

[Ejercicio 1.1](#)
[Ejercicio 1.2](#)
[Ejercicio 1.8](#)
[Ejercicio 1.17](#)
[Ejercicio 1.18](#)
[Ejercicio 1.19](#)
[Ejercicio 1.3](#)
[Ejercicio 1.4](#)
[Ejercicio 1.10](#)
[Ejercicio 1.14](#)

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} .

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

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

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.

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

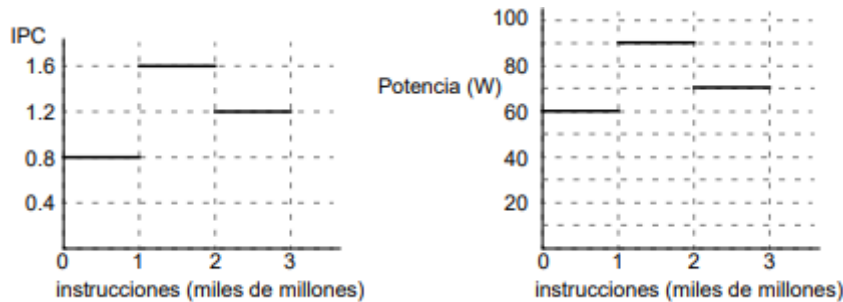
$$T'_{CPU} = N * CPI_{CPU} * T_c * \frac{T'_c}{T_c} = T_{CPU} * T'_c$$

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

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

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?

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

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.

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

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

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

$$F_i = 0.2$$

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

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.

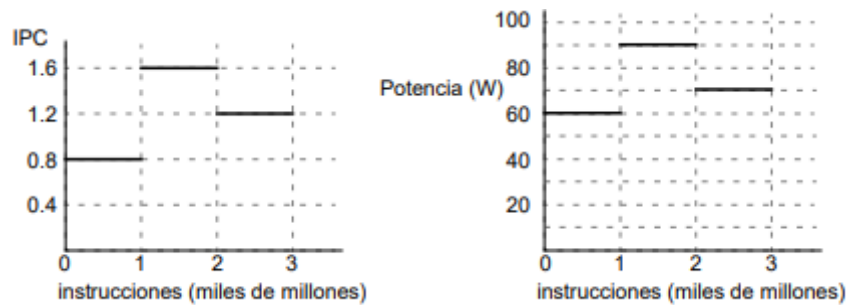
Pregunta 3: Calcule el CPI de las instrucciones MEM que fallan en CD y deduzca la penalización (en ciclos) por fallo.

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

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.

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

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

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

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

$$P = 22.5W$$

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

Ejercicio 1.3

Deduzca 1) la cantidad de MIPS, 2) el CPI medio y 3) el tiempo total para cada una de las tres preguntas siguientes.

$$MIPS = \frac{Instrucciones * 10^{-6}}{Tiempo} = \frac{Iteraciones Instrucciones}{ciclos \frac{Tiempo}{ciclos}}$$
$$CPI_m = \frac{Ciclos}{Instr}$$
$$T = N * CPI * T_c = \frac{Instrucciones}{Programa} * \frac{Ciclos}{Instruccion} * \frac{Tiempo}{1Ciclo} = \frac{Tiempo}{Programa} (s)$$

Pregunta 1: Procesador con reloj de 900 MHz que ejecuta 1000 iteraciones de un bucle de 12 instrucciones y tarda 18 ciclos en cada iteración.

$$MIPS = \frac{1000 \text{ iteraciones} * 12 \text{ instrucciones} * 10^{-6}}{18 \text{ ciclos} * (900MHz)^{-1}} = 600.000 \text{ MIPS}$$
$$CPI_m = 18 \text{ Ciclos} / 12 \text{ Instrucciones} = 1.5 \text{ Ciclos} / Instr$$
$$T = (1000 * 12) * 1.5 * (900MHz)^{-1} = 2 * 10^{-5} s$$

Pregunta 2: Procesador con reloj de 450 MHz que ejecuta 2000 iteraciones de un bucle de 12 instrucciones y tarda 18 ciclos en cada iteración.

$$MIPS = \frac{2000 * 12 * 10^{-6}}{18 * (450MHz)^{-1}} = 600.000 \text{ MIPS}$$
$$CPI = \frac{18 \text{ Ciclos}}{12 \text{ Instrucciones}} = 1.5$$
$$T = (2000 * 12) * CPI * (450MHz)^{-1} = 8 * 10^{-5} s$$

Pregunta 3: Procesador con reloj de 900 MHz que ejecuta 2000 iteraciones de un bucle de 12 instrucciones y tarda 36 ciclos en cada iteración.

$$MIPS = \frac{2000 * 12 * 10^{-6}}{36 * (900MHz)^{-1}} = 600.000 \text{ MIPS}$$
$$CPI = \frac{36}{12} = 3$$
$$T = 2000 * 12 * CPI * (900MHz)^{-1} = 8 * 10^{-5} s$$

Sabemos que un procesador emplea la mitad de su tiempo en hacer accesos a cache. Queremos reducir el tiempo total del procesador en un 20% del original.

Pregunta 4: Deduzca cuanto más rápida debería ser la nueva cache para conseguirlo, expresando la respuesta como el valor del cociente: tiempo cache vieja / tiempo cache nueva

La ley de Amdahl nos ayudara, buscaremos la ganancia para $G = \frac{T}{T*0.2} = 5$. Deberemos calcular F_m y G_m , la fracción de tiempo que se utiliza el component a mejorar y la ganancia cuando se utiliza al 100%. G_m es justo el cociente que pide el enunciado.

$$G_m = T_{cache} / T'_{cache}$$

$$F_m = 0.5T / T = 0.5$$

$$G = \frac{1}{(1 - F_m) + F_m / G_m} = \frac{1}{0.5 + 0.5 / G_m} = 2 + \frac{G_n}{0.5}$$

$$G_m = (G - 2) * 0.5 = 1.5$$

El resultado es que debiera ser 1.5 veces más rápida la nueva cache.

Ejercicio 1.4

Un programa de prueba P contiene 20 millones de operaciones en coma flotante.

Una estación de trabajo E tiene un procesador a 90 MHz, incluye un coprocesador numérico y usa un compilador con optimizaciones. Este compilador permite que E efectúe los cálculos coma flotante bien en modo 1 (habilitando el coprocesador), o bien en modo 2 (inhabilitando el coprocesador y usando rutinas software). La ejecución de P tarda 1.5 s cuando E funciona en modo 1, mientras que tarda 13.6 s en modo 2.

Se ha medido que CPI = 5 en modo 1, mientras que CPI = 3 en modo 2.

Pregunta 1: Deduzca la velocidad en MIPS para cada uno de los dos modos.

$$MIPS = \frac{Instrucciones * 10^{-6}}{Tiempo} = IPC * Freq * 10^{-6}$$

$$MIPS = \frac{instr}{ciclos} * \frac{1ciclo}{Tiempo} = instr / tiempo$$

$$MIPS^1 = (5)^{-1} * 90MHz = 18 MIPS$$

$$MIPS^2 = (3)^{-1} * 90MHz = 30 MIPS$$

Pregunta 2: Deduzca el número total de instrucciones ejecutadas en cada uno de los dos modos.

Tenemos el tiempo de ciclo, CPI y el tiempo total, por tanto:

$$T = N * CPI * T_C \rightarrow N = T / (CPI * T_C)$$

$$N_1 = 1.5s / (5 * (90MHz)^{-1}) = 2.7 * 10^7 \text{ instrucciones}$$

$$N_2 = 13.6s / (3 * (90MHz)^{-1}) = 4.08 * 10^8 \text{ instrucciones}$$

Pregunta 3: En el modo 2, deduzca el promedio de instrucciones necesarias para ejecutar una operación coma flotante.

Como el programa contiene 20 millones de operaciones y conocemos el numero de instrucciones que se ejecutan en el modo 2:

$$408 \text{ millones de instr} / 20 * \text{ millones de ops} = 20.4$$

Obtemos que se necesitan 20.4 instrucciones de promedio.

Pregunta 4: En el modo 1, deduzca la velocidad en MFLOPS.

El enunciado nos da el tiempo que tarda en ejecutarse y el numero de operaciones:

$$MFLOPS = \frac{FloatsOp * 10^{-6}}{Tiempo} = \frac{20}{1.5} = 13.3 MFLOPS$$

Se puede modificar el modo 1, cambiando el coprocesador por otro que es capaz de acelerar F veces la ejecución de las operaciones responsables del 30% del tiempo total de ejecución de P.

Pregunta 5: Deduzca el valor que debe tener F para que el tiempo de ejecución se reduzca a la mitad. Deduzca cuál sería el nuevo tiempo de ejecución si sabemos que F = 2.

Aplicaremos la ley de Amdahl, tenemos que el tiempo de mejora es 'F' y se aplica en el 30% de las instrucciones:

$F_m :=$ fracción de uso respecto el total = 0.3

$$G_m := T_{Componente}^0 / T'_{Componente} = F$$

$$G = \frac{1}{(1-F_m) + \frac{F_m}{G_m}} = \frac{1}{(1-0.3) + \frac{0.3}{F}}$$

Para reducir el tiempo a la mitad, necesitamos $G = \frac{T_0}{T_0 * 0.5} = 2$, sustituimos y buscamos F:

...

Con F=2 obtenemos que la G = 1.18, entonces el tiempo de ejecución será:

$$G = \frac{T_0}{T'} = 1.18 \rightarrow T' = 1.5 / 1.18 = 1.27s$$

Ejercicio 1.10

Al interpretar un programa se mide la siguiente distribución de instrucciones y CPI asociado.

TIPO	DIST	CPI
STORE	15%	1
LOAD	25%	2
SECUENCIAMIENTO	15%	4
ARITMÉTICAS, ENTEROS	35%	1
DESPLAZAMIENTO ARITMÉTICO	5%	1
MULTIPLICACIÓN	5%	10

Pregunta 1: Calcule el CPI.

$$CPI = 0.15 * 1 + 0.25 * 2 + 0.15 * 4 + 0.35 * 1 + 0.05 * 1 + 0.05 * 10 = 2.15$$

Una optimización efectuada por el compilador reemplaza operaciones complejas o difíciles por operaciones más simples, con el objetivo de reducir el tiempo de ejecución. Un ejemplo es reemplazar una multiplicación por una constante por una secuencia de instrucciones de desplazamiento aritmético y de suma. En el programa anterior se ha medido que el 50% de las multiplicaciones se pueden convertir en secuencias de instrucciones de desplazamiento aritmético y de suma, siendo el número medio de instrucciones de la secuencia igual a 3.

Pregunta 2: Calcule el incremento relativo en el número de instrucciones.

El 50% de las operaciones de multiplicación (suponían el 5%) tendrán un CPI = 3. Entonces el nuevo número de instrucciones N' será el anterior número de instrucciones mas 3 instrucciones por cada operación de multiplicación que podemos transformar:

$$N' = N + (N * 0.05 * 0.5 * 3) = N + (N * 0.075)$$

$$N' = 1.075 * N$$

Pregunta 3: Calcule el CPI cuando se interpreta el código optimizado. Previamente calcule la distribución de instrucciones.

Para la nueva distribución, podemos suponer cualquier combinación de desplazamiento aritmético y suma, pues el CPI de las dos es 1, añadimos esta optimización como un tipo 'operación' y calculamos el CPI:

multiplicación -> dist = $0.05 * 0.5 = 0.025 = 2.5\%$, CPI = 10

multiplicaciónOptimizada -> dist = $0.05 * 0.5 = 2.5\%$, CPI = 1

$$CPI' = 0.15 * 1 + 0.25 * 2 + 0.15 * 4 + 0.35 * 1 + 0.05 * 1 + 0.025 * 10 + 0.025 * 1 = 1.925$$

Pregunta 4: Calcule la ganancia que aporta la optimización de código. Efectúe el cálculo utilizando la ley de Amdahl. Previamente calcule la fracción de tiempo en la cual se utiliza la mejora.

$$G = \frac{1}{(1-F_m) + \frac{F_m}{G_m}}$$

$F_m :=$ Fracción de uso original = 0.025

$G_m :=$ Ganancia en el 100%

$$G_m = \frac{1N * 10Ciclos * T_c}{3N * 1Ciclos * T_c} = 10/3$$

$$G = \frac{1}{(1-0.025) + \frac{0.025}{10/3}} = 1.02$$

Comprobando con la otra formula de la ganancia:

$$G = T_0/T_N = \frac{N * CPI_0}{1.075 * N * CPI'} = 1.03$$

Ejercicio 1.14

Un procesador con una frecuencia de reloj f MHz ejecuta una carga de trabajo de n programas. Sea N el número total de instrucciones ejecutadas y CIC el número total de ciclos de ejecución del conjunto de programas. De cada programa se conoce:

$p_i :=$ fracción de tiempo (CIC_i/CIC)

donde CIC_i es el número de ciclos de ejecución

$CPI_i :=$ CPI medio, $MIPS_i =$ media de MIPS

$W_i :=$ potencia media W consumida por P

$$T = \sum f_i * T_i$$

Pregunta 1: Deduzca una expresión para evaluar el CPI medio de la carga de trabajo en función del CPI de cada programa.

$$CPI_{carga} = \sum CPI_i * p_i$$

Pregunta 2: Deduzca una expresión para evaluar el rendimiento en MIPS de la carga de trabajo en función de los MIPS de cada programa.

$$MIPS_{carga} = \sum MIPS_i * p_i$$

Pregunta 3: Calcule la potencia media W consumida por el procesador (vatios) al ejecutar la carga de trabajo en función de la potencia en cada programa.

$$W_{carga} = \sum W_i * p_i$$

Pregunta 4: Calcule la energía (julios) por millón de instrucciones (EMPI) en función del rendimiento en MIPS.

$$EMPI = \frac{Energia}{10^6 Inst}, Potencia_{total} = \sum W_i * p_i, E = P_t * TiempoTotal$$

$$(MIPS \frac{Instrucciones}{Segundo} * 10^6)^{-1} * Potencia_{total}$$