Prácticas 1, 2 y 3 de *Estructura de los Computadores*

Eduardo Espuch

Universitat d'Alacant
Universidad de Alicante

Resumen

Documento que reune los entregables de las prácticas 1, 2 y 3 de la asignatura *Estructura de los Computadores*.

Índice

1. Ejercicios práctica 1

1.1. Ejercicio 1

Escribe el código que haga las siguientes acciones utilizando el convenio de registros y utilizando la instrucción addi:

$$\$12 = 5 \qquad \$10 = 8 \qquad \$13 = \$12 + 10$$

$$\$10 = \$10 - 4 \quad \$14 = \$13–30 \qquad \$15 = \$10$$

Ensamblad y ejecutad el programa y comprobad que el resultado final es $\$t7 = \$t2 = 4$, $\$t6 = -15$, $\$t4 = 5$, $\$t5 = 15$.

En el archivo **p1ej1.asm** se observa que el resultado es correcto.

1.2. Ejercicio 2

¿Se podría escribir el mismo código utilizando la instrucción addiu? Haz la prueba.

En el archivo **p1ej2.asm** se modifica el anterior usando addiu y se observa que se obtiene el mismo resultado, esto es porque ninguna operación anterior producía OVERFLOW por lo tanto no era necesario realizar la función sin signo.

1.3. Ejercicio 3

¿Cuál es el código de operación de la instrucción addiu?

Tomando el código en hexadecimal de cualquier instrucción con addiu, vemos que los dos primeros valores son $0x24\ldots$ o $0x25\ldots$ (entre otros), esta diferencia no nos interesa, únicamente los 6 primeros bits que seria 0010 01(00 o 01 pero no tienen relevancia), es decir, que la codificación de addiu seria $0010\ 01_2$

1.4. Ejercicio 4

Codifica en binario la instrucción addiu(función) $v0(rt), $zero(rs), 1(K).

La codificación sería $0010\ 0100\ 0000\ 0010\ 0000\ 0000\ 0000\ 0001_2$

2. Ejercicios práctica 2

2.1. Ejercicio 1

Modifica el código del último ejercicio del apartado 4 para que aparezca en la pantalla el contenido del registro $t2=0x0000CAFE$_{16}$.

Dando a $a0 el valor de $t2 sabemos que será este el valor que se imprima y si usamos la función correspondiente a $v0=34, al realizar un syscall se imprimirá en hexadecimal. El ejercicio adjunto es **p2ej1.asm**.

Se pide en el apartado inicial que $t1 se meta en el registro manualmente al iniciar pero he decidido asignarle el valor al comienzo de la ejecución, considerando el decimal de $0x0000FACE_{16}=64206_{10}$.

Para hacer la conversión de un hexadecimal a otro, hemos trabajado con ellos en binario, observando que usando la puerta XOR junto al hexadecimal $0x0003030_{16}$ se obtiene el resultado deseado.

2.2. Ejercicio 2

Escribe el código que lee un valor entero por teclado y escribe el mismo valor en binario por la consola.

Usando las funciones $v0=5 (lee un entero) y $v0=35 (imprime un binario) se realiza fácilmente este ejercicio, considerando que imprimirá el valor de $a0. El ejercicio adjunto es **p2ej2.asm**

3. Ejercicios práctica 3

3.1. Ejercicio 1

Escribe un programa que lea del teclado una letra en mayúscula y la escríba en minúscula en la consola.

En el archivo **p3ej1.asm** se observa la solución. Podemos observar que entre los caracteres en mayúsculas y en minúsculas, existe una diferencia de 32_{10}, con lo cual si al valor numérico correspondiente a un carácter en ASCII le sumamos 32 en decimal, obtenemos su equivalente en minúsculas. Se ha añadido un salto de linea por comodidad visual.

Habría que utilizar las funciones $v0=12 y $v0=11 para leer e imprimir un carácter.

3.2. Ejercicio 2

Itera el código que acabas de escribir.

Usando etiquetas y la función j podemos realizar una iteración. Donde pongamos la función j sera el final y la dirección a continuación sera el inicio, de ahí que usemos una etiqueta en la que se guardara la dirección de inicio.

El archivo adjunto es **p3ej2.asm** y contiene también saltos de linea para una mayor comodidad visual.

3.3. Ejercicio 3

Convierte caracteres numéricos. Escribe el código que lea del teclado un carácter numérico (del '0' al '9') y lo convierta en un valor numérico (del 0 al 9) y lo escriba por pantalla. Itera el código.

Partiendo del ejercicio anterior, sabiendo que el valor numérico decimal en ASCII del '0' es 48, sabremos que los valores entre '0' y '9' se encuentran en el intervalo [48,57]. Es facil ver que, si restamos 48 al valor numerico del caracter, obtendremos el entero decimal correspondiente.

Habría que utilizar las funciones $v0=12 y $v0=1 para leer un carácter e imprimir un entero, el archivo adjunto es **p3ej3.asm**.