

## Examen Segundo Parcial de FCO – Temas 5 y 6

10 de Enero de 2020

APELLIDOS: \_\_\_\_\_

NOMBRE: \_\_\_\_\_

DNI: \_\_\_\_\_ GRUPO: \_\_\_\_\_

FIRMA: \_\_\_\_\_

Normativa:

- La duración del examen es de 2 horas
- Escriba el Nombre, Apellidos y Grupo y firme en la primera hoja.
- Debe responder en el espacio asignado.
- No se permiten calculadoras ni apuntes.
- Debe permanecer en silencio durante la realización del examen.
- No se puede abandonar el examen hasta que el profesor lo indique.
- Debe tener una identificación en la mesa a la vista del profesor (DNI, carnet UPV, tarjeta residente, etc.)

1.- (1,5 puntos) Complete la tabla siguiente:

Decimal	Signo y Magnitud 8 bits	Complemento a 2 8 bits	Exceso 127 8 bits
-7	10000111	11111001	01111000
-13	10001101	11110011	01110010
Rango	[-127, +127]	[-128, +127]	[-127, +128]

**Nota:** Indique el rango en decimal

2.- (1 punto) Dados los números enteros  $A = 111111_{Ca2}$  y  $B = 100000_{Ca2}$  representados en complemento a 2 con 6 bits, realice las operaciones siguientes sin cambiar de representación. Indique claramente y justifique si hay o no hay desbordamiento. Muestre el detalle de su solución.

a)  $A + B$  (0,4 puntos)

Dado que se trata de una suma, los números se suman tal como están:

100000. (Acarreos)

111111 (A)

+100000 (B)

011111 (se descarta el acarreo final)

Los dos últimos bits de acarreos, indicados por el recuadro, son diferentes, luego Sí hay desbordamiento.

El resultado de  $A + B$  es: Número no representable se produce desbordamiento.

**b) B - A (0,6 puntos).**

Como se trata de una resta, convertimos la resta en una suma con la operación  $B + \text{Ca2}(A)$  (se cambia el signo al sustraendo haciendo el complemento a dos, y se convierte la resta en una suma):

$$\text{Ca2}(A) = \text{Ca2}(111111) = 000001$$

Ahora realizamos la suma  $B + \text{Ca2}(A)$

000000

100000

+000001

100001

Los dos últimos bits de acarreo, indicados por el recuadro, son IGUALES, luego NO hay desbordamiento.

El resultado es válido:  $B - A = 100001_{\text{Ca2}}$  y es igual a  $-31_{10}$

**3.- (1 punto)** Dado el número real representado en el formato IEEE-754 de simple precisión  $R = 0xC1F10000$ , obtenga su representación en DECIMAL. Muestre los pasos realizados y la representación final en decimal.

$R = 1 \mid 10000011 \mid 111000100000000000000000$

Signo = 1, Negativo  
Magnitud = 1,1110001

Exponente + Exceso = Número Representado

Exceso=127  
Número Representado = 131

Exponente =  $131 - 127 = 4$

La solución es:  $(R = \text{Signo Magnitud} * 2^{\text{Exponente}})$

Signo: -  
Magnitud: 1,1110001  
Exponente (en decimal): 4

$-11110,001 \times 2^0$

**SOLUCIÓN:**  $R = -30,125_{10}$

**4.- (5,5 puntos)** A partir del siguiente código, escrito en ensamblador del MIPS R2000:

```
.globl __start
.data 0x10001000
Tam: .byte 5
A1: .half 5, 8, -7, 9, 4
letrero: .asciiz "GII-fco"
A2: .byte 4, 7, -6, 8, 5
resultado: .word 0

.text 0x00400100
__start: la $2, Tam
        lb $2, 0($2)
        li $10, 0
        la $3, A1
        la $4, A2

mientras: beq $2, $0, fin
        lh $5, 0($3)
        lb $6, 0($4)
        sub $8, $5, $6
        mult $8, $8
        mflo $8
        add $10, $10, $8
        addi $2, $2, -1
        addi $3, $3, 2
        addi $4, $4, 1
        j mientras

fin:     la $2, Tam
        lb $2, 0($2)
        div $10, $2
        mflo $10
        la $2, resultado
        sw $10, 0($2)
        .end
```

Responda a las siguientes preguntas:

- a) **(0,5 puntos)** Indique el contenido del segmento de datos antes de ejecutarse el programa, teniendo en cuenta que los datos se almacenan en formato “little endian”. El contenido debe especificarse por cada byte, en hexadecimal para los datos numéricos, y con los caracteres correspondientes entre comillas en el caso de las cadenas. Indique claramente las zonas de memoria de contenido desconocido mediante un interrogante o guión.

31	...	24	23	...	16	15	...	8	7	...	0	Dirección
0x00			0x05			-			0x05			0x10001000
0xff			0xf9			0x00			0x08			0x10001004
0x00			0x04			0x00			0x09			0x10001008
'_'			'l'			'l'			'G'			0x1000100c
0x00			'o'			'c'			'f'			0x10001010
0x08			0xfa			0x07			0x04			0x10001014
-			-			-			0x05			0x10001018
0x00			0x00			0x00			0x00			0x1000101c

- b) (1 punto) Indique el contenido en hexadecimal de los siguientes registros **después** de ejecutar por **primera** vez la instrucción `j mientras`.

Registro	Contenido Hexadecimal
\$2	0x00000004
\$3	0x10001004
\$4	0x10001015
\$5	0x00000005
\$6	0x00000004

- c) (1 punto) Indique el contenido de los siguientes registros al finalizar completamente la ejecución del programa. Exprese el contenido en hexadecimal.

Registro	Contenido Hexadecimal
\$3	0x1000100c
\$4	0x10001019
\$5	0x00000004
\$6	0x00000005
\$10	0x00000001

- d) **(0,5 puntos)** Indique solo las zonas de la memoria que se hayan modificado tras la ejecución del programa. Expresé el valor de cada byte en hexadecimal.

31 ... 24	23 ... 16	15 ... 8	7 ... 0	Dirección
0x00	0x00	0x00	0x01	0x1000101c

- e) **(0,5 puntos)** Utilizando una ecuación indique cuál es el resultado que se almacena en el registro \$10.

$$f(A1[], A2[]) = \frac{\left( \sum_{i=0}^{n-1} (A1[i] - A2[i])^2 \right)}{n}$$

- f) **(0,5 puntos)** ¿Cuál es el resultado si se cambian los valores del vector por los siguientes bytes?:

A2: byte 3, 6, -9, 11, 6

El resultado sería igual a 4.

- g) **(0,5 puntos)** Indique los valores asociados a las siguientes etiquetas: (en hexadecimal)

Tam	0x10001000
A1	0x10001002
A2	0x10001014
__start	0x00400100
mientras	0x00400120

- h) **(0,5 puntos)** Indique la secuencia de instrucciones por las que el ensamblador del MIPS R2000 traduciría la pseudoinstrucción:

la \$4, A2

lui \$1, 0x1000

ori \$4, \$1, 0x1014

NOTA: Debe utilizar el registro \$1 para los cálculos intermedios.

- i) **(0,5 puntos)** Codifique la instrucción `div $10,$2`. Indique el resultado en binario y hexadecimal y detalle los pasos realizados.

Instrucción tipo R: COP rs rt rd Desp funcion

COP = 0x00 funcion = 0x1A

rs = \$10 rt = \$2 rd = \$0 Desp=0x00

Binario: 000000 01010 00010 00000 00000 011010

Hexadecimal: 0x0142001a

**5.- (1 punto)** Realice un programa en ensamblador del MIPS R2000 que realice la operación:

$$\text{var\_d} = (\text{var\_a} * \text{var\_b}) / \text{var\_c}$$

Teniendo en cuenta las siguientes consideraciones:

- Los datos “var\_a”, “var\_b” y “var\_c” se definirán como enteros de 16 bits ubicados a partir de la posición de memoria 0x10000000
- A continuación, se reservará espacio para almacenar el “resultado”, var\_d, como un entero de 32 bits
- Considere los siguientes valores:
  - var\_a= 20
  - var\_b= 10
  - var\_c= 5
- Etiquete en la memoria de datos cada elemento con su valor alfabético (var\_a, var\_b, var\_c, var\_d).

### Posible solución

```
.globl __start

.data 0x10001000

var_a: .half 20
var_b: .half 10
var_c: .half 5
var_d: .word 0

.text 0x00400000
__start:    la $2,var_a
            lh $2,0($2)    # $2 = contenido de var_a (20)
            la $3,var_b
            lh $3,0($3)    # $3 = contenido de var_b (10)
            mult $2,$3
            mflo $2        # $2 = $2 x $3 (var_a x var_b)

            la $3,var_c
            lh $3,0($3)    # $3 = contenido de var_c (5)
            div $2,$3
            mflo $2        # $2 = cociente división $2/$3

            la $3,var_d
            sw $2,0($3)    # actualizar var_d con el cociente

.end
```