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?

2. La instrucción `lw $s1,100($zero)` implementa un modo de direccionamiento relativo a registro (*base addressing*). ¿Es eso cierto? ☐ ☐

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*?

4. A=**10010011** y B=**00111110** representan dos números binarios con signo representados en complemento a 2. ¿Es cierto que el |A|< |B|? ☐ ☐

5. El número **3E400000** es un número en coma flotante de precisión sencilla en formato IEEE 754. ¿Es cierto que este número representa **0.1875** en base 10? ☐ ☐

PARTE 2

PREGUNTA 2.1. Considere el siguiente fragmento de código escrito en Java

```
int[] arr = {11, 22, 33, 44};
len = arr.length;
// la traducción de estas líneas es dada

// complete la traducción de las líneas que siguen...
int evensum = 0;
for (int i=0; i<len; i++) {
  if (arr[i] & 1 == 0) {
    evensum += arr[i];
  }
}
```

Analice el código anterior y explique brevemente qué es lo que hace. A continuación, represente el algoritmo implementado por el fragmento anterior mediante un diagrama de flujo. Finalmente, complete la traducción (que se muestra a continuación) en leguaje ensamblador MIPS del fragmento anterior. Utilice el registro `$t0` para almacenar el valor de la variable `evensum`.

```
.data
  arr: .word 11 22 33 44
  len: .word 4

.text
  la   $s0, arr
  la   $s1, len
  subu $s1, $s1, $s0
  srl  $s1, $s1,2
  ....
```

Explique también por qué, después de ejecutar el código ensamblador anterior, el registro `$s1` contiene el tamaño del vector `arr`.

PREGUNTA 2.2. Asuma que un procesador dado no tiene soporte hardware para la multiplicación de manera que la única forma de realizar una multiplicación es por software mediante sumas sucesivas. Si para realizar una multiplicación en hardware se tardan 4 ciclos, mientras que por software se precisan 200 ciclos, ¿cuál sería la mejora del rendimiento (Speedup) de una implementación hardware si el procesador pasa el 10% de su tiempo ejecutando multiplicaciones? ¿Cuál sería el Speedup de la implementación hardware si el procesador pasa el 40% de su tiempo ejecutando multiplicaciones? Finalmente, explique, justificando adecuadamente sus conclusiones, en cuál de los dos escenarios anteriores es más conveniente utilizar una implementación hardware.

PREGUNTA 2.3. 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?

PARTE 3

PROBLEMA 3.1. NVIDIA tiene un "medio" formato parecido al IEEE 754 excepto por el hecho que es de 16 bits. El bit más significativo representa el signo, seguido por un exponente sobre 5 bits representado en exceso-16 y una mantisa de 10 bits con *hidden bit*. Escriba el patrón binario asumiendo del número -1.5625×10^{-1}. Compare el rango y la precisión de este formato con el de precisión sencilla del IEEE 754.

Finalmente, asuma A=2.6125×10^{1} y B=4.150390625×10^{-1}. Calcule la suma de A y B a mano asumiendo el "medio" formato NVIDIA descrito anteriormente. Asuma un bit de guarda, 1 bit de redondeo y un *sticky bit*. Redondee al número par más próximo. Ilustre todos los pasos seguidos.

Arquitectura de ordenadores (23/10/2018)

Pregunta 1

dirección beq 0X4FF10000

dirección addi 0X4FF2A000

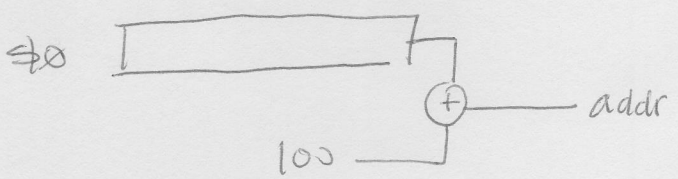
diferencia entre direcciones: 0X4FF2A000 - 0X4FF10000 = 0X1A000 ✗

el offset del salto no se puede representar sobre 16 bits en complemento a2

R: el código es incorrecto

Pregunta 2

lw $s1, 100($zero)



R: es un modo de direccionamiento relativo a registro ($zero en este caso)

Pregunta 3

Addr.	data (big endian)	data (little endian)
~~100~~	AB	76
~~101~~	CD	98
~~102~~	98	CD
~~103~~	76	AB

R: la lb en 101 devuelve 98 por lo que el procesador es de tipo little endian.

Pregunta 4

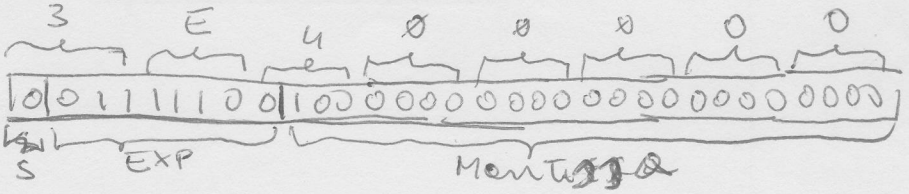
$A = 10010011 = -109_{10}$

$B = 00111110 = 72_{10}$

$|A| = 109 > |B| = 72$ R: falso

Pregunta 5

X = 3E400000 ⇒

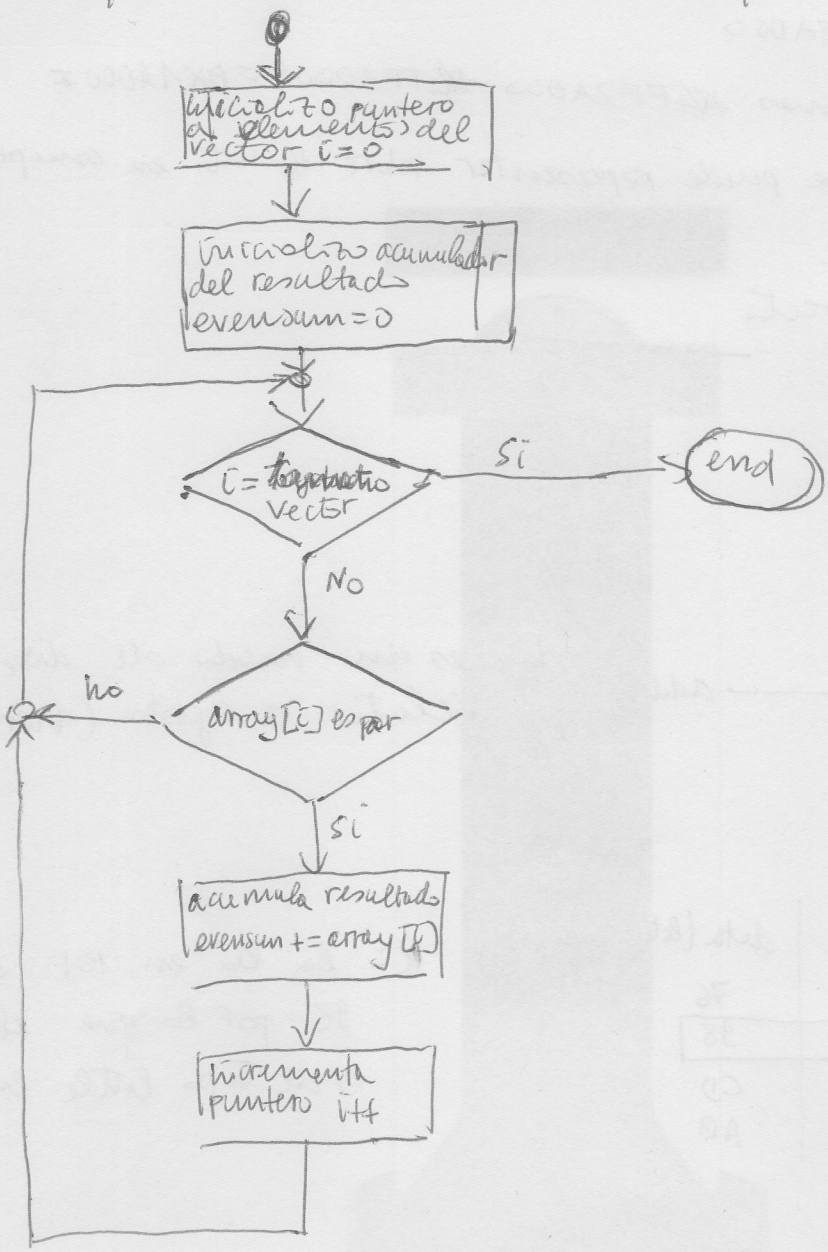


EXP = 01111100 = 124 − 127 = −3

$X = +2^{-3} \cdot 1.0 = 0.0011_2 = \frac{1}{8} + \frac{1}{16} = \frac{3}{16} = 0.1875$ R: cierto

Pregunta 2.1 (la traducción en ensamblador se deja como ejercicio)

- el programa realiza un barrido de un vector de enteros sumando aquellos elementos del vector que son números pares.



Pregunta 2.2

Se trata de aplicar la ley de Ahmdal:

$$\text{Tiempo mejorado} = (\text{Fracción mejorada}) \cdot \frac{\text{Tiempo no mejorado}}{\text{factor de mejora}} + (\text{Fracción no mejorada}) \cdot \text{Tiempo no mejorado}$$

$$\text{Tiempo mejorado} = \left[\frac{(\text{fracción mejorada})}{\text{factor de mejora}} + (\text{fracción no mejorada})\right] \text{Tiempo no mejorado}$$

$$\text{Speedup} = \frac{\text{Tiempo no mejorado}}{\text{Tiempo mejorado}} = \frac{1}{(\text{fracción no mejorada}) + \frac{\text{fracción mejorada}}{\text{factor de mejora}}}$$

(a) 10% multiplicaciones $\Rightarrow$ fracción mejorada $= 0.1$
fracción no mejorada $= 1 - \text{fracción mejorada} = 0.9$

$$\text{factor de mejora} = \frac{200}{4} = 50$$

$$\text{Speedup} = \frac{1}{0.9 + \frac{0.1}{50}} = 1.11$$

(b) 40% multiplicaciones $\Rightarrow$ fracción mejorada $= 0.4$
fracción no mejorada $= 1 - 0.4 = 0.6$

$$\text{Speedup} = \frac{1}{0.6 + \frac{0.4}{50}} = 1.64$$

es más conveniente utilizar la implementación hardware en el caso (b) ya que se alcanzaría un speedup mayor.

Pregunta 2-3

Compilador A		Compilador B	
# instr.	T. exec	# instr.	T. exec
$1 \cdot 10^9$	1.85	$1.2 \cdot 10^9$	1.6 s

$f_{clk} = 1\,GHz$

(a) $CPI = \dfrac{T_{exec} \cdot f_{clk}}{\#\,instr.} = \begin{cases} \dfrac{1.8 \cdot (1 \cdot 10^9)}{1 \cdot 10^9} = 1.8 & \text{(Compilador A)} \\ \dfrac{1.6 \cdot (1 \cdot 10^9)}{1.2 \cdot 10^9} \simeq 1.3 & \text{(Compilador B)} \end{cases}$

(b) $\dfrac{f_{clk_A}}{f_{clk_B}} = \dfrac{CPI_A \cdot \#\,instr_A}{T_{exec}} \dfrac{T_{exec}}{CPI_B \cdot \#\,instr_B} = \dfrac{CPI_A \cdot \#\,instr_A}{CPI_B \cdot \#\,instr_B} = \dfrac{1.8 \cdot (1 \cdot 10^9)}{1.3 \cdot (1.2 \cdot 10^9)} = \dfrac{1.8}{1.56} =$

$= 1.15$ el procesador A ejecuta un programa más corto que el B en el mismo tiempo procesando un f_{clk} un 15% más elevada que B

Problema 3-1

$-1.5625 \cdot 10^{-1} = -0.15625 \cdot 10^0 = -0.00101 \cdot 2^0 = -1.01 \cdot 2^{-3}$ (hidden bit)

1) $exp = -3 + 16 = 13 = 01101_2 \Rightarrow$

1	01101	0100000000
s	exp	mantissa

2) $A = 2.6125 \cdot 10^1 = 26.125 = 11010.001 = 1.1010001 \cdot 2^4$

$B = 4.150390625 \cdot 10^{-1} = 0.4150390625 = 1.1010100111 \cdot 2^{-2}$

tengo que alinear B con el exponente mayor (corriendo el número 6 posiciones a la derecha

$B = 0.000000\,1101010011\,1$

Sumo los números

$1.1010001000\,;\,GR = 00$

$0.0000011010\,;\,100111$ (bits descartados $\neq 0 \Rightarrow$ sticky bit = 1)

$1.1010001000\,;\,GRS = 000$

$0.0000011010\,;\,101$

$1.1010100010\,;\,101$ $\quad \frac{5}{8} \cdot 2^{-3} = 5 \cdot 2^{-3} \cdot 2^{-10} = \frac{5}{8} 2^{-10} = \frac{5}{8} LSB > \frac{1}{2} LSB$

por lo que tengo que redondear el resultado

$A+B = 1.1010100011 \cdot 2^4 = 11010.100011 \cdot 2^0 = 26.546875 =$

$= 2.654\,6875 \cdot 10^1$