

Examen Segon Parcial d'FCO – Temes 5 i 6

18 de Gener de 2017

COGNOMS: _____

NOM: _____

DNI: _____

SIGNATURA: _____

Normativa:

- La durada de l'examen és de 2 hores.
- Escriviu el nom i els cognoms en lletres MAJÚSCULES i signeu en TOTS els fulls.
- Heu de respondre dins l'espai assignat.
- No es permeten calculadores ni apunts.
- Heu de romandre en silenci durant la realització de l'examen.
- No es pot abandonar l'examen fins que el professor ho indique.
- Heu de tenir una identificació damunt la taula, a la vista del professor (DNI, carnet UPV, targeta de resident, etc.).

1.- (1,5 punts) Completeu la taula següent: (els valors en negreta són l'enunciat i el valors en cursiva són els que es demanen a l'alumne)

Decimal	Signe i Magnitud 9 bits	Complement a 2 9 bits	Excés 255 9 bits
+5	000000101	000000101	100000100
-12	100001100	111110100	011110011
Rang	[-255 , +255]	[-256 , +255]	[-255 , +256]

Nota: Indiqueu el rang en decimal

2.- (1 punt) Amb els nombres enters representats en complement a 2 amb 8 bits, $A = 01110011_{Ca2}$ i $B = 10011101_{Ca2}$, realitzeu les operacions següents sense canviar de representació. Mostreu el detall de les operacions i indiqueu clarament i justifiqueu si hi ha o no desbordament.

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

Com que es tracta d'una suma, els nombres se sumen tal com estan:

11111111

01110011

+10011101

00010000 (el carry final es descarta)

Els dos últims bits de carry són iguals, per tant no hi ha DESBORDAMENT i el resultat d' $A+B$ és

00010000_{Ca2}

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

Com que es tracta d'una operació de resta, és necessari canviar el signe del subtrahend. Fent el complement a dos al subtrahend li canviem el signe i d'aquesta manera convertim la resta en suma.

$$\text{Ca2}(B) = \text{Ca2}(10011101) = 01100011$$

Ara realitzem l'operació $A + (-B)$:

$$\begin{array}{r} 01100011 \\ 01110011 \\ +01100011 \\ \hline 11010110 \quad (\text{el carry final es descarta}) \end{array}$$

Els dos últims bits de *carry* són diferents i SÍ hi ha DESBORDAMENT i per tant NO hi ha resultat.

3.- (1 punt) Obteniu la representació del nombre real -259,75 en el format de simple precisió de l'estàndard IEEE-754. Mostreu els passos seguits i la representació final en hexadecimal.

Convertim la part entera a binari: $259_{10} = 2^8 + 2^1 + 2^0 = 100000011_2$

Convertim la part fraccionaria a binari:

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

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

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

Ajuntem les dues parts: $259,75_{10} = 100000011,11_2$

Convertim a coma flotant: $100000011,11_2 \times 2^0$

Normalitzem la mantissa: $100000011,11_2 \times 2^0 = 1,0000001111_2 \times 2^8$

Representem l'exponent (8) en excés 127: $127+8 = 135_{10} = 10000111_2$

Completem el format de simple precisió d'IEEE 754 en binari:

S Exponent Mantissa

1 | 10000111 | 0000001111000000000000

Convertim la representació binària a hexadecimal:

1100 0011 1000 0001 1110 0000 0000 0000

C 3 8 1 E 0 0 0

Solució final: 0xC381E000

- 4.- Considerant el programa següent, escrit en llenguatge d'assemblador del MIPS R2000, responeu les qüestions que es presenten a continuació

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

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

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

- a) (1 punt) Indiqueu el contingut dels registres següents **després** d'executar per primera vegada la instrucció *J mentre*.

Registre	Contingut Hexadecimal
\$2	0x00000006
\$3	0x10000004
\$4	0x0000000C
\$5	0x0000000C

- b) **(1 punt)** Indiqueu el contingut dels registres següents en finalitzar correctament l'execució del programa. Expresseu el contingut en hexadecimal.

Registre	Contingut Hexadecimal
\$2	0x00000007
\$3	0x10000010
\$4	0x00000007
\$5	0x0000000E

- c) **(0,5 punts)** Indiqueu el contingut del segment de dades abans d'executar el programa. Teniu en compte que les dades s'emmagatzemen en format "little endian", i heu d'indicar el contingut de cadascun dels bytes en hexadecimal. Indiqueu les zones de memòria de contingut desconegut amb un interrogant o un guionet.

31 ... 24	23 ... 16	15 ... 8	7 ... 0	Adreça
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 punts)** Indiqueu en la taula següent només les posicions de memòria que s'han modificat en executar el programa. Escriviu els valors en hexadecimal per a cadascun dels bytes.

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

- e) **(1 punt)** Indiqueu la instrucció o instruccions que caldria canviar en el programa si les dades del vector passaren a ser de tipus byte. És a dir, si la directiva de reserva de memòria per al vector fóra la següent:

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

Canviaria ____2____ instruccions.

Instrucció _____ lh \$5, 0(\$3) _____ per _____ lb \$5, 0(\$3) _____

Instrucció _____ addi \$3, \$3, 2 _____ per _____ addi \$3, \$3, 1 _____

Instrucció _____ per _____

Instrucció _____ per _____

NOTA Pot ser cal canviar 1, 2, 3 o 4 instruccions. És a dir, si no és necessari no cal omplir totes les files..

Canviaria ____2____ instruccions.

Instrucció _____ lh \$5, 0(\$3) _____ per _____ lb \$5, 0(\$3) _____

Instrucció _____ addi \$3, \$3, 2 _____ per _____ addi \$3, \$3, 1 _____

Instrucció _____ per _____

Instrucció _____ per _____

NOTA Pot ser cal canviar 1, 2, 3 o 4 instruccions. És a dir, si no és necessari no cal omplir totes les files..

Canviaria ____2____ instruccions.

Instrucció _____ lh \$5, 0(\$3) _____ per _____ lb \$5, 0(\$3) _____

Instrucció _____ addi \$3, \$3, 2 _____ per _____ addi \$3, \$3, 1 _____

Instrucció _____ per _____

Instrucció _____ per _____

NOTA Pot ser cal canviar 1, 2, 3 o 4 instruccions. És a dir, si no és necessari no cal omplir totes les files..

Canviaria ____2____ instruccions.

Instrucció _____ lh \$5, 0(\$3) _____ per _____ lb \$5, 0(\$3) _____

Instrucció _____ addi \$3, \$3, 2 _____ per _____ addi \$3, \$3, 1 _____

Instrucció _____ per _____

Instrucció _____ per _____

NOTA Pot ser cal canviar 1, 2, 3 o 4 instruccions. És a dir, si no és necessari no cal omplir totes les files..

Canviaria ____2____ instruccions.

Instrucció _____ lh \$5, 0(\$3) _____ per _____ lb \$5, 0(\$3) _____

Instrucció _____ addi \$3, \$3, 2 _____ per _____ addi \$3, \$3, 1 _____

Instrucció _____ per _____

Instrucció _____ per _____

NOTA Pot ser cal canviar 1, 2, 3 o 4 instruccions. És a dir, si no és necessari no cal omplir totes les files..

Canviaria ____2____ instruccions.

Instrucció _____ lh \$5, 0(\$3) _____ per _____ lb \$5, 0(\$3) _____

Instrucció _____ addi \$3, \$3, 2 _____ per _____ addi \$3, \$3, 1 _____

Instrucció _____ per _____

Instrucció _____ per _____

NOTA Pot ser cal canviar 1, 2, 3 o 4 instruccions. És a dir, si no és necessari no cal omplir totes les files..

f) **(0,5 punts)** Indiqueu en hexadecimal els valors associats a les etiquetes següents:

f) **(0,5 punts)** Indiqueu en hexadecimal els valors associats a les etiquetes següents:

g) **(0,5 punts)** Indiqueu la seqüència d'instruccions per les que l'assemblador del MIPS R2000 traduiria la pseudoinstrucció:

g) **(0,5 punts)** Indiqueu la seqüència d'instruccions per les que l'assemblador del MIPS R2000 traduiria la pseudoinstrucció:

lui \$1, 0x1000
ori \$3, \$1, 2
NOTA: utilitzeu el registre \$1 per als valors intermedis.

lui \$1, 0x1000
ori \$3, \$1, 2
NOTA: utilitzeu el registre \$1 per als valors intermedis.

lui \$1, 0x1000
ori \$3, \$1, 2
NOTA: utilitzeu el registre \$1 per als valors intermedis.

h) **(0,5 punts)** Codifiqueu la instrucció `sw $4,0($3)`. Indiqueu el resultat tant en binari com en hexadecimal, i detal·leu els passos realitzats.

C0= 0x2B RS= 0x3 RT= 0x4 Desp= 0x0				
	COP	Rs	Rt	Desp
Binari:	101011	00011	00100	0000000000000000
Hexadecimal:	0xAC640000			

C0= 0x2B RS= 0x3 RT= 0x4 Desp= 0x0				
	COP	Rs	Rt	Desp
Binari:	101011	00011	00100	0000000000000000
Hexadecimal:	0xAC640000			

C0= 0x2B RS= 0x3 RT= 0x4 Desp= 0x0				
	COP	Rs	Rt	Desp
Binari:	101011	00011	00100	0000000000000000
Hexadecimal:	0xAC640000			

C0= 0x2B RS= 0x3 RT= 0x4 Desp= 0x0				
	COP	Rs	Rt	Desp
Binari:	101011	00011	00100	0000000000000000
Hexadecimal:	0xAC640000			

5.- (1 punt) Escriviu un programa en assemblador del MIPS R2000 que realitzi l'operació:

var_d= var_a+var_b-var_c

tenint en compte les consideracions següents:

- Les dades “var_a”, “var_b” i “var_c” han de definir-se com a enters de 8 bits ubicats a partir de la posició de memòria 0x10000000.
- Darrere d'aquestes dades es reservarà espai en memòria per a emmagatzemar el resultat, var_d, com a enter de 32 bits.
- Considereu el valors següents per a les variables:
 - var_a= 9
 - var_b= -2
 - var_c= 3

Etiqueteu en la memòria de dades cadascun dels elements amb el corresponent valor alfabètic (var_a, var_b, var_c, var_d).

Solució: com sempre, poden haver versions diferents d'un programa i ser totes correctes. Es presenten, com a exemple, dues solucions:

Solució A:	Solució B:
<pre>.data 0x10000000 a: .byte 9 b: .byte -2 c: .byte 3 d: .space 5 #Alineació manual .globl __start .text 0x00400000 __start: la \$8, a lb \$8, 0(\$8) #\$8 <- a la \$9, b lb \$9, 0(\$9) #\$9 <- b la \$10, c lb \$10, 0(\$10) #\$10 <- c add \$2, \$8, \$9 #(a + b) sub \$2, \$2, \$10 #(a + b)- c la \$7, d sw \$2, 0(\$7) #Emmagatzema el resultat</pre>	<pre>.data 0x10000000 a: .byte 9, -2, 3 d: .word 0 #Alineació automàtica .globl __start .text 0x00400000 __start: la \$8, a lb \$8, 0(\$8) #\$8 <- a lb \$9, 1(\$9) #\$9 <- b lb \$10, 2(\$10) #\$10 <- c add \$2, \$8, \$9 #(a + b) sub \$2, \$2, \$10 #(a + b)- c la \$7, d sw \$2, 0(\$7) #Emmagatzema el resultat</pre>