apartado a

El CPI, teniendo en cuenta sólo las detenciones debidas a los saltos será:

$$CPI = CPI_{base} + frac_{branches} \times pen_{branches}$$

donde $CPI_{base} = 1$ y $frac_{branches} = 0.20$ (fracción de instrucciones que son saltos condicionales). Para calcular $pen_{branches}$ (penalización de los saltos condicionales), podemos hacer un análisis por casos de los ciclos de penalización:



Con lo que:

$$pen_{branches} = 0.25 \times (0.60 \times (0.95 \times 2 + 0.05 \times 6) + 0.40 \times 0.05 \times 6) + 0.75 \times 0.60 \times 2 = 1.26$$

$$CPI = 1 + 0.20 \times 1.26 = 1.252$$

Apartado b

Cada instrucción de salto condicional con acceso a memoria (25% de los saltos condicionales, que a su vez son el 20% de las instrucciones totales) del programa escrito para el procesador original se transformará en dos instrucciones: un acceso a memoria para leer el operando a comparar y un salto condicional normal. Además, entre estas dos instrucciones existirá un riesgo de datos que provocará una detención de dos ciclos (de la etapa M del acceso a memoria a la etapa D del salto).

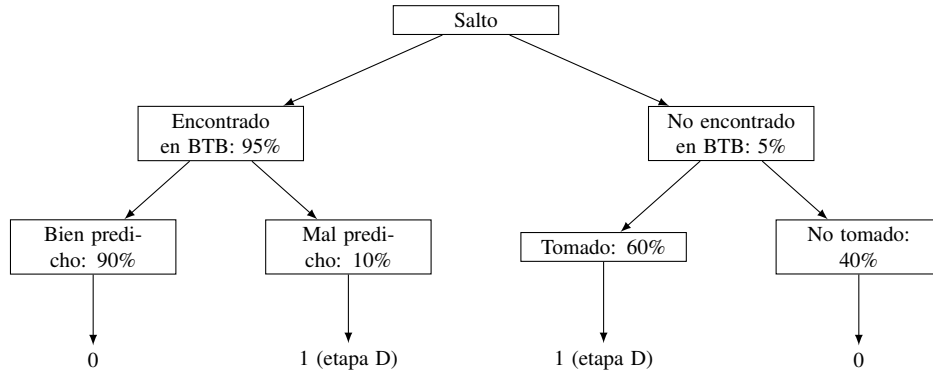
Por tanto, el número total de instrucciones y su distribución en tipos cambian: habrá una instrucción extra por cada salto que accediera a memoria en el programa original.

$$\begin{aligned}
 NI_{segundo} &= NI_{primero} \times (1 + 0.20 \times 0.25) = 1.05 \times NI_{primero} \\
 frac_{branches-segundo} &= \frac{frac_{branches-primero} \times NI_{primero}}{NI_{segundo}} = 0.1904762 \\
 frac_{loads\ extra} &= \frac{frac_{branches-primero} \times frac_{branches\ a\ memoria} \times NI_{primero}}{NI_{segundo}} \\
 &= \frac{0.20 \times 0.25}{1.05} = 0.0476190
 \end{aligned}$$

El CPI deberá tener en cuenta las instrucciones de acceso a memoria añadidas ($frac_{loads\ extra}$) y su penalización ($pen_{loads\ extra} = 2$):

$$CPI = CPI_{base} + frac_{branches} \times pen_{branches} + frac_{loads\ extra} \times pen_{loads\ extra}$$

Para calcular la nueva penalización por salto, realizamos un nuevo análisis por casos teniendo en cuenta el BTB:



Por tanto:

$$\begin{aligned}
 pen_{branches} &= 0.95 \times 0.10 \times 1 + 0.05 \times 0.60 \times 1 = 0.125 \\
 CPI &= 1 + 0.1904762 \times 0.125 + 0.0476190 \times 2 = 1.119
 \end{aligned}$$

Apartado c

$$\begin{aligned}
 f_{primero} &= 1 \text{ GHz} \\
 T_{primero} &= NI_{primero} \times CPI_{primero} \times 1/f_{primero} \\
 T_{segundo} &= NI_{segundo} \times CPI_{segundo} \times 1/f_{segundo} \\
 T_{primero} = T_{segundo} &\Rightarrow f_{segundo} = \frac{NI_{segundo} \times CPI_{segundo}}{NI_{primero} \times CPI_{primero} \times 1/f_{primero}} \\
 &= \frac{NI_{segundo}/NI_{primero} \times CPI_{segundo} \times f_{primero}}{CPI_{primero}} \\
 &= \frac{1.05 \times 1.119 \times 1 \text{ GHz}}{1.252} \\
 &= 938.5 \text{ MHz}
 \end{aligned}$$

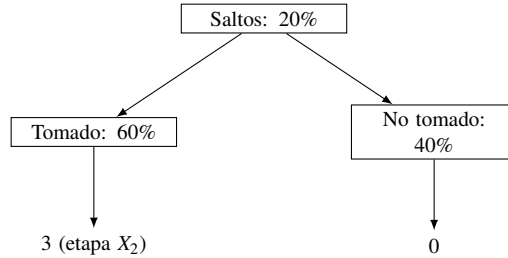
Nota: En la realización de este ejercicio se pueden cometer un error muy fácilmente al no considerar el riesgo de datos entre las instrucciones de carga y las instrucciones de salto condicional cuando se eliminan las instrucciones de salto condicional con acceso a memoria. En un examen, este error daría lugar una pérdida parcial de la puntuación del ejercicio.

Ejercicio 6

Apartado a

En primer lugar calcularemos el CPI de la configuraciones i) e ii).

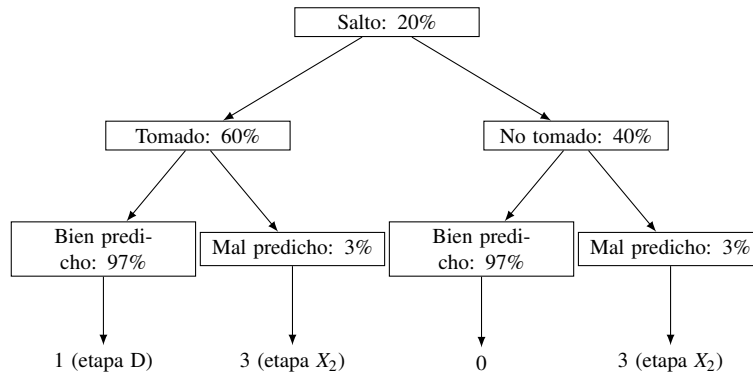
El análisis de casos para la configuración i) es relativamente sencillo, teniendo en cuenta que el salto se resuelve en la etapa X_2 , por tanto, en caso de fallo de predicción tendremos 3 ciclos de penalización:



Por lo que:

$$CPI_i = CPI_{base} + pen_{branches} = 1 + 0.2 \times 0.6 \times 3 = 1.36$$

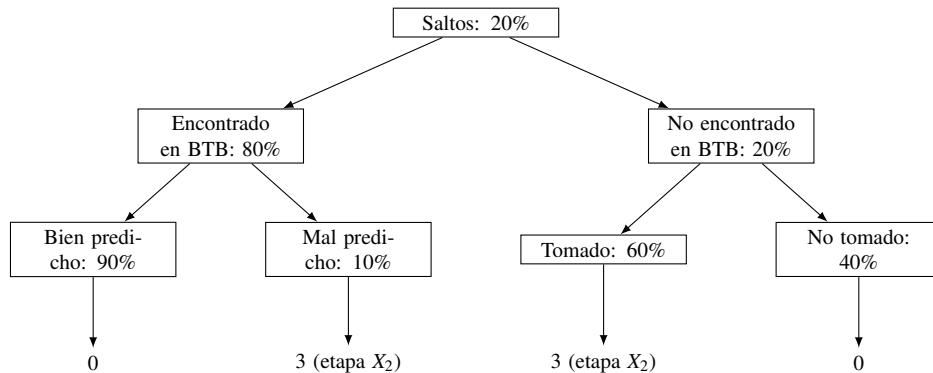
Para la configuración ii) tenemos:



$$CPI_{ii} = 1 + 0.2 \times 0.6 \times 0.97 \times 1 + 0.2 \times 0.03 \times 3 = 1.1344$$

Al tener el mismo número de instrucciones podemos compararlas solo con el CPI. La segunda opción obtiene un 19.8% de mejora en el tiempo de ejecución, respecto a la primera opción.

Apartado b



$$CPI_{iii} = 1 + 0.2 \times (0.8 \times 0.1 \times 3 + 0.2 \times 0.6 \times 3) = 1.12$$

Para obtener la mejora de iii) respecto a ii) comparamos sus CPIs:

$$SpeedUp = \frac{CPI_{iii}}{CPI_{ii}} = \frac{1.1344}{1.12} = 1.013$$

La tercera opción obtiene un 1.3% de mejora respecto a la segunda opción.

Apartado c

En el caso i con una etapa menos, la penalización por fallo de predicción de salto al reducir una etapa del cauce pasa de 3 a 2 ciclos.

$$CPI_{nuevo-i} = 1 + 0.2 \times 0.6 \times 2 = 1.24$$

El número de instrucciones (NI) no cambia pero el tiempo de ciclo ahora es:

$$T_{c_{nuevo-i}} = 1.3 \times T_{c_{ii}}$$

Por tanto la mejora de ii respecto a esta nueva configuración es:

$$SpeedUp = \frac{T_{nuevo-i}}{T_{ii}} = \frac{NI \times CPI_{nuevo-i} \times T_{c_{nuevo-i}}}{NI \times CPI_{ii} \times T_{c_{ii}}} = \frac{1.24 \times 1.3}{1.1344} = 1.421$$

La antigua opción ii obtiene un 42.1% de mejora en el tiempo de ejecución, respecto al nuevo diseño.

Ejercicio 7

Suponiendo que el BTB tiene un sólo puerto que se usa tanto para consultarlo como para actualizarlo:

$$CPI = 1.5755 \text{ ciclos}$$

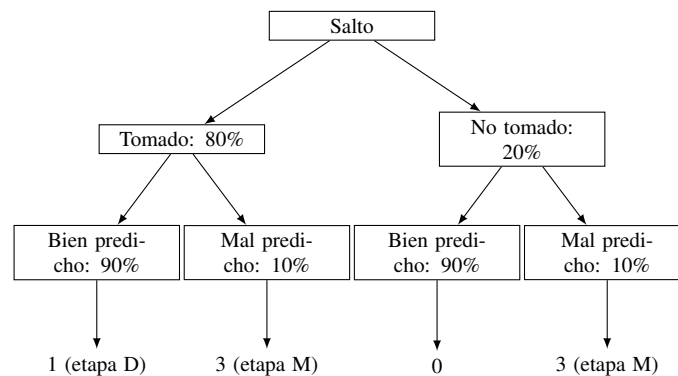
$$Tiempo = 0.437 \text{ ns}$$

Ejercicio 8

Apartado a

$$CPI = 1.75 \text{ ciclos}$$

Apartado b



$$pen_{saltos} = 0.25 \times (0.80 \times 0.9 \times 1 + 0.8 \times 0.1 \times 3 + 0.20 \times 0.10 \times 3) = 0.255$$

$$pen_{datos} = 0.2 \times 0.15 \times 1 = 0.03$$

$$CPI = CPI_{base} + pen_{saltos} + pen_{datos} = 1 + 0.255 + 0.03 = 1.285 \text{ ciclos}$$

Apartado c

$$CPI = 1.19 \text{ ciclos}$$

Apartado d

$$CPI = 1.601 \text{ ciclos}$$

Ejercicio 9

a)

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
Loop: LW R1,0(R2)	IF	ID	EX	MEM	WB																					
ADDI R1,R1,#1		IF			ID	EX	MEM	WB																		
SW 0(R2),R1					IF			ID	EX	MEM	WB															
ADDI R2,R2,#4								IF	ID	EX	MEM	WB														
SUB R4,R3,R2									IF			ID	EX	MEM	WB											
BNEZ R4,Loop												IF			ID	EX	MEM	WB								
Loop: LW R1,0(R2)																		IF	ID	EX	MEM	WB				
ADDI R1,R1,#1																			IF			ID	EX	MEM	WB	

...

17 ciclos \times 98 iteraciones + 18 ciclos de última iteración = 1684 ciclos

b)

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
Loop: LW R1,0(R2)	IF	ID	EX	MEM	WB																					
ADDI R1,R1,#1		IF		ID	EX	MEM	WB																			
SW 0(R2),R1				IF	ID	EX	MEM	WB																		
ADDI R2,R2,#4					IF	ID	EX	MEM	WB																	
SUB R4,R3,R2						IF	ID	EX	MEM	WB																
BNEZ R4,Loop							IF	ID	EX	MEM	WB															
instr x								IF	ID	EX	xxx															
instr y									IF	ID	xxx															
instr z										IF	xxx															
Loop: LW R1,0(R2)										IF	ID	EX	MEM	WB												
ADDI R1,R1,#1											IF		ID	EX	MEM	WB										

...

10 ciclos \times 98 iteraciones + 11 ciclos de última iteración = 991 ciclos

e) Usando como unidad de tiempo el tiempo de ciclo de la opción a (TC_a), los tiempos de ejecución de las diferentes opciones son:

$$T_a = 1684 \times TC_a$$

$$T_b = 991 \times 1.9 \times TC_a = 1882.9 \times TC_a$$

Por tanto, la opción b acaba con un mayor tiempo de ejecución que la opción a , en concreto un 11.8% más largo.

Ejercicio 10

- a) 6.145 ns por instrucción.
- b) 1.471 ciclos.
- c) 239.3 MHz.

Ejercicio 11

a)

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
Loop: LD F0,0(R2)	IF	ID	EX	MEM	WB																					
LD F4,0(R3)		IF	ID	EX	MEM	WB																				
MULTD F0,F0,F4			IF			ID	M1	M2	M3	M4	M5	M6	M7	MEM	WB											
ADDD F2,F0,F2						IF								ID	A1	A2	A3	A4	MEM	WB						
ADDI R2,R2,#8														IF	ID	EX	MEM	WB								
ADDI R3,R3,#8															IF	ID	EX	MEM	WB							
SUB R5,R4,R2																IF	ID	EX	MEM	WB						
BNEZ R5,Loop																	IF	ID	EX	MEM	WB					
Loop: LD F0,0(R2)																										
LD F4,0(R3)																										...

24 ciclos × 98 iteraciones + 25 ciclos de última iteración = 2377 ciclos

b)

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
Loop: LD F0,0(R2)	IF	ID	EX	MEM	WB																					
LD F4,0(R3)		IF	ID	EX	MEM	WB																				
MULTD F0,F0,F4			IF		ID	M1	M2	M3	M4	M5	M6	M7	MEM	WB												
ADDD F2,F0,F2					IF							ID	A1	A2	A3	A4	MEM	WB								
ADDI R2,R2,#8												IF	ID	EX	MEM	WB										
ADDI R3,R3,#8													IF	ID	EX	MEM	WB									
SUB R5,R4,R2														IF	ID	EX	MEM	WB								
BNEZ R5,Loop															IF	ID	EX	MEM	WB							
instr x																	IF	ID	EX	xxx						
instr y																		IF	ID	xxx						
instr z																			IF	xxx						
Loop: LD F0,0(R2)																				IF	ID	EX	MEM	WB		
LD F4,0(R3)																					IF	ID	EX	MEM	WB	

19 ciclos × 98 iteraciones + 20 ciclos de última iteración = 1882 ciclos

c) Usando como unidad de tiempo el tiempo de ciclo de la opción *a* (TC_a), los tiempos de ejecución de las diferentes opciones son:

$$T_a = 2377 \times TC_a$$
$$T_b = 1882 \times 1.5 \times TC_a = 2823.0 \times TC_a$$

Por tanto, la opción *b* empeora la opción *a*, en aproximadamente un 18.7%