Pregunta	**si**	**no**
1. Considere el siguiente fragmento de código para el que al lado de cada instrucción aparece su dirección física en la memoria.	☐	■

```
0x4FF10000           beq $s1, $s2, mylabel
                     ...
                     ...
                     ...
0x4FF2A000 mylabel:  addi $t1, $t2,4
```

El código anterior es correcto. ¿Es eso cierto?

$$\begin{array}{r} 4FF2A000 - \\ 4FF10000 \\ \hline 0001A000 \end{array}$$

$1A000_{16} = 106'496_{10}$. El máximo offset representable en complemento a 2 sobre 16 bits es $7FFF_{16} = 32767_{10} < 106'496_{10}$ por tanto el código anterior es incorrecto porque no es posible alcanzar "mylabel"

2. Un programa tarda 10^6 ciclos en ejecutarse en un procesador dado. Si el sistema tiene un CPI medio de 40, ¿es cierto que el programa tiene 15000 instrucciones?	☐	■

$$\frac{10^6 \text{ ciclos}}{40 \frac{\text{ciclos}}{\text{instrucción}}} = 25'000 \text{ instrucciones}$$

3. Considere el fragmento de código que se muestra a continuación	■	☐

```
li $t0, 0xABCD9876
sw $t0, 100($0)
lb $s5, 101($0)
```

La ejecución de este fragmento produce que se cargue en el registro `$s5` el valor `0x98`. ¿Es cierto que el procesador que ejecuta este código es de tipo *Little endian*?

100	101	102	103
76	98	CD	AB

Si la máquina fuera de tipo little endian por lo que el resultado de la lb es 98

4. Un procesador pasa el 40% del tiempo procesando referencias a memoria. ¿Es cierto que para realizar un speedup de 1.2 tenemos que mejorar el subsistema memoria de un factor 1.8?	■	☐

Se trata de una aplicación de la ley de Amdahl:

$$T_{improved} = \frac{T_{affected}}{\text{improvement factor}} + T_{unaffected}$$

$$\text{Speed up} = \frac{T_{old}}{T_{improved}} = \frac{T_{old}}{T_{unaffected} + \frac{T_{affected}}{\text{improvement factor}}} \Rightarrow \frac{T_{old}}{0.6\,T_{old} + \frac{0.4\,T_{old}}{1.8}} \simeq 1.2$$

5. El número **3E400000** es un número en coma flotante de precisión sencilla en formato IEEE 754. ¿Es cierto que este número representa **0.6875** en base 10?

S	EXP	MANTISSA
0	01111100	1000--- 0

$exp = 01111100_2 = 124$

el exponente denormalizado es $exp = 124 - 127 = -3$

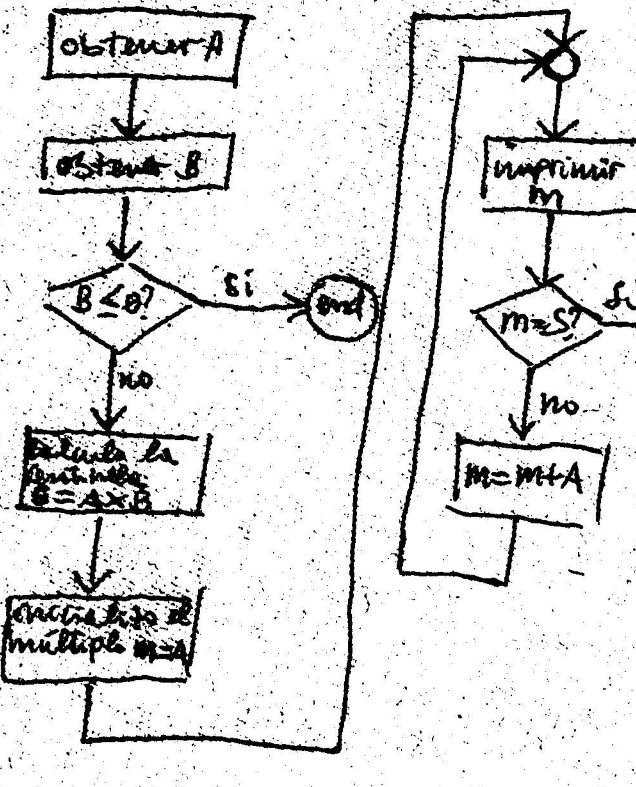
Teniendo en cuenta el hidden bit la mantisa es $1.1000\text{---}0_2 = 1{,}5_{10}$

por lo que el numero es $1.5 \cdot 2^{-3} = 0.1875_{10}$

PREGUNTA 2.1. Confeccione un programa en lenguaje ensamblador que lea dos números enteros A y B e imprima en pantalla todos los múltiplos de A, desde A hasta A x B. Utilice las *syscalls* que se muestran en la tabla siguiente:

Servicio	Valor en $v0	Argumentos	Resultado
Leer entero	5	-	$v0 contiene el entero

Se pide también representar de forma esquemática el algoritmo utilizado mediante un diagrama de flujo.




```
main:
  # Cargo A en $t0, B en $t1 (se omite)
  blez $t1, exit   # if B≤0 exit
  mul $t2, $t0, $t1   # s = A*B
  move $t3, $t0    # m = A
loop:
  # print $t3 (se omite)
  beq $t2, $t3, exit
  add $t3, $t3, $t0
  # imprimo un espacio (se omite)
  b loop
exit:
  # salgo del programa (se omite)
```

PREGUNTA 2.2. ¿Cuántos modos de direccionamiento tiene el MIPS? Descríbalos detalladamente. Asimismo ¿Cuál es la diferencia entre una instrucción de direccionamiento inmediato, y una de direccionamiento indirecto? ¿Cuántas referencias de memoria se necesitan para cada uno de estos tipos de instrucciones para traer un operando a un registro del procesador? Poner ejemplos de instrucciones del MIPS que implementan estos modos de direccionamiento.

Ver teoría en transparencias o libro de texto.

PREGUNTA 2.3 (2 puntos). Se quiere valorar las prestaciones de dos compiladores distintos compilando un mismo programa. La tabla siguiente muestra para cada uno de los compiladores utilizados el número de instrucciones en las que es compilado el programa y los tiempos de ejecución.

Compiler A		Compiler B	
Núm. instrucciones	Tiempo de ejec.	Núm. instrucciones	Tiempo de ejec.
1×10^9	1.8s	1.2×10^9	1.6s

1. Halle el CPI medio para cada uno de los programas compilados sabiendo que el procesador tiene una frecuencia de reloj de 1 GHz.
2. Asuma ahora el mismo CPI hallado en el punto anterior; sin embargo, esta vez los dos programas se ejecutan en procesadores distintos. Si los tiempos de ejecución de los dos procesadores son idénticos, ¿de cuánto es más elevada la frecuencia de reloj del procesador que ejecuta el código del compilador A respecto a la del procesador que ejecuta el código del compilador B?

1) El CPI puede calcularse como: $CPI = \dfrac{T_{exec} \cdot f_{clk}}{\text{núm. instrucciones}}$

Sustituyendo los datos del problema hallo los CPIs para el código generado por los compiladores A y B:

$$CPI_A = \frac{1.8 \cdot 1 \cdot 10^9}{1 \cdot 10^9} = 1.8 \qquad CPI_B = \frac{1.6 \cdot (1 \cdot 10^9)}{1.2 \cdot 10^9} = 1.33$$

2) La relación entre las frecuencias de reloj de los dos procesadores es

$$\frac{f_{clk_A}}{f_{clk_B}} = \frac{CPI_A \cdot \text{núm. instrucciones}_A}{\cancel{T_{exec}}} \cdot \frac{\cancel{T_{exec}}}{CPI_B \cdot \text{núm. instrucciones}_B} = \frac{CPI_A \cdot \text{núm. instruc.}_A}{CPI_B \cdot \text{núm. instruc}_B}$$

por tanto $\dfrac{f_{clk_A}}{f_{clk_B}} = \dfrac{1.8 \cdot (1 \cdot 10^9)}{1.33 \cdot (1.2 \cdot 10^9)} \simeq 1.127$ por tanto f_{clk_A} es un 12,7% más elevada que f_{clk_B}

PROBLEMA 3.1. La tabla muestra las características de dos versiones de un mismo procesador.

Versión	Tensión de alimentación	Frecuencia de reloj
1	3V	800 MHz
2	3.3V	1.5 GHz

1. De cuánto se ha reducido la carga capacitiva total entre versiones si la versión 2 disipa el 90% de la potencia disipada por la versión 1.
2. De cuánto se reduce la potencia dinámica si la carga capacitiva no cambia.
3. Asumiendo que la capacidad total de la versión 2 es el 80% de la de la versión 1, halle la nueva tensión de alimentación de manera que la potencia disipada por la versión 2 se reduzca del 30% con respecto a la potencia disipada por la versión 1.

La potencia dinámico disipada por un procesador que opera a la frecuencia f_{CLK}, con una tensión de alimentación V y una capacidad total conmutada es:

$$P = V^2 \cdot f_{CLK} \cdot C$$

por tanto las versiones 1 y 2 del procesador disipan respectivamente:

$$P_1 = 5^2 \cdot (0.8 \cdot 10^9) \cdot C_1 \qquad P_2 = 3{,}3^2 \cdot (1{,}5 \cdot 10^9) \cdot C_2$$

pero, de los datos del problema es sabido que $P_2 = 0{,}9\, P_1$, por tanto:

$$3.3^2 \cdot (1{,}5 \cdot 10^9) \cdot C_2 = 0{,}9 \left[5^2 \cdot (0.8 \cdot 10^9) \cdot C_1 \right]$$

Finalmente sigue que:

$$\frac{C_2}{C_1} = \frac{0.9 \cdot 5^2 \cdot (0.8 \cdot 10^9) \cdot}{(3.3)^2 \cdot (1.5 \cdot 10^9)} = \frac{18}{16{,}335} \simeq 1.1$$

La versión 2 debe conmutar una capacidad total un 10% superior a la de la versión 1.

Asumiendo que $C_2 = 0.8 C_1$ y teniendo en cuenta que una reducción de potencia del 30% equivale a $P_2 = 0{,}7 P_1$, se obtiene:

$$V_2^2 \cdot f_{CLK_2} \cdot (0.8 \cancel{C_1}) = 0{,}7\, V_1^2 \cdot f_{CLK1} \cdot \cancel{C_1}$$

por tanto, la tensión de alimentación de la versión 2 será:

$$V_2 = \sqrt{\frac{0.7\, V_1^2\, f_{CLK1}}{0.8\, f_{CLK2}}} = \sqrt{\frac{0{,}7 \cdot 5^2 \cdot (0.8 \cdot 10^9)}{0.8 \cdot (1.5 \cdot 10^9)}} = \sqrt{\frac{0.7 \cdot 20}{1{,}2}} = \sqrt{\frac{14}{1.2}} = 3{,}4 V$$