

Tercera prueba voluntaria de Fundamentos de Computadores. Grupo 2. 16/05/2013

APELLIDOS Y NOMBRE:

DNI:

1. Sean dos registros de un computador, llamémosles A y B que contienen los siguientes patrones de bits:

A 0x704a0bf2

B 0x9ff74af0

- i) (0,5 puntos) Suponiendo que los registros A y B se interpretan como enteros de 32 bits en complemento a 2, calcule la suma $A + B$ de los números representados en ambos registros tal y como la efectuaría un procesador. Expresar el resultado en hexadecimal.

Explicación:

Suma $A + B$: 0x 104156e2

$$\begin{array}{r} A \quad 704a0bf2 \\ B \quad 9ff74af0 \\ \hline 1|104156e2 \end{array}$$

- ii) (0,5 puntos) Suponiendo que los registros A y B se interpretan como enteros de 32 bits en complemento a 2, calcule la diferencia $A - B$ de los números representados en ambos registros tal y como la efectuaría un procesador. Expresar el resultado en hexadecimal.

Explicación:

Diferencia $A - B$: 0x d052c102

$$\begin{array}{r} A \quad 704a0bf2 \\ -B \quad 6008b510 \\ \hline d052c102 \end{array}$$

- iii) (0,5 puntos) ¿Habrá desbordamiento en las operaciones anteriores? (rodee con un círculo la respuesta correcta)

Explicación:

¿Habrá desbordamiento en $A + B$?	SI	<input checked="" type="radio"/> NO
¿Habrá desbordamiento en $A - B$?	<input checked="" type="radio"/> SI	NO

En $A + B$ no puede haber desbordamiento porque se están sumando números de diferente signo.

En $A - B$ hay desbordamiento porque al sumar $A + (-B)$ ambos operandos son positivos y el resultado es negativo

Tercera prueba voluntaria de Fundamentos de Computadores. Grupo 2. 16/05/2013

APELLIDOS Y NOMBRE:

DNI:

1. Sean dos registros de un computador, llamémosles A y B que contienen los siguientes patrones de bits:

A **0x704a0bf2**

B **0x9ff74af0**

- i) (0,5 puntos) Suponiendo que los registros A y B se interpretan como enteros de 32 bits en complemento a 2, calcule la suma $A + B$ de los números representados en ambos registros tal y como la efectuaría un procesador. Expresar el resultado en hexadecimal.

Explicación:

Suma $A + B$: 0x **104156e2**

$$\begin{array}{r} A \quad 704a0bf2 \\ B \quad 9ff74af0 \\ \hline 1 \quad 104156e2 \end{array}$$

- ii) (0,5 puntos) Suponiendo que los registros A y B se interpretan como enteros de 32 bits en complemento a 2, calcule la diferencia $A - B$ de los números representados en ambos registros tal y como la efectuaría un procesador. Expresar el resultado en hexadecimal.

Explicación:

Diferencia $A - B$: 0x **d052c102**

$$\begin{array}{r} A \quad 704a0bf2 \\ -B \quad 6008b510 \\ \hline d052c102 \end{array}$$

- iii) (0,5 puntos) ¿Habrá desbordamiento en las operaciones anteriores? (rodee con un círculo la respuesta correcta)

Explicación:

¿Habrá desbordamiento en $A + B$?	SI	<input checked="" type="radio"/> NO
¿Habrá desbordamiento en $A - B$?	<input checked="" type="radio"/> SI	NO

En $A + B$ no puede haber desbordamiento porque se están sumando números de diferente signo.

En $A - B$ hay desbordamiento porque al sumar $A + (-B)$ ambos operandos son positivos y el resultado es negativo

2. (0,5 puntos) Sea el patrón de bits 0xc1158000. Suponiendo que ese patrón de bits está codificado mediante la norma IEEE-754 para la representación de números en punto flotante con simple precisión: ¿Qué número decimal representa?

Cálculos:

Número representado:

0xc1158000

- 9,34375

$$\begin{array}{c}
 \text{S} \quad \text{E} \quad \text{F} \\
 1 \mid 10000010 \mid 001010110000000000000000
 \end{array}$$

$$x = (-1)^S \times 1, F \times 2^{E - \text{Exceso}} = -1,00101011 \times 2^{130-127} = -1,00101011 \times 2^3 = -1001,01011$$

Parte entera: $2^3 + 2^0 = 8 + 1 = 9$

Parte fraccionaria: $2^{-2} + 2^{-4} + 2^{-5} = 0,25 + 0,0625 + 0,03125 = 0,34375$

Por tanto el número representado es: -9,34375

3. (1 punto) Escriba en hexadecimal la representación del número -24,6875 utilizando la norma IEEE-754 de simple precisión.

Cálculos:

Representación de -24,6875: 0x C1C58000

Pasaremos el número a binario

Parte entera: $24 \begin{array}{l} \div 2 \\ 0 \\ \div 2 \\ 0 \\ \div 2 \\ 0 \\ \div 2 \\ 1 \end{array} \Rightarrow 24 = 11000_2$

Parte fraccionaria:

$0,6875 \times 2 = 1,3750 \Rightarrow 0,6875 = 0,1011$

$0,3750 \times 2 = 0,750$

$0,75 \times 2 = 1,5$

$0,5 \times 2 = 1$

$-24,6875 = -11000,1011 =$

$= -1,10001011 \times 2^4$

$S=1 \quad E=4+127=131=10000011_2$
 $F=10001011$

$$\begin{array}{c}
 \text{S} \quad \text{E} \quad \text{F} \\
 1 \mid 10000011 \mid 100010110000000000000000
 \end{array}$$

C 1 C 5 8 0 0 0

- Cálculos:**

$A = -1,00101011 \times 2^3$
 $B = -1,10001011 \times 2^4$
 Igualamos las características desplazando la mantisa correspondiente a la característica menor:

$$\begin{array}{r} A = -0,100101011 \times 2^4 \\ B = -1,100010110 \times 2^4 \\ \hline -10,001000001 \times 2^4 \end{array}$$

Como ambos son negativos
los sumamos dejando el
signo - en el resultado

$-10,001000001 \times 2^7$

Dado que el resultado no ha quedado normalizado,
lo normalizamos ajustando la característica:

$S \quad S = S + 127 = 132 =$

$$-1.0001000001 \times 2^5$$

$$E = 5 + 127 = 132 = 10000100_2$$

S E
1'10000100'000100000100000000000000
C 2 0 8 2 0 0 0 0 0 0 0 0

... auf der Karte

No habrá desbordamiento ni subdesbordamiento porque hemos conseguido representar el resultado en la forma $(-1)^S \times 1, F \times 2^{E-E_{\text{exceso}}}$

Los bits de reserva son 0 porque no ha salido ningún bit por la derecha que exceda los 23 bits de la mantisa

5. Para cada una de las afirmaciones siguientes, indique si es verdadera o falsa rodeando con un círculo la V si la considera verdadera y la F si la considera falsa: (1 punto)

Cuando efectuamos una suma en complemento a 2, se produce desbordamiento siempre que tenemos llevada al sumar los dígitos de mayor peso de los sumandos.	V	<input checked="" type="radio"/> F
Si se restan dos números de igual signo representados en complemento a 2 con 32 bits no puede haber desbordamiento.	<input checked="" type="radio"/> V	F
En los algoritmos de multiplicación hay que reservar para el resultado un registro con doble número de bits que cada uno de registros de los operandos (supuesto que ambos operandos tienen igual longitud).	<input checked="" type="radio"/> V	F
En el algoritmo de división con restauración ésta consiste en sumar el divisor al resto parcial cuando el respectivo dígito del cociente es 0.	<input checked="" type="radio"/> V	F
El rango de una representación en punto flotante depende del número de bits dedicados a representar a la mantisa.	V	<input checked="" type="radio"/> F

Prueba práctica voluntaria de Fundamentos de Computadores. Grupo G2. 21/05/13

APELLIDOS Y NOMBRE:

DNI:

EJERCICIO 1. (0,5 puntos)

Escriba el código en lenguaje ensamblador MIPS correspondiente a la sentencia de lenguaje de alto nivel: $w = x + y - (4 * z - 5)$. Supóngase que las variables x , y , z y w son enteras y se encuentran, respectivamente, en los registros $\$s0$, $\$s1$, $\$s2$ y $\$s3$, y que los contenidos de los citados registros no pueden modificarse (esto es, debe conservarse el valor de las variables x , y y z).

```
sll $s3, $s2, 2      # 4z
addi $s3, $s3, -5     # 4z-5
sub $s3, $s1, $s3      # x-(4z-5)
add $s3, $s3, $s0      # x+y-(4z-5)
```

EJERCICIO 2. (1 punto)

Escriba el código en lenguaje ensamblador MIPS correspondiente a la sentencia de lenguaje de alto nivel: $a[i][j] = a[i][j-1] + a[i][j-2]$. Supóngase que a es una matriz cuadrada de 10×10 componentes enteras cuya dirección se encuentra en el registro $\$s5$, que el índice i se halla en el $\$s0$ y que el índice j se encuentra en el $\$s1$. Incluya también la declaración de la citada matriz.

Declaración de la matriz:

```
a: .space 400
```

Código:

```
addi $t0, $t0, 10
mul $t0, $t0, $s0 # 10i
add $t0, $t0, $s1 # 10i+j
sll $t0, $t0, 2 # 4(10i+j)
add $t0, $t0, $s5 # a+4(10i+j)
lw $t1, -4($t0) # a[i][j-1]
lw $t2, -8($t0) # a[i][j-2]
add $t1, $t1, $t2
sw $t1, 0($t0)
```

EJERCICIO 3. (1 punto)

Expresa, en decimal el contenido del registro \$f4 después de la ejecución de la siguiente porción de código:

<pre> .data P: .float 20.0, 150.0, 40.0, 13.7 . . . la \$t0, P l.s \$f0, 0(\$t0) l.s \$f1, 4(\$t0) l.s \$f2, 8(\$t0) add.s \$f3, \$f1, \$f0 s.s \$f3, 12(\$t0) addi \$t1, \$t0, 12 l.s \$f4, 0(\$t1) div.s \$f4, \$f4, \$f2 </pre>	<p>Contenido final de \$f4 (decimal): 4.25</p> <p>Explicación:</p> <p>$\\$t0 \leftarrow \text{dirección } P$ $\\$f0 \leftarrow 20.0$ $\\$f1 \leftarrow 150.0$ $\\$f2 \leftarrow 40.0$ $\\$f3 \leftarrow \\$f1 + \\$f0 = 170.0$ Se guarda 170 en $P+12$ (en lugar de 13.7) $\\$t1 \leftarrow \\$t0 + 12 = P+12$ $\\$f4 \leftarrow 170$ $\\$f4 \leftarrow \\$f4 / \\$f2 = 170 / 40 = 4.25$</p>
--	--

EJERCICIO 4. (1 punto)

<p>Escriba una función en lenguaje ensamblador de MIPS denominada intercambio que tome tres parámetros:</p> <ul style="list-style-type: none"> • La dirección de un vector cuyas componentes son números enteros. • Dos índices (i y j) <p>La función debe intercambiar las componentes i y j del vector.</p> <p>Parámetros:</p> <p>$\\$a0 \leftarrow \text{Dirección vector}$ $\\$a1 \leftarrow i$ $\\$a2 \leftarrow j$</p>	<p>Código de la función:</p> <p>intercambio:</p> <pre> sll \$t0, \$a1, 2 add \$t0, \$t0, \$a0 sll \$t1, \$a2, 2 add \$t1, \$t1, \$a0 lw \$t2, 0(\$t0) lw \$t3, 0(\$t1) sw \$t2, 0(\$t1) sw \$t3, 0(\$t0) jr \$ra </pre>
--	--

EJERCICIO 5. (1,5 puntos)

Supuesta construida la función anterior, escriba un programa que declare e inicialice un vector de enteros con los valores 1, 5, 8, 7, 4, 2, 10 y 320, llame a la función anterior para intercambiar la primera componente con la última y luego imprima el nuevo vector con una componente en cada línea.

Datos:**Función para impresión de números enteros:**

syscall con \$v0=1 . Número a imprimir en \$a0

Función para impresión de caracteres:

syscall con \$v0=11 . Código ASCII del carácter a imprimir en \$a0.

Códigos ASCII de caracteres usuales:

Espacio en blanco: 32 (decimal).

Retorno de carro: 13 (decimal).

Código:

```
.data
x: .word 1, 5, 8, 7, 4, 2, 10, 320
.globl --start
.text
--start:
    la $s0, x
    add $a0, $s0, $zero
    add $a1, $zero, $zero
    addi $a2, $zero, 7
    jal intercambio
    addi $s1, $zero, 8
bucle:
    lw $a0, 0($s0)
    li $v0, 1
    syscall      # imprime x[i]
    addi $a0, $zero, 13    # 13=CR
    li $v0, 11
    syscall      # imprime CR
    addi $s0, $s0, 4
    addi $s1, $s1, -1
    bne $s1, $zero, bucle
    li $v0, 10
    syscall      # Salida del programa
```