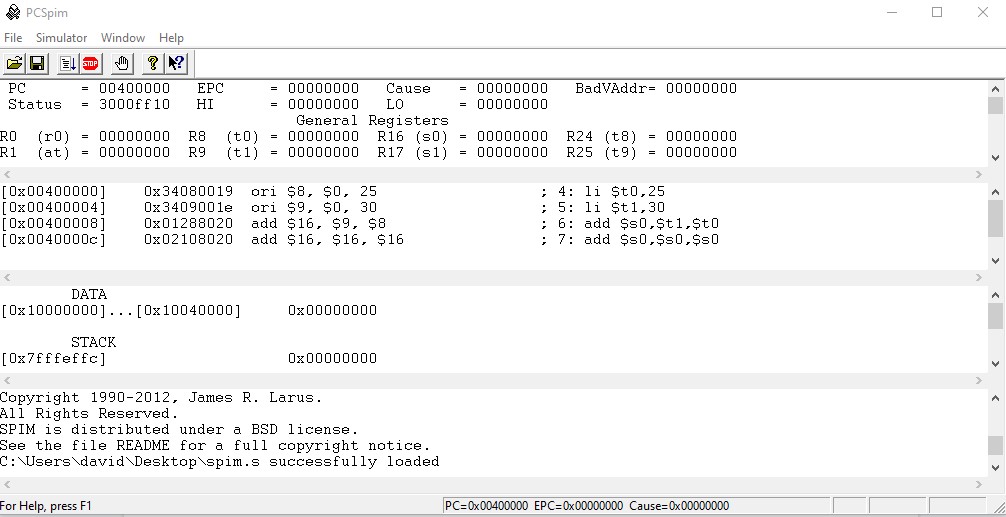
**Práctica 1**

**Ejercicio 1**

* ¿Cuántas instrucciones máquina comprende el programa? 4.



• ¿En qué instrucciones máquina se traducen las pseudoinstrucciones presentes?

• ¿En qué dirección de memoria se encuentra la instrucción add $s0,$s0,$s0? 0x0040000C.

• ¿Qué instrucción del programa se codifica como 0x01288020? add $s0,$t1,$t0.

• Diga en hexadecimal el valor del perímetro calculado por el programa. 0x0000006e.

**Ejercicio 2**

* ¿Cuántos bytes de la memoria principal están ocupados por las variables del

programa? 3 palabras \* 4 bytes = 12.

• ¿Cuántas instrucciones de acceso a la memoria contiene el programa? Solo las lw, es decir, 2.

• ¿En qué dirección se escribe el valor del perímetro? 0x00400024.

• ¿Por qué la pseudoinstrucción la $t0, A se traduce en solo una instrucción máquina y la $t1, B lo hace en dos? la $t0, A solo tiene datos en la parte alta y, por tanto, se traduce en únicamente como instrucción lui. Por otra parte, $t1, B tiene datos en la parte y baja, por lo que se traduce como una combinación entre ori y lui.

• Justificar el valor (4) que aparece en la directiva .space 4? 1 space te deja 8 bits = 1 byte y, por tanto, te deja 32 bits, es decir, 4 bytes para escribir datos.

• Afectaría al valor final de P si en lugar de la directiva .space 4 hiciésemos uso de la directiva .word 0? No afecta.

• ¿Qué valor contiene el registro $t1 cuando se ejecuta la instrucción lw $s1,0($t1)? 0x10000000.

**Ejercicio 3**

* ¿Cuál es la codificación de la instrucción máquina syscall? 0x0000000c.

• ¿En qué instrucción máquina se ha traducido la pseudoinstrucción move $a0,$a1? addu $4, $0, $18.

• Sustituye ahora la instrucción sw $s2,0($t2) por sw $s2,2($t2). ¿Qué ocurre

cuando se intenta ejecutar el programa? Razone la respuesta. No se puede acceder a la dirección de memoria donde apunta.

**Ejercicio 4**

1. **¿Cuál de estas instrucciones es una traducción incorrecta de la pseudoinstrucción de movimiento de datos move $t0, $t1 (copia el contenido de $t1 en $t0)?**

and $t0, $t1, $zero

2. ¿Cuáles de las siguientes instrucciones son buenas traducciones de la pseudoinstrucción li $t0,100 (almacena el valor decimal 100 en $t0)?

* ori $t0, $zero, 0x64
* addi $t0, $zero, 0x64
* ori $t0, $zero, 100
* addi $t0, 0x64, $zero
* xori $t0, $zero, 100
* addi $t0, $zero, 100

3. El código siguiente origina un error durante su ejecución. ¿Dónde y por qué se produce?

**li $t0, 0x10003000**

**lw $t1, 2($t0)**

NO se puede cargar una dirección de memoria en un inmediato.

4. Aunque la letra ele ("l") de las instrucciones lui y lw (y también lh y lb) significa load en inglés, ¿qué diferencia fundamental hay entre estas dos instrucciones? La diferencia fundamental entre las instrucciones es que lui solo carga la parte alta y lw carga la palabra entera.

5. Suponga que tenemos una variable N declarada de la siguiente manera:

**.data 0x10000000**

**N: .space 4**

Indica la instrucción o instrucciones máquina adecuadas para asignar los siguientes

valores:

• N = 0 → li.

• N = -1 → li.

• N = 0x100000 → la u ori y lui.

• N = 0x100040 → la u ori y lui.

• N = 200000 (en decimal) → ori y lui (más optimizado, con un lui).

6. ¿Cómo traducirías a instrucciones máquina la pseudoinstrucción li $t0, -1?

lui $1, -1

ori $8, $1, -1

7. Supón que el segmento de datos de un programa en ensamblador MIPS contiene la siguiente descripción:

.data 0x10000000

X: .space 4

En cada uno de los siguientes casos, indica qué valor escriben las instrucciones sobre la

posición de memoria X:

la $t0,X

sw $zero,0($t0)

9

la $t0,X

sh $zero,0($t0)

sh $zero,2($t0)

la $t0,X

sb $zero,0($t0)

sb $zero,1($t0)

sb $zero,2($t0)

sb $zero,3($t0)

la $t0, X

lui $t1, 0x0001

sw $t1,0($t0)

la $t0,X

lui $t1,0xFFFF

ori $t1,$t1,0xFFFF

sw $t1,0($t0)

lui $t0,0x1000

andi $t1,$t1,0x0000

sw $t1,0($t0)

la $t0,X

lw $t1,0($t0)

sw $t1,0($t0)

li $t1,50

sw $t1,X

li $t0,0x50

sw $t0,X

li $t0,0x10000000

li $t1,0xFFFFFFFF

sw $t1,0($t0)