

Arquitectura de Computadoras

2026 - 1

6 de febrero de 2026

Expediente	Nombre
223210350	Amaya Soria Angel Alberto
223216745	Flores Salazar Luis Angel
223215039	Miranda Sanchez Javier Leonardo
223203899	Tostado Cortes Dante Alejandro

Tarea 1

Nota: En los ejercicios siguientes no escribir solo el resultado, se califica procedimiento y resultado.

1. Convierta los siguientes números a las bases indicadas.

a) Convertir el entero -72 de base 10 a binario con 8 bits.

Conversion del valor absoluto 72:

Division	Cociente	Residuo
$\frac{72}{2}$	36	0
$\frac{36}{2}$	18	0
$\frac{18}{2}$	9	0
$\frac{9}{2}$	4	1
$\frac{4}{2}$	2	0
$\frac{2}{2}$	1	0
$\frac{1}{2}$	0	1

$$72_{10} = 01001000_2$$

Intercambiamos ceros por unos y viceversa.

Numero Invertido: 10110111

Sumamos 1 al valor invertido: 10111000

$$-72_{10} = 10111000_2$$

b) Convertir el entero 11001110 de base 2 con 8 bits a base 10.

$$x = (1 \cdot 2^7) + (1 \cdot 2^6) + (0 \cdot 2^5) + (0 \cdot 2^4) + (1 \cdot 2^3) + (1 \cdot 2^2) + (1 \cdot 2^1) + (0 \cdot 2^0)$$

$$x = (1 \cdot 128) + (1 \cdot 64) + (0 \cdot 32) + (0 \cdot 16) + (1 \cdot 8) + (1 \cdot 4) + (1 \cdot 2) + (0 \cdot 1)$$

$$x = 128 + 64 + 8 + 4 + 2 = 206$$

c) Convertir el número 1010010.101 de base 2 a base 16.

Agregar los ceros necesarios:

01010010.1010

Dividir en bloques:

0101 – 0010. – 1010

Convertir cada bloque de 4 dígitos a hexadecimal:

52.A₁₆

d) Convertir el número 48.D de base 16 a base 2.

4 = 0100

8 = 1000

D = 1101

Juntamos todo:

01001000.1101₂

e) Mostrar cómo se representa el número de punto flotante -121.47 (base 10) en el estándar IEEE-754 con precisión sencilla.

El número es negativo, por lo tanto:

S = 1

Conversión del valor absoluto 121.47₁₀ a binario:

Parte entera: 121₁₀ = 1111001₂

Parte fraccionaria:

0.47 = $\frac{47}{100}$

Multiplicaciones sucesivas por 2:

Paso	Resultado	Bit
$0.47 \times 2 = 0.94$	0.94	0
$0.94 \times 2 = 1.88$	0.88	1
$0.88 \times 2 = 1.76$	0.76	1
$0.76 \times 2 = 1.52$	0.52	1
$0.52 \times 2 = 1.04$	0.04	1
$0.04 \times 2 = 0.08$	0.08	0
$0.08 \times 2 = 0.16$	0.16	0
$0.16 \times 2 = 0.32$	0.32	0
$0.32 \times 2 = 0.64$	0.64	0
$0.64 \times 2 = 1.28$	0.28	1
$0.28 \times 2 = 0.56$	0.56	0
$0.56 \times 2 = 1.12$	0.12	1
$0.12 \times 2 = 0.24$	0.24	0
$0.24 \times 2 = 0.48$	0.48	0
$0.48 \times 2 = 0.96$	0.96	0
$0.96 \times 2 = 1.92$	0.92	1
$0.92 \times 2 = 1.84$	0.84	1

Parte fraccionaria en binario:

0.47₁₀ = 0.01111000010100011₂

Número completo en binario:

121.47₁₀ = 1111001.01111000010100011₂

Normalización:

$$1.11100101111000010100011_2 \times 2^6$$

Cálculo del exponente:

$$E = 6 + 127 = 133_{10} = 10000101_2$$

Representación IEEE-754 en precisión sencilla:

$$1 \quad 10000101 \quad 11100101111000010100011$$

2. Considerar dos implementaciones del mismo ISA. Las instrucciones se dividen en 4 clases: A, B, C y D. P1 tiene un reloj de 2 GHz y CPIs de 1, 2, 3 y 3. P2 tiene un reloj de 3 GHz y CPIs de 2, 2, 4 y 4. Suponer que un programa hace siete mil millones de instrucciones con la siguiente mezcla: 10 % de A, 20 % de B, 50 % de C y 20 % de D. Responder las siguientes preguntas:

a) ¿Cuál implementación es más rápida?

Instrucciones por clase:

$$A = 10 \% = 7 \times 10^9 \cdot 0.1 = 7 \times 10^8$$

$$B = 20 \% = 7 \times 10^9 \cdot 0.2 = 14 \times 10^8$$

$$C = 50 \% = 7 \times 10^9 \cdot 0.5 = 35 \times 10^8$$

$$D = 20 \% = 7 \times 10^9 \cdot 0.2 = 14 \times 10^8$$

P1 :

$$CPIs \begin{array}{|c|c|c|c|} \hline A = 1 & B = 2 & C = 3 & D = 3 \\ \hline \end{array}$$

$$CiclosTotalesP1 = (1 \cdot 7 \times 10^8) + (2 \cdot 14 \times 10^8) + (3 \cdot 35 \times 10^8) + (3 \cdot 14 \times 10^8) = 18.2 \times 10^9$$

$$Tiempo1 = \frac{18.2 \times 10^9}{2 \times 10^9} = 9.1 \text{ Segundos}$$

P2 :

$$CPIs \begin{array}{|c|c|c|c|} \hline A = 2 & B = 2 & C = 4 & D = 4 \\ \hline \end{array}$$

$$CiclosTotalesP2 = (2 \cdot 7 \times 10^8) + (2 \cdot 14 \times 10^8) + (4 \cdot 35 \times 10^8) + (4 \cdot 14 \times 10^8) = 23.8 \times 10^9$$

$$Tiempo2 = \frac{23.8 \times 10^9}{3 \times 10^9} = 7.93 \text{ Segundos}$$

Respuesta: La implementación P2 es más rápida

b) ¿Cuál es el CPI del programa en cada implementación?

Instrucciones por clase, $CPIs1, CPIs2$:

$$A = 10 \%, 1, 2$$

$$B = 20 \%, 2, 2$$

$$C = 50 \%, 3, 4$$

$$D = 20 \%, 3, 4$$

$$CPI1 = (0.1 \cdot 1) + (0.2 \cdot 2) + (0.5 \cdot 3) + (0.2 \cdot 3) = 2.6$$

$$CPI2 = (0.1 \cdot 2) + (0.2 \cdot 2) + (0.5 \cdot 4) + (0.2 \cdot 4) = 3.4$$

c) ¿Cuántos ciclos hace el programa en cada implementación?

Tomamos los ciclos totales del ejercicio 2A

$$CiclosTotalesP1 = 18.2 \times 10^9$$

$$CiclosTotalesP2 = 23.8 \times 10^9$$

d) ¿Cuál es la velocidad pico de cada implementación?

$$VelocidadPico = \frac{Frecuencia}{CPI_{Minimo}}$$

$$VelocidadPico1 = \frac{2 \times 10^9}{1} = 2 \times 10^9 \text{ instrucciones/s}$$

$$VelocidadPico2 = \frac{3 \times 10^9}{2} = 1.5 \times 10^9 \text{ instrucciones/s}$$

3. Suponer que, para un programa, el compilador A genera un número de instrucciones de $1.0E9$ y un tiempo de ejecución de 1.1 s, mientras que el compilador B genera un número de instrucciones de $1.2E9$ y un tiempo de ejecución de 1.5 s.

- a) Encontrar el CPI promedio para cada programa, dado que el procesador tiene un ciclo de reloj de 1 ns.

Formula a usar:

$$T = I \cdot CPI \cdot T_c$$

$$CPI = \frac{T}{I \cdot T_c}$$

Valor de $T_c = 1ns = 1 \times 10^{-9}$

- Compilador A:

Datos:

$$I = 1 \times 10^9$$

$$T = 1.1 \text{ s}$$

Proceso:

$$CPI_A = \frac{1.1}{(1 \times 10^9) \cdot (1 \times 10^{-9})} \quad CPI_A = \frac{1.1}{1} = 1.1$$

- Compilador B:

Datos:

$$I = 1.2 \times 10^9$$

$$T = 1.5 \text{ s}$$

Proceso:

$$CPI_B = \frac{1.5}{(1.2 \times 10^9) \cdot (1 \times 10^{-9})}$$

$$CPI_B = \frac{1.5}{1.2} = 1.25$$

- b) Suponer que los programas compilados se ejecutan en dos procesadores diferentes. Si los tiempos de ejecución en ambos procesadores son iguales, ¿Cuánto más rápido es el reloj del procesador que ejecuta el código del compilador A en comparación con el reloj del procesador que ejecuta el código del compilador B?

$T_A = T_B$, pero en diferentes procesadores

Formula a usar:

$$T = I \cdot CPI \cdot T_C$$

$$I_A \cdot CPI_A \cdot T_{CA} = I_B \cdot CPI_B \cdot T_{CB}$$

$$f = \frac{1}{T_C}$$

$$\frac{T_{CA}}{T_{CB}} = \frac{I_B \cdot CPI_B}{I_A \cdot CPI_A} = \frac{(1.2 \times 10^9) \cdot (1.25)}{(1 \times 10^9) \cdot (1.1)}$$

$$\frac{T_{CA}}{T_{CB}} = \frac{1.5}{1.1} = 1.36$$

- c) Se desarrolla un nuevo compilador C que utiliza solo 6.0E8 instrucciones y tiene un CPI promedio de 1.1. ¿Cuál es el speedup al usar este nuevo compilador en comparación con usar el compilador A o el compilador B en el procesador original?

Datos: $I = 6 \times 10^8$

$CPI = 1.1$

$T_A = 1.1$ s

$T_B = 1.5$ s

Formulas a usar:

$$SpeedUP = \frac{T_A}{T_C}$$

$$T_C = I \cdot CPI \cdot T_c$$

Proceso:

$$T_C = (6 \times 10^8) \cdot (1.1) \cdot (1 \times 10^{-9})$$

$$T_C = 0.66 \text{ s}$$

$$SpeedUP_{A/C} = \frac{1.1}{0.66} = 1.67$$

$$SpeedUP_{B/C} = \frac{1.5}{0.66} = 2.27$$

4. Suponer que a una computadora se le cambia la tarjeta gráfica por otra y que un benchmark tardaba en correr 90 segundos usando la tarjeta original. Al correr el benchmark con la tarjeta nueva se detecta un speedup de 1.5. ¿Qué tan rápida es la nueva tarjeta gráfica si se sabe que el benchmark pasa el 40 % del tiempo ejecutando instrucciones gráficas?

a) Plantear claramente la ecuación a resolver.

Usar ley de Amdahl, el speedup total de un sistema está dado por:

$$S_{total} = \frac{1}{(1 - f) + \frac{f}{S_{mejorado}}}$$

donde:

- S_{total} es el speedup total del programa,
- f es la fracción del tiempo de ejecución que se ve mejorada,
- $S_{mejorado}$ es el speedup de la parte mejorada del sistema.

La parte que se mejora aquí es la de las instrucciones gráficas, por lo que:

$$S_{total} = 1.5$$

$$f = 0.40$$

$$1 - f = 0.60$$

Sustituyendo estos valores en la ecuación da:

$$1.5 = \frac{1}{0.60 + \frac{0.40}{S_{GPU}}}$$

Esa es la ecuación que representa el speedup de la nueva tarjeta gráfica.

b) Resolver la ecuación correctamente.

Despejando S_{GPU} :

$$\frac{1}{1.5} = 0.60 + \frac{0.40}{S_{GPU}}$$

$$0.6667 = 0.60 + \frac{0.40}{S_{GPU}}$$

$$0.0667 = \frac{0.40}{S_{GPU}}$$

$$S_{GPU} = \frac{0.40}{0.0667}$$

$$S_{GPU} \approx 6$$