

En el problema asumimos que todas las instrucciones de transferencia son de carga.

- a) Para que los dos tipos de máquinas tengan el mismo rendimiento debemos igualar las siguientes expresiones:

$$Re\ n\ dim\ iento_{nuevo} = Re\ n\ dim\ iento_{viejo}$$

El rendimiento de los dos tipos de CPU se expresa como:

$$Re\ n\ dim\ iento_{viejo} = Re\ cuento_{viejo} \times CPI \times Ciclo_{viejo}$$

$$Re\ n\ dim\ iento_{nuevo} = Re\ cuento_{nuevo} \times CPI \times Ciclo_{nuevo}$$

y teniendo en cuenta las duraciones de los ciclos de reloj:

$$Ciclo_{nuevo} = Ciclo_{viejo} \times 1.1$$

Se deduce:

$$Re\ n\ dim\ iento_{viejo} = Re\ cuento_{viejo} \times CPI \times Ciclo_{viejo}$$

$$Re\ n\ dim\ iento_{nuevo} = Re\ cuento_{nuevo} \times CPI \times Ciclo_{viejo} \times 1.1$$

Igualando estas dos expresiones tendremos:

$$1.1 \times Re\ cuento_{nuevo} = Re\ cuento_{viejo}$$

$$Re\ cuento_{nuevo} = 0.91 \times Re\ cuento_{viejo}$$

Por lo tanto, para que ambas máquinas tengan el mismo rendimiento es necesario que el recuento nuevo sea un 9% menor que el recuento de la máquina original. De acuerdo con esto, habrá que eliminar ese 9% del 36% de instrucciones de carga que existen en el repertorio de instrucciones antiguo (instrucciones de transferencia), y que equivale a eliminar un **25%** de ellas:

$$\frac{0.09 \times 100}{0.36} = 25\%$$

- b) Ya que la nueva instrucción elimina la utilización del registro en donde se almacena el resultado de la carga, bastará con escribir una secuencia de instrucciones en la cual dicho registro sea utilizado de nuevo por alguna otra instrucción. Por ejemplo:

```
LD    R1, 0(Rb)
ADD   R2, R2, R1
.....
ADD   R3, R3, R1
```

Si sustituimos las dos primeras instrucciones por la nueva, anulamos el registro R1 impidiendo la correcta ejecución de la última instrucción. En el código que queda al sustituir las dos primeras instrucciones se aprecia esto claramente:

```
ADD  R2, 0(Rb)
.....
ADD  R3, R3, R1
```