

Ejercicio 1.1  
Ejercicio 1.2  
Ejercicio 1.8  
Ejercicio 1.17  
Ejercicio 1.18  
Ejercicio 1.19

## Ejercicio 1.1

El computador C1 con: Procesador de 500 MHz de frecuencia de reloj. Cache de datos de 32 KB, mapeo directo, bloque de 32B, 60 ns de penalización en caso de fallo. Cache de instrucciones ideal.

Ejecuta el programa:

```
int i;  
float s;  
float x[10000];  
for(i = 0; i < 10000; i++)  
    s = s + x[i];
```

que el compilador traduce en 5 instrucciones de lenguaje máquina. El programa tarda 300 microsegundos en ejecutarse.

Pregunta 1: Calcule: MFLOPS (millones de operaciones en coma flotante por segundo), MIPS, CPI.

$$MFLOPS = \frac{N_{ops} FLOAT}{\Delta T} * 10^{-6} = \frac{1 * 10^4}{300 * 10^{-6}} * 10^{-6} = 33.3 MFLOPS$$

$$MIPS = \frac{N_{ops}}{\Delta T} * 10^{-6} = \frac{5 * 10^{-4}}{300 * 10^{-6}} = 166.6 MIPS$$

$$CPI = \frac{ciclos}{N} = \frac{\frac{T_{exec}}{T_c}}{N} = 3$$

$$T_{exec} = N * CPI * T_c \rightarrow CPI = \frac{T_{exec}}{N * T_c}$$

Pregunta 2: Calcule la fracción (m) de accesos a datos que son fallo de cache y  $T_{mem}$ .

$$m = \frac{fallos}{accesos} = \frac{2,5 * 10^3}{10^4} = 0.25$$

$$T_{mem} = N * m * t_{penlaty} = 10^4 * 0.25 * 60ns = 1,5 * 10^{-4} s$$

El tiempo de memoria será, el numero de accesos a memoria (1 en cada iteración), por la tasa de fallos por acceso por la penalización de un fallo.

Mejoras tecnológicas permiten implementar el procesador y las caches con un tiempo de ciclo de reloj de 1.5 ns. Ahora bien, la organización del procesador y del subsistema de memoria es idéntica a la del computador C1.

Pregunta 3: Calcule el tiempo de ejecución  $T'_{ex}$  del programa en el nuevo computador y la ganancia.

$$T'_{ex} = T'_{CPU} + T_{MEM}$$

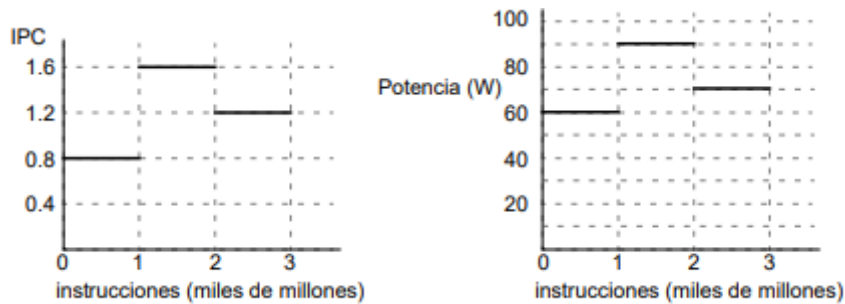
$$T'_{CPU} = N * CPI_{CPU} * T_c * \frac{T'_c}{T_c} = T_{CPU} * T'_c$$

$$T'_{ex} = T_{CPU} * \frac{T'_c}{T_c} + T_{MEM} = (150\mu s) * \frac{1.5ns}{2ns} + 150\mu s = 112.5\mu s + 150\mu s = 262.5\mu s$$

$$G = \frac{T_{ori}}{T_{new}} = \frac{300\mu s}{262.5\mu s} = 1.14$$

El tiempo de memoria no se ve alterado, solamente afecta al tiempo de CPU; por tanto, debemos calcular el nuevo tiempo de CPU, donde el nuevo tiempo de ciclo se debe aplicar.

Ganancia Usando ley de Amdahl:



$$F_m := \text{Fracción de tiempo original donde se usa la mejora} = \frac{T_2}{T_0} = \frac{150\mu s}{300\mu s} = 0.5$$

$$G_m := \text{Ganancia de la mejora.} = \frac{T_2}{T_3} = \frac{150\mu s}{150\mu s} * \frac{1.5ns}{2ns} = 1 + 1/3$$

$$G_{amdahl} = \frac{1}{(1 - F_m) + \frac{F_m}{G_m}} = \frac{1}{0.5 + 0.38} = 1.14$$

En este caso, T2 es el Tcpu y T1 es Tmem. Comprovamos que las G concuerdan.

## Ejercicio 1.2

En un procesador las instrucciones Load y Store representan un 26% de las instrucciones siendo 2.3 el CPI. Todas las demás instrucciones tienen 1.5 de CPI.

Pregunta 1: Calcule el CPI del procesador original.

$$CPI^0 = \sum \frac{N_i * CPI_i}{N} = \sum C_i * CPI_i = 0.26 * 2.3 + 0.74 * 1.5 = 1.71$$

En la próxima generación del procesador, debido a un cambio en el proceso tecnológico, se puede disponer de más transistores y se plantean dos opciones en la forma de utilizar estos transistores adicionales.

1. Incrementar el tamaño de la cache de primer nivel, lo cual reduce en un 20% el CPI de las instrucciones Load y Store, porque no se incrementa la latencia y la tasa de fallos es menor.
2. Utilizar los transistores en mejorar todos los componentes del procesador en el chip, lo cual permite reducir el periodo del reloj en un 10%.

Pregunta 2: Calcule el CPI de la 1ª opción de diseño.

$$CPI^1 = 0.26 * 2.3 * 0.8 + 0.74 * 1.5 = 1.58$$

Pregunta 3: Calcule el tiempo de ejecución en la 2ª opción de diseño.

$$T_{exec}^2 = T_{exec} * 0.9$$

Pregunta 4: ¿Cuál de las dos opciones permite obtener un procesador con mayor rendimiento?

$$\frac{T_1}{T_2} = \frac{N * CPI_1 * T_c}{N * CPI_0 * T_c * 0.9} = \frac{CPI_1}{CPI_0 * 0.9} = \frac{1.58}{1.71 * 0.9} = > 1$$

**Pregunta 5:** ¿Cuál es la ganancia, respecto del procesador original, de la mejor opción?

$$G = \left( \frac{T_{ori}}{T_{new}} - 1 \right) * 100 = \frac{N * CPI_0 * T_c}{N * CPI_0 * T_c * 0.9} = \frac{1}{0.9} = 1.11$$

La ganancia es del 11%.

## Ejercicio 1.8

Un procesador interpreta instrucciones de 3 tipos: enteras (ENT), accesos a memoria (MEM) y saltos (BR). La frecuencia de reloj del procesador es de 800 MHz.

El primer nivel de la jerarquía de memoria está integrado en el mismo chip del procesador y consta de una cache de instrucciones y una cache de datos (CD). Para reducir la latencia media de accesos a memoria, el procesador dispone de una cache (segundo nivel) externa. En este ejercicio consideraremos que la cache de instrucciones y la cache de segundo nivel son ideales (no hay fallos) y que las instrucciones MEM son bloqueantes.

La penalización por fallo en la cache de datos es de 12 ciclos. El procesador ejecuta un programa P. La siguiente tabla muestra la distribución de instrucciones y el CPI:

- ENT -> 40% CPI = 1
- MEM hit en CD -> 20% CPI = 2
- MEM miss en CD -> 20% CPI = 14
- BR -> 20% CPI = 2

**Pregunta 1:** Calcule el IPC medio, CPI, la cantidad de MIPS, fallos en CD por instrucción, fallos en CD por acceso a memoria al ejecutar el programa P.

$$CPI = 0.4 * 1 + 0.2 * 2 + 0.2 * 14 + 0.2 * 2 = 4$$

$$IPC = CPI^{-1} = 0.25$$

$$MIPS = \frac{N}{\Delta T} = \frac{N}{N * CPI * T_c} = IPC * \nu = 0.25 * 800 MHz = 200 MIPS$$

$$F_i = 0.2$$

$$F_a = \frac{fallos}{Acesos} = \frac{0.2}{1+0.4} = 0.14$$

Mejoras tecnológicas permiten integrar la cache de segundo nivel en el chip (reduciendo la latencia de acceso) y aumentar la frecuencia de reloj a 1 GHz. En el nuevo diseño se mantiene la microarquitectura original. La potencia consumida por el chip es de 50 w (vatios). La batería que alimenta el chip tiene una capacidad energética de 25 wh (vatios x hora). Al ejecutar el programa P en el nuevo diseño se obtiene un CPI medio igual a 3.

**Pregunta 2:** Calcule el número máximo de instrucciones de P para que el procesador pueda ejecutar completamente el programa sin recargar la batería.

$$E = P * \Delta T = J * S; P(J/s) = \frac{E}{\Delta T} = \frac{1J}{1s}$$

$$E_b = 25W * Hora * 3600s = 90 * 10^3 J$$

$$E_c = P * T = 50W * (1GHz)^{-1} = 50nJ$$

Calculamos la energía de un ciclo, usando el tiempo de ciclo

$$Ciclos = \frac{E_b}{E_c} = 1,8 * 10^9 ciclos \rightarrow Instr = \frac{ciclos}{CPI} = 6 * 10^{11} instrucciones$$

Dividimos la energía de la batería entre la de un ciclo, obteniendo los ciclos y entonces con el CPI podremos saber el número de instrucciones.

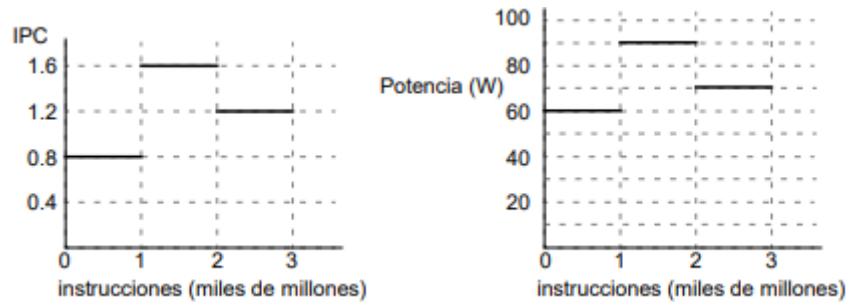
Pregunta 3: Calcule el CPI de las instrucciones MEM que fallan en CD y deduzca la penalización (en ciclos) por fallo.

$$CPI = 3 = \sum CPI_i * F_i = 1.2 + 0.2 * x \rightarrow x = 9 \text{ ciclos}$$

Como se tardan 2 ciclos en caso de acierto, **la penalización es de 7 ciclos.**

## Ejercicio 1.17

En las siguientes figuras se muestra el IPC y la potencia consumida a medida que se interpretan instrucciones de un programa, en un procesador que funciona a una frecuencia de 2 Ghz.



Pregunta 1: Calcule el tiempo de ejecución del programa.

$$T_{exe} = \left( \sum \frac{1}{IPC_i} * N_i \right) * T_c = \left( \frac{1}{0.8} * 1 * 10^9 + \frac{1}{1.2} * 1 * 10^9 + \frac{1}{1.6} * 1 * 10^9 \right) * (2GHz)^{-1} = 4.06s$$

Pregunta 2: Calcule la potencia de conmutación media. Recuerde que la potencia es la energía por unidad de tiempo.

$$P = \frac{E}{T} = \frac{\sum P_i * T_i}{T_{exe}}$$

$$P_1 * T_1 = 60W * ((0.8^{-1} * 10^9) * 2GHz^{-1}) = 0.625 * 60W = 37.5J = E_1$$

$$P_2 * T_2 = 80W * (1.6^{-1} * 10^9 * 2GHz^{-1}) = 5/16 * 80W = 25J = E_2$$

$$P_3 * T_3 = 70W * (1.2^{-1} * 10^9) * 2GHz^{-1}) = 0.42 * 70W = 29.17J = E_3$$

$$P = 22.5W$$

Para calcular la potencia, necesitaremos la energía por tramos, usando el  $T_i$  que se calcula siguiendo la fórmula de la pregunta 1.

Pregunta 3: Calcule el tiempo de ejecución cuando se utiliza escalado tensión-frecuencia para mantener la potencia consumida dentro del límite de 80 W en cada región.

Pregunta 4: Calcule la potencia de conmutación media en el escenario descrito.

## Ejercicio 1.18

Un procesador convencional tiene un primer nivel de cache de datos de 32 Kbytes 2-asociativa, siendo el tamaño de bloque de 16 bytes. La cache de datos es bloqueante y la latencia de acceso a memoria son 4 ciclos. Por otro lado, supondremos que la cache de instrucciones es ideal; es decir, no se producen fallos de cache.

En este procesador se efectúa el cálculo especificado en el siguiente código:

## Ejercicio 1.19