

Examen Segundo Parcial de FCO – Tems 5 y 6

18 de Enero de 2017

APELLIDOS: _____ NOMBRE: _____

DNI: _____

FIRMA: _____

Normativa:

- La duración del examen es de 2 horas
- Escriba el nombre y los apellidos y firme en TODAS las hojas.
- 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 9 bits	Complemento a 2 9 bits	Exceso 255 9 bits
+5	000000101	000000101	100000100
-12	100001100	111110100	011110011
Rango	[-255 , +255]	[-256 , +255]	[-255 , +256]

Nota: Indique el rango en decimal

2.- (1 puntos) Dados los números enteros $A = 01110011_{Ca2}$ y $B = 10011101_{Ca2}$ representados en complemento a 2 con 8 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:

11111111

01110011

+10011101

400010000 (se descarta el acarreo final)

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

El resultado de $A + B$ es: 00010000_{Ca2}

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

Como se trata de una resta, se cambia el signo al sustraendo haciéndole el complemento a dos, y se convierte la resta en una suma:

$$Ca_2(B) = Ca_2(10011101) = 01100011$$

Ahora realizamos la suma $A + (-B)$

01100011

01110011

+01100011

011010110

Los dos últimos bits de acarreo, indicados por el recuadro, son DISTINTOS, luego SÍ hay desbordamiento.

No hay resultado válido porque ha habido desbordamiento

3.- (1 punto) Dado el número decimal real -259,75, obtenga su representación en el formato de IEEE-754 de simple precisión. Muestre los pasos realizados y la representación final en hexadecimal.

$$259_{10} = 2^8 + 2^1 + 2^0 = 100000011_2$$

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

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

$$0,75_{10} = 0,11_2$$

$$-259,75_{10} = -100000011,11_2$$

$$\text{Normalizamos } 100000011,11_2 = 1,0000001111_2 \times 2^8$$

$$\text{Expresamos el exponente (6) en exceso 127 (127+8)= } 135_{10} = 10000111_2$$

Los campos del número en IEEE 754 serían los siguientes:

1 | 10000111 | 0000001111000000000000

Pasamos a hexadecimal: 1100 0011 1000 0001 1110 0000 0000 0000
C 3 8 1 E 0 0 0

Solución final: 0xC381E000

Examen Segundo Parcial de FCO – Temas 5 y 6

APELLIDOS: _____

NOMBRE: _____

DNI: _____

FIRMA: _____

4.- A partir del siguiente código, escrito en ensamblador del MIPS R2000:

```
.globl __start
.data 0x10000000
Tam: .byte 7
Vector: .half 12, 3, 4, 5, 6, 10, 14
Media: .word 0

.text 0x00400000
__start: la $2, Tam
        lb $2, 0($2)
        la $3, Vector
        li $4, 0
mientras: beq $2, $0, fin
        lh $5, 0($3)
        addi $4, $4, $5
        addi $2, $2, -1
        addi $3, $3, 2
        j mientras

fin:     la $2, Tam
        lb $2, 0($2)
        div $4, $2
        mflo $4
        la $3, Media
        sw $4, 0($3)
.end
```

Responda a las siguientes preguntas:

- a) (1 punto) Indique el contenido de los siguientes registros **después** de ejecutar por **primera** vez la instrucción J mientras.

Registro	Contenido Hexadecimal
\$2	0x00000006
\$3	0x10000004
\$4	0x0000000C
\$5	0x0000000C

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

Registro	Contenido Hexadecimal
\$2	0x00000007
\$3	0x10000010
\$4	0x00000007
\$5	0x0000000E

- c) **(0,5 puntos)** Indique el contenido del segmento de datos después 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	0x0C	–	0x07	0x10000000
0x00	0x04	0x00	0x03	0x10000004
0x00	0x06	0x00	0x05	0x10000008
0x00	0x0E	0x00	0x0A	0x1000000C
0x00	0x00	0x00	0x00	0x10000010

- d) **(0,5 puntos)** Memoria después de la ejecución del programa. Solo las direcciones que se modifiquen, en hexadecimal.

31 ... 24	23 ... 16	15 ... 8	7 ... 0	Dirección
0x00	0x00	0x00	0x07	0x10000010

Examen Segundo Parcial de FCO – Temas 5 y 6

APELLIDOS: _____

NOMBRE: _____

DNI: _____

FIRMA: _____

- e) **(1 punto)** Indique la instrucción o instrucciones que cambiaría en el programa si los datos del vector pasaran a ser de tipo byte. O sea, si la directiva de reserva del vector fuera:

Vector: `.byte 12, 3, 4, 5, 6, 10, 14`

Cambiaría _____ instrucciones.

Instrucción _____ `lh $5, 0($3)` _____ por _____ `lb $5, 0($3)` _____

Instrucción _____ `addi $3, $3, 2` _____ por _____ `addi $3, $3, 1` _____

Instrucción _____ por _____

Instrucción _____ por _____

NOTA 1: se pueden cambiar 1, 2, 3 ó 4 instrucciones. No todos los huecos hay que rellenarlos si no es necesario.

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

Tam `0x10000000`

Vector `0x10000002`

Media `0x10000010`

__start `0x00400000`

Mientras `0x00400014`

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

`la $3, Vector`

`lui $1, 0x1000`

`ori $3, $1, 2`

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

h) (0,5 puntos) Codifique la instrucción `sw $4,0($3)`. Indique el resultado en binario y hexadecimal y detalle los pasos realizados.

CO= 0x2B RS= 0x3 RT= 0x4 Desp= 0x0

Binario: 101011000110010...0

Hexadecimal: 0xAC640000

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

`var_d = var_a + var_b - var_c`

teniendo en cuenta las siguientes consideraciones:

- Los datos “var_a”, “var_b” y “var_c” se definirán como enteros de 8 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= 9
 - var_b= -2
 - var_c= 3

Etiquete en la memoria de datos cada elemento con su valor alfabético (var_a, var_b, var_c, var_d).

```
.data 0x10000000
var_a: .byte 9
var_b: .byte -2
var_c: .byte 3
var_d: .word 0 (alternativa: .space 5)

.globl __start
.text 0x00400000
__start:
la $8, var_a
lb $8, 0($8)    #$8 <- var_a
la $9, var_b
lb $9, 0($9)    #$9 <- var_b
la $10, var_c
lb $10, 0($10)  #$10 <- var_c

add $2, $8, $9   #(var_a + var_b)
sub $2, $2, $10  #(var_a + var_b) - var_c

la $7, var_d     #quizá alguien lo etiquete como resultado
sw $2, 0($7)     #almacena resultado en Memoria
```

