

## Tema 2. EL LENGUAJE DEL COMPUTADOR. Problemas

**2.1.** Sean las siguientes instrucciones del lenguaje C en que todas las variables son enteras de 32 bits:

a)  $f = g + h + i + j$ ;

b)  $f = f + f + i$ ;

- I) Para cada una de las instrucciones anteriores, muestre el código correspondiente en lenguaje ensamblador de MIPS, empleando el mínimo número de instrucciones y suponiendo que las variables  $f$ ,  $g$ ,  $h$ ,  $i$  y  $j$  se encuentran, respectivamente, en los registros  $\$s0$ ,  $\$s1$ ,  $\$s2$ ,  $\$s3$  y  $\$s4$ .
- II) ¿Cuántas instrucciones de MIPS son necesarias para ejecutar cada una de las instrucciones anteriores de lenguaje C?
- III) Si las variables  $f$ ,  $g$ ,  $h$ ,  $i$  y  $j$  tienen los valores 1, 2, 3, 4 y 5 respectivamente ¿Cuál será el valor final de  $f$ ?

**2.2.** Sean las siguientes instrucciones de lenguaje ensamblador de MIPS

a) `add $s0, $s1, $s2`

b) `add $s0, $s1, $s1`

`add $s0, $s0, $s2`

Supongamos que los registros  $\$s0$ ,  $\$s1$  y  $\$s2$  corresponden respectivamente a las variables  $f$ ,  $g$  y  $h$  que son enteras y de 32 bits:

- I) ¿Cuál es la instrucción del lenguaje C correspondiente para cada caso?
- II) Si las variables  $f$ ,  $g$  y  $h$  tienen respectivamente los valores 1, 2 y 3: ¿Cuál es el valor final de  $f$ ?

**2.3.** Sean las siguientes instrucciones de lenguaje ensamblador de MIPS

a) `add $s0, $s0, $s1`

b) `add $t0, $s0, $s1`

`add $s0, $s0, $s2`

`sub $t1, $s2, $s3`

`add $s0, $s0, $s3`

`sub $s0, $t0, $t1`

`add $s0, $s0, $s4`

- I) ¿Cuál es la instrucción del lenguaje C correspondiente al código MIPS para cada caso?
- II) Reescriba las instrucciones de lenguaje ensamblador de MIPS para cada uno de los casos de forma que se efectúe la misma función pero minimizando el número de instrucciones, si ello fuera posible.
- III) ¿Cuántos registros son necesarios para ejecutar cada una de las funciones? Reescriba el código para efectuar la misma función minimizando el número de registros utilizados

**2.4.** Sean las siguientes instrucciones del lenguaje C en que todas las variables son enteras de 32 bits:

a)  $f = A[1] + g$ ;

b)  $A[4] = f + B[8]$ ;

- I) Para cada una de las instrucciones anteriores, muestre el código correspondiente en lenguaje ensamblador de MIPS, empleando el mínimo número de instrucciones y suponiendo que las variables  $f$  y  $g$  se encuentran, respectivamente, en los registros  $\$s0$  y  $\$s1$  y que las direcciones de los vectores  $A$  y  $B$  se encuentran, respectivamente, en los registros  $\$s6$  y  $\$s7$
- II) ¿Cuántas instrucciones de MIPS son necesarias para ejecutar cada una de las instrucciones anteriores de lenguaje C?
- III) Para cada una de las instrucciones anteriores de lenguaje C: ¿Cuántos registros diferentes son necesarios para ejecutar la instrucción?

**2.5.** Sean las siguientes instrucciones de lenguaje ensamblador de MIPS

a) `sw $s0, 4($s6)`

b) `lw $t0, 12($s6)`

`sw $t0, 8($s6)`

- I) ¿Cuál es la instrucción del lenguaje C correspondiente al código MIPS para cada caso?
- II) Reescriba las instrucciones de lenguaje ensamblador de MIPS para cada uno de los casos de forma que se efectúe la misma función pero minimizando el número de instrucciones, si ello fuera posible.
- III) ¿Cuántos registros son necesarios para ejecutar cada una de las funciones? Reescriba el código para efectuar la misma función minimizando el número de registros utilizados

**2.6.** Un computador MIPS tiene en su memoria los datos mostrados en cada una de las siguientes tablas (dos casos):

a)

Dirección	Dato
24	2
28	4
32	3
36	6
40	1

b)

Dirección	Dato
16	13
20	20
24	12
28	45
32	5

- Para las posiciones de memoria mostradas en la tabla, escriba el código C necesario para ordenar los datos de menor a mayor, dejando el menor en la posición de memoria con dirección más baja. Supóngase que los datos mostrados representan una variable de C denominada Array que es un vector de enteros de 4 bytes.
- Para los datos mostrados en las tablas superiores, escriba el código MIPS para ordenar los datos de menor a mayor, dejando el menor en la posición de memoria con dirección más baja. Trate de utilizar el mínimo número de instrucciones posible. Suponga que la dirección del vector Array se halla en el registro \$s6.
- Para cada uno de los programas de ordenación del apartado anterior: ¿Cuántas instrucciones MIPS son necesarias?
- Si no estuviera permitido utilizar desplazamientos en las instrucciones lw y sw: ¿Cuántas instrucciones se necesitarían?

**2.7.** Sean las siguientes instrucciones del lenguaje C en que todas las variables son enteras de 32 bits:

a)  $f = g + (i + 5);$

b)  $f = g + (j + 2);$

- Para cada una de las instrucciones anteriores, muestre el código correspondiente en lenguaje ensamblador de MIPS, empleando el mínimo número de instrucciones y suponiendo que las variables f, g, i y j se encuentran, respectivamente, en los registros \$s0, \$s1, \$s2 y \$s3.
- ¿Cuántas instrucciones de MIPS son necesarias para ejecutar cada una de las instrucciones anteriores de lenguaje C?
- Si las variables f, g, i y j tienen los valores 1, 2, 3 y 4, respectivamente: ¿Cuál será el valor final de f?

**2.8.** Sean las siguientes instrucciones de lenguaje ensamblador de MIPS

a) `addi $s0, $s0, 7`

b) `sub $s0, $zero, $s0`  
`addi $s0, $s0, 1`

Supongamos que el registro \$s0 corresponde a la variable f que es entera y de 32 bits:

- ¿Cuál es la instrucción del lenguaje C correspondiente para cada caso?
- Si la variable f contiene inicialmente el valor 3: ¿Cuál es el valor final de f?

**2.9.** Sean las siguientes instrucciones del lenguaje C en que todas las variables son enteras de 32 bits:

a)  $f = f + g + h + i + j + 2;$

b)  $f = g - (f + 5);$

- Para cada una de las instrucciones anteriores, muestre el código correspondiente en lenguaje ensamblador de MIPS, empleando el mínimo número de instrucciones y suponiendo que las variables f, g, h, i y j se encuentran, respectivamente, en los registros \$s0, \$s1, \$s2, \$s3 y \$s4.
- ¿Cuántas instrucciones de MIPS son necesarias para ejecutar cada una de las instrucciones anteriores de lenguaje C?
- Si las variables f, g, h, i y j tienen los valores 1, 2, 3, 4 y 5 respectivamente ¿Cuál será el valor final de f?

**2.10.** Sean las siguientes instrucciones de lenguaje ensamblador de MIPS:

a) `sub $s1, $zero, $s1`  
`add $s0, $s1, $s2`

b) `addi $s2, $s0, 1`  
`sub $s0, $s1, $s2`

Supongamos que los registros \$s0, \$s1 y \$s2 corresponden respectivamente a las variables f, g y h:

- ¿Cuál es la instrucción del lenguaje C correspondiente para cada caso?
- Si las variables f, g y h tienen respectivamente los valores 1, 2 y 3: ¿Cuál es el valor final de f?

**2.11.** Para los números siguientes, que están escritos en binario:

a) 1010 1101 0001 0000 0000 0000 0000 0010    b) 1111 1111 1111 1111 1011 0011 0101 0011

- I) Escriba, en decimal, los números que están representados en cada caso si suponemos que el formato utilizado es el complemento a 2.
- II) Escriba, en decimal, los números representados en cada caso suponiendo que son enteros sin signo.
- III) Escriba, en decimal, los números que están representados en cada caso si suponemos que se emplea la representación signo-magnitud.
- IV) Escriba los números anteriores en hexadecimal.

**2.12.** Para los números siguientes, que están escritos en decimal:

a) 2147483647    b) 1000

- I) Escribalos en binario con la representación complemento a 2 con 32 bits.
- II) Escriba, en hexadecimal, cada uno de los números en complemento a 2 con 32 bits.
- III) Escriba, en hexadecimal, el opuesto de cada uno de los números en complemento a 2 con 32 bits.

**2.13.** Un registro en un procesador de 32 bits contiene cada uno de los números siguientes (2 casos):

a) 0xfffff7a    b) 0xffffffa

Si el contenido de ese registro se interpreta como un número entero codificado en complemento a 2:

- I) ¿A qué número decimal representa?
- II) ¿Cabría ese número en un registro de 16 bits? ¿Y en uno de 8 bits?

**2.14.** Sean los siguientes patrones de bits escritos en hexadecimal:

a) 0xABCDEF12    b) 0x1EAD030D

- I) Escriba en decimal los números que representan interpretando los patrones en binario natural.
- II) Muestre cómo esos datos pueden ser alojados en memoria tanto en un esquema *little-endian* como en uno *big-endian*. Suponga que los datos se almacenan en memoria a partir de la dirección 0.

**2.15.** Sean los siguientes patrones de bits escritos en binario:

a) 0000 0010 0001 0000 1000 0000 0010 0000    b) 1000 1101 0000 1000 0000 0000 0100 0000

- I) ¿Qué instrucciones MIPS representan?
- II) Indique el tipo de formato de cada instrucción (R o I)
- III) Escriba en hexadecimal dichos patrones de bits.
- IV) Suponiendo que esos patrones de bits representan números enteros codificados en complemento a 2, escriba los números representados en decimal.

**2.16.** Para las siguientes instrucciones:

a) add \$t0, \$s0, \$zero    b) sw \$t1, 32(\$t2)

- I) Muestre la representación interna de cada una de las instrucciones en hexadecimal.
- II) ¿Cuál es el formato de cada instrucción (R o I)?
- III) Para cada instrucción muestre, en hexadecimal, el valor de los campos *op*, *rs* y *rt*. Si es de tipo I, muestre el valor del campo *constante/desplazamiento* y si es de tipo R, el valor de los campos *rd* y *funct*.

**2.17.** Sean los siguientes valores escritos en hexadecimal

a) 0xAE0BFFFC    b) 0x8D08FFC0

- I) Escriba los números anteriores en binario natural con 32 bits.
- II) Escriba los números anteriores en decimal interpretando las cadenas de bits anteriores en binario natural.
- III) ¿Qué instrucciones MIPS representan los números anteriores?

**2.18.** Sean los siguientes valores escritos en hexadecimal:

a) 0x01098020    b) 0x8D090012

- I) Escriba los números anteriores en decimal interpretándolos en binario natural.
- II) ¿Qué instrucciones MIPS representan los números anteriores?
- III) ¿A qué tipo de formato (R o I) corresponde cada una de las instrucciones?
- IV) ¿Cuál es el valor de los campos *op* y *rt*?

**2.19.** Las siguientes modificaciones se podrían hacer en la arquitectura MIPS:

a) 8 registros

b) El campo *constante/desplazamiento* con 10 bits

- I) Para cada uno de los cambios, ¿Cuál sería el tamaño de cada uno de los campos de una instrucción con formato de tipo R?
- II) ¿Cuál sería el tamaño total, en bits, de una instrucción con formato R?
- III) Para cada uno de los cambios, ¿Cuál sería el tamaño de cada uno de los campos de una instrucción con formato de tipo I?
- IV) ¿Cuál sería el tamaño total, en bits, de una instrucción con formato I?
- V) ¿Por qué podrían esos cambios hacer decrecer el tamaño de un programa en lenguaje máquina MIPS?
- VI) ¿Por qué podrían esos cambios hacer crecer el tamaño de un programa en lenguaje máquina MIPS?

**2.20.** Traduzca la siguiente instrucción de código C a ensamblador de MIPS. Supóngase que las variables *i* y *j* se asignan, respectivamente, a los registros *\$s0* y *\$s1*. Supóngase también que las direcciones de los vectores *A* y *B* se encuentran respectivamente en los registros *\$s6* y *\$s7* y que las componentes de estos vectores son enteros de 32 bits.

$B[8] = A[i] + B[j];$

**2.21.** Sean las siguientes instrucciones del lenguaje C en que todas las variables son enteras de 32 bits:

a)  $f = g + h + B[g];$

b)  $f = g - A[B[4]];$

- I) Para cada una de las instrucciones anteriores, muestre el código correspondiente en lenguaje ensamblador de MIPS, empleando el mínimo número de instrucciones y suponiendo que las variables *f*, *g* y *h* se encuentran, respectivamente, en los registros *\$s0*, *\$s1* y *\$s2* y que las direcciones iniciales de los vectores *A* y *B* se encuentran respectivamente en los registros *\$s6* y *\$s7*.
- II) ¿Cuántas instrucciones de MIPS son necesarias para ejecutar cada una de las instrucciones anteriores de lenguaje C?
- III) Para cada una de las instrucciones anteriores de lenguaje C: ¿Cuántos registros diferentes son necesarios para ejecutar la instrucción?

**2.22.** Supongamos las siguientes instrucciones de lenguaje C:

a)  $B[5] = A[f - g];$

b)  $f = A[B[g] + 1];$

- I) Escriba el código ensamblador MIPS necesario para implementar cada una de las instrucciones C anteriores. Se puede suponer que las variables *f* y *g* corresponden respectivamente a los registros *\$s0* y *\$s1* y que las direcciones iniciales de los vectores *A* y *B* se encuentran en los registros *\$s6* y *\$s7* respectivamente.
- II) ¿Cuántas instrucciones de MIPS son necesarias para ejecutar cada una de las instrucciones C anteriores?
- III) ¿Cuántos registros MIPS son necesarios como mínimo para ejecutar cada una de las instrucciones de lenguaje C anteriores sin modificar las variables?

**2.23.** Sea el siguiente fragmento de programa en lenguaje ensamblador MIPS: ¿cuál es la instrucción correspondiente en lenguaje C?. Supóngase que las variables *f* y *g* están asignadas respectivamente a los registros *\$s0* y *\$s1* y que los registros *\$s6* y *\$s7* contienen respectivamente las direcciones de dos vectores llamados *A* y *B*.

```
sll  $t0, $s0, 2
add  $t0, $s6, $t0
sll  $t1, $s1, 2
add  $t1, $s7, $t1
lw   $s0, 0($t0)
addi $t2, $t0, 4
lw   $t0, 0($t2)
add  $t0, $t0, $s0
sw   $t0, 0($t1)
```

**2.24.** Para cada una de las siguientes situaciones:

a) \$t0 = 0x55555555  
\$t1 = 0x12345678

b) \$t0 = 0xBEADFEED  
\$t1 = 0xDEADEADE

¿Cuál será el valor final de \$t2 después de cada una de las siguientes secuencias de instrucciones?

- I) sll \$t2, \$t0, 4  
or \$t2, \$t2, \$t1
- II) sll \$t2, \$t0, 4  
andi \$t2, \$t2, 0xFFFF
- III) srl \$t2, \$t0, 3  
andi \$t2, \$t2, 0xFFEF

**2.25.** Sean las siguientes secuencias de instrucciones:

a) sll \$t2, \$t0, 1  
or \$t2, \$t2, \$t1

b) srl \$t2, \$t0, 1  
andi \$t2, \$t2, 0x00F0

- I) Suponga que \$t0 = 0x0000A5A5 y \$t1 = 0x00005A5A ¿Cuál será valor final de \$t2 después de cada una de las secuencias de instrucciones?
- II) Suponga que \$t0 = 0xA5A50000 y \$t1 = 0xA5A50000 ¿Cuál será valor final de \$t2 después de cada una de las secuencias de instrucciones?
- III) Suponga que \$t0 = 0xA5A5FFFF y \$t1 = 0xA5A5FFFF ¿Cuál será valor final de \$t2 después de cada una de las secuencias de instrucciones?

**2.26.** Cuando en un procesador MIPS se ejecuta la secuencia de instrucciones siguiente, los registros \$t0, \$s0 y \$s1 contienen respectivamente los valores 0xffff00ff, 0x412c450a, 0x00000030. ¿Cuál es el contenido final del registro \$s0 expresado en hexadecimal?

```
and $s0, $t0, $s0
sll $s1, $s1, 8
or $s0, $s0, $s1
```

**2.27.** Sean las siguientes cadenas de caracteres:

a) "A byte"

b) "computer"

- I) Escriba los valores decimales correspondientes a los códigos ASCII de los caracteres de cada cadena.
- II) Escriba los valores hexadecimales correspondientes a los códigos Unicode UTF-16 de los caracteres de cada una de esas cadenas, utilizando el conjunto de caracteres Latino Básico.

**2.28.** Sean las siguientes secuencias de números hexadecimales

a) 61, 64, 64

b) 73, 68, 69, 66, 74

Escriba la cadena resultante de interpretar los números de cada una de esas secuencias como caracteres del código ASCII.

**2.29.** A partir de la dirección de memoria 0x00403580 están almacenados cuatro bytes con los valores 65, 89, 0 y 0 (decimal):

- I) Si interpretamos el contenido de esa zona de memoria como una cadena de caracteres codificados en ASCII: ¿Cuál es la cadena almacenada?
- II) Suponiendo que se emplea el criterio *little-endian*, si interpretamos los citados datos como un entero representado en complemento a 2: ¿Cuál es el valor de ese entero en decimal?

**2.30.** Los registros \$t0, \$t1 y \$s0 de un computador con arquitectura MIPS contienen respectivamente 0x00008a00, 0x3532340a y 0x000cb848. En su memoria se encuentran almacenados los datos siguientes en las direcciones indicadas:

Dirección (hex.)	8a00	8a01	8a02	8a03	8a04	8a05	8a06	8a07	8a08	8a09	8a0a	8a0b
Contenido (hex.)	a1	eb	6f	41	42	d0	14	44	a0	1b	f1	a7

- Suponiendo que se ejecuta la instrucción `lw $s0, 8($t0)`, exprese en hexadecimal el contenido final del registro \$s0 después de la ejecución de la citada instrucción si se sigue el convenio *little-endian*.
- Exprese en hexadecimal la dirección de la posición de memoria que cambia después de ejecutarse la instrucción `sb $t1, 4($t0)`
- ¿Cuál será el contenido final de la posición de memoria afectada después de la ejecución de la instrucción `sb` mencionada en el apartado anterior? Exprese el resultado en hexadecimal.

**2.31.** En la memoria de un computador MIPS se encuentra la instrucción `sw $s0, 40($s5)`. Supóngase que cuando se ejecuta esa instrucción los registros \$s0 y \$s5 contienen respectivamente 0x0a12b3c4 y 0x00db040c.

- Codifique esa instrucción en lenguaje máquina expresando el resultado en hexadecimal.
- ¿Cuál es la dirección de la posición de memoria modificada por la citada instrucción expresada en hexadecimal?
- Escriba en hexadecimal el valor del **byte** contenido en esa dirección de memoria si se sigue el convenio *big-endian*

**2.32.** Suponga que el registro \$t1 contiene la dirección 0x10000000, que el registro \$t2 contiene la dirección 0x10000010 y que se ejecutan cada una de las siguientes secuencias de instrucciones:

- |                                  |                                  |
|----------------------------------|----------------------------------|
| a) <code>lb \$t0, 0(\$t1)</code> | b) <code>lb \$t0, 0(\$t1)</code> |
| <code>sw \$t0, 0(\$t2)</code>    | <code>sb \$t0, 0(\$t2)</code>    |

- Suponga que los datos almacenados a partir de la dirección 0x10000000 son (en hex.): 12, 34, 56 y 78. ¿Cuál será el valor almacenado en la dirección de memoria apuntada por el registro \$t2? Se puede suponer que la citada posición de memoria apuntada por \$t2 está inicializada a 0xFFFFFFFF. También se puede suponer que se utiliza el criterio *big-endian*
- Suponga que los datos almacenados a partir de la dirección 0x10000000 son (en hex.): 80, 80, 80 y 80. ¿Cuál será el valor almacenado en la dirección de memoria apuntada por el registro \$t2? Se puede suponer que la citada posición de memoria apuntada por \$t2 está inicializada a 0x00000000. También se puede suponer que se utiliza el criterio *little-endian*
- Suponga que los datos almacenados a partir de la dirección 0x10000000 son (en hex.): 11, 00, 00 y FF. ¿Cuál será el valor almacenado en la dirección de memoria apuntada por el registro \$t2? Se puede suponer que la citada posición de memoria apuntada por \$t2 está inicializada a 0x55555555 y que se utiliza el criterio *big-endian*.

**2.33.** Sean las siguientes secuencias de instrucciones de ensamblador MIPS:

- |                                     |  |
|-------------------------------------|--|
| a) <code>lui \$t0, 0x1234</code>    | b) <code>ori \$t0, \$t0, 0x5678</code> |
| <code>ori \$t0, \$t0, 0x5678</code> | <code>lui \$t0, 0x1234</code>          |

- ¿Cuál es el valor final del registro \$t0 después de cada una de las secuencias de instrucciones?
- Escriba el código C equivalente a cada una de esas secuencias de instrucciones.

**2.34.** Cuando se ejecuta cada una de las secuencias de instrucciones siguientes, los registros \$s0 y \$s1 contienen, respectivamente, 0x00000037 y 0x34333231 ¿Cuál es el contenido final del registro \$s1 expresado en hexadecimal?

- |                                     |                                     |
|-------------------------------------|-------------------------------------|
| a) <code>lui \$t1, 0xffff</code>    | b) <code>lui \$t1, 0xffff</code>    |
| <code>ori \$t1, \$t1, 0x00ff</code> | <code>ori \$t1, \$t1, 0xff00</code> |
| <code>and \$s1, \$s1, \$t1</code>   | <code>and \$s0, \$s0, \$t1</code>   |
| <code>sll \$t0, \$s0, 8</code>      | <code>or \$s0, \$s0, \$t0</code>    |
| <code>or \$s1, \$s1, \$t0</code>    |                                     |

- 2.35.** Suponga que la instrucción `bne $t1, $s0, bucle` está almacenada en la dirección `0x002010a4`. Si la etiqueta `bucle` representa a la dirección `0x0020100c`, codifique la instrucción en lenguaje máquina.
- 2.36.** La ejecución de un programa llega a una instrucción codificada como `0x1620ff24`, que representa a la instrucción `bne $s1, $zero, Y`. Esta instrucción provoca la bifurcación del programa a la instrucción situada en la dirección `0x203042fc`. ¿Cuál es la dirección donde se encuentra la citada instrucción `bne`? Exprese el resultado en hexadecimal.
- 2.37.** Sean los siguientes valores escritos en binario:
- a) `0000 0001 0001 0000 0000 0000 0000 0100`    b) `1111 1111 1111 1111 1111 1111 1111 1100`
- I) Escriba el código MIPS que almacena cada una de esas constantes en el registro `$t1`.
- II) Si el valor actual del contador de programa (PC) fuera `0x00000400`, ¿Podría usarse una sola instrucción de salto (*jump*) para transferir el control a alguna de esas direcciones?
- III) Si el valor actual del contador de programa (PC) fuera `0x00400600`, ¿Podría usarse una sola instrucción de bifurcación para transferir el control a alguna de esas direcciones?
- IV) Si el campo constante/desplazamiento de las instrucciones MIPS tuviera solo 8 bits, escriba el código en ensamblador de MIPS para cargar cada una de las constantes anteriores en el registro `$t1`. Evite el uso de la instrucción `lui`.
- 2.38.** Las siguientes modificaciones se podrían hacer en la arquitectura MIPS. Supóngase que, con estos cambios, la longitud de instrucción continuara siendo de 32 bits y que los cambios solo afectan a los campos de desplazamiento o dirección:
- a) 8 registros    b) El campo constante/desplazamiento con 10 bits
- I) ¿Cuál es el impacto de cada uno de estos cambios en el alcance de las instrucciones de bifurcación?
- II) ¿Cuál es el impacto de cada uno de estos cambios en el alcance de las instrucciones `j` (*jump*)?
- III) ¿Cuál es el impacto de cada uno de estos cambios en el alcance de las instrucciones `j r` (*jump register*)?
- 2.39.** Sean los siguientes modos de direccionamiento:
- a) Direccionamiento por registro    b) Direccionamiento relativo al PC
- I) Ponga ejemplos de instrucciones MIPS que utilicen cada uno de esos modos de direccionamiento.
- II) ¿Cuál es el tipo de formato de cada una de las instrucciones citadas en el apartado anterior?
- III) Describa las ventajas e inconvenientes de cada uno de esos modos de direccionamiento.
- 2.40.** Sean las siguientes secuencias de instrucciones:
- a) `0x00000000 lui $s0, 100`  
`0x00000004 ori $s0, $s0, 40`
- b) `0x00000100 addi $t0, $zero, 0`  
`0x00000104 lw $t1, 0x4000($t0)`
- I) Escriba la representación interna de cada una de las instrucciones en hexadecimal.
- II) Supongamos que en las instrucciones de tipo I, se reduce el campo constante/desplazamiento a 8 bits. Reescriba cada una de las secuencias de instrucciones evitando el uso de la instrucción `lui`.
- III) ¿Cuántas instrucciones adicionales necesita el código del apartado anterior en cada caso?
- 2.41.** Sean los siguientes valores de los campos de diferentes instrucciones MIPS.
- a)  $op = 0, rs = 3, rt = 2, rd = 3, shamt = 0, funct = 34$  (decimal)
- b)  $op = 0x23, rs = 1, rt = 2, constante/desplazamiento = 0x4$
- I) ¿A qué tipo de formato (R o I) corresponde cada una de las instrucciones?
- II) Escriba cada una de las instrucciones en lenguaje ensamblador de MIPS.
- III) Escriba en hexadecimal el código máquina MIPS de cada instrucción.