



**UNIVERSIDAD DE COSTA RICA  
FACULTAD DE INGENIERÍA  
ESCUELA DE CIENCIAS DE LA  
COMPUTACIÓN E INFORMÁTICA**

**CI0120 – ARQUITECTURA DE COMPUTADORAS  
Prof. Ileana Alpízar**

**TAREA CORTA #3**

**Elaborado por:**

**Rodrigo Vílchez Ulloa B78292**  
**[rvilchez99@gmail.com](mailto:rvilchez99@gmail.com)**

**16 de septiembre del 2020**

### Ejercicio 1.

Considero que la mejor estrategia es la del branch retrasado, pues como se indica en el enunciado, se supone que el compilador incluye 2 instrucciones buenas. En la mayoría de los casos, un branch tarda 3 ciclos en saber si la comparación de la condición se cumple o no, por lo que la solución más eficiente sería que el procesador realice estas dos instrucciones, pues el compilador las elige de manera que no afecte el desarrollo del branch. Esto es mejor que la predicción no tomada estática, pues en esta solución es posible que el procesador tenga que matar dos instrucciones si el branch indica que debe saltar a otra instrucción. También es mejor que la solución de predicción dinámica, ya que se habla que el procesador toma 2 instrucciones buenas, en la vida real, la predicción dinámica es la forma más eficiente pero para este caso específico sería más costosa que el branch retrasado.

### Ejercicio 2.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
<u>flw</u> f6, 40(x9)	<u>IF</u>	ID	EX	<u>M</u>	<u>Wb</u>																		
<u>fadd</u> f4, f6, f7		<u>IF</u>	ID	<u>A1</u>	<u>A1</u>	A2	A3	A4	<u>M</u>	<u>Wb</u>													
<u>add</u> x3, x4, x5			<u>IF</u>	ID	EX	<u>M</u>	<u>Wb</u>																
<u>sw</u> x3, 80(x3)				<u>IF</u>	ID	EX	<u>M</u>	<u>Wb</u>															
<u>fmul</u> f8, f6, f4					<u>IF</u>	ID	<u>M1</u>	<u>M1</u>	<u>M1</u>	<u>M2</u>	M3	M4	M5	M6	M7	<u>M</u>	<u>Wb</u>						
<u>fsw</u> f4, 100(x0)						<u>IF</u>	ID	EX	<u>M</u>	<u>M</u>	<u>Wb</u>												
<u>fsub</u> f1, f8, f4							<u>IF</u>	ID	<u>A1</u>	<u>A1</u>	<u>A1</u>	<u>A1</u>	<u>A1</u>	<u>A1</u>	<u>A1</u>	<u>A1</u>	A2	A3	A4	<u>M</u>	<u>Wb</u>		

CPI: 17/7 --> aprox 2,43

### Ejercicio 3.

ETIQ:

```
addi    x2, x3, 2
add    x1, x1, x2
addi   x4, x4, 1
addi    x3, x3, -1
bnez    x3, ETIQ
```

ETIQ:

```
addi    x2, x3, 2
addi    x3, x3, -1
bnez    x3, ETIQ
add    x1, x1, x2
addi   x4, x4, 1
```

Código:

```
for(i = 10; i > 0; i--) {
    b = i+2
    a = a+b
    c++
}
```

```
x1 = variable a
x2 = variable b
x3 = contador i
x4 = variable c
```