

Ejercicio 11

- 25% branches { 80% salta
20% no salta
- 5% j
- 10% jal
- 60% otras (ALU | lw | sw)

Nuevo CPU → saltos incondicionales en ID y branches en EX

a) Con este diseño se eliminan los riesgos de control?

No, porque si falla la predicción entran instrucciones incorrectas.

b) Calcula el rendimiento de la nueva arquitectura MIPS respecto al uso en clase, en el que los saltos terminaban en M, suponiendo que ambas usan predicción de salto no realizado.

CPU new :	beq	IF	ID	EX	M	WB	
salta		IF	-	IF	ID	EX	M WB 3 ciclos
no salta		IF	ID	EX	M	WB	1 ciclo
1/jal		IF	ID	EX	M	WB	
			IF	ID	EX	M WB	2 ciclos

$$CPI_{\text{new}} = 0'6 \cdot 1 + 0'15 \cdot 2 + 0'25 (0'8 \cdot 3 + 0'2 \cdot 1) ;$$

$$CPI_{\text{new}} = 1'55 //$$

$$CPI_{\text{mips}} = 0'6 \cdot 1 + 0'15 \cdot 4 + 0'25 (0'8 \cdot 4 + 0'2 \cdot 1) ;$$

$$CPI_{\text{mips}} = 2'05 //$$

$$\text{Aceleración : } S = \frac{2'05}{1'55} = 1'32 //$$

c) Realiza el mismo procedimiento para predicción de salto tomado. Asumir una BTB en etapa IF.

CPU new :	beq	IF	ID	EX	M	WB	
salta		IF	ID	EX	M	WB	1 ciclo
no salta			IF	ID	EX	M WB	3 ciclos
1/jal		IF	ID	EX	M	WB	
			IF	ID	EX	M WB	1 ciclo

$$CPI_{\text{new}} : 0'6 \cdot 1 + 0'15 \cdot 1 + 0'25 (0'8 \cdot 1 + 0'2 \cdot 1) = 1'1 //$$

$$CPI_{\text{mips}} : 0'6 \cdot 1 + 0'15 \cdot 1 + 0'25 (0'8 \cdot 1 + 0'2 \cdot 4) = 1'15 //$$

$$\text{Aceleración : } S = \frac{1'15}{1'1} = 1'05 //$$

- ¿ Que estrategia de predicción es la más efectiva? ¿ Por qué? Salto realizado, porque es con la que se consigue menor CPI y por tanto menor tiempo de ejecución.