Alejandro Martínez Cedillo – 74393989Y

***Práctica 2 Estructura de los Computadores***

*a) Haz un código que lea un valor x del teclado y escriba x+1 en la consola*

*.text 0x00400000*

*addi $v0, $0, 5*

*syscall*

*addi $a0, $v0, +1*

*addi $v0, $0, 1*

*syscall*

*addi $v0, $0, 10*

*syscall*

*b) Haz un código que lea un valor x de teclado y escriba x-1 en la consola*

*.text 0x00400000*

*addi $v0, $0, 5*

*syscall*

*addi $a0, $v0, -1*

*addi $v0, $0, 1*

*syscall*

*addi $v0, $0, 10*

*syscall*

*c) ¿Cómo se escribe la instrucción que hace* ***$t2 = $t1 + $t0****?*

Se escribe de la siguiente manera

**add $t2, $t1, $t0**

**.text 0x00400000**

**addiu $t0, $zero, 25**

**addiu $t1, $zero, 5**

**sub $t2,$t0,$t1**

**addi $v0, $zero, 10**

**syscall**

*d) En el anterior código ¿ qué hace cada línea?*

La primera instrucción suma 25 al registro $t0,

La segunda le suma 5 al $t1,

La siguiente introduce el contenido de $t1 con el signo cambiado y lo guarda en $t2

Por último ejecuta la función nº10 y con un syscall apaga el programa.

*e) ¿En qué dirección de memoria se almacena la instrucción* ***sub****?*

La instrucción sub se almacena en el registro **$t2**

*f) ¿Cuál es el valor final del registro* ***$t2?***

El valor final de la resta y el almacenado en **$t2** es 20

*g) Observa el código anterior. ¿Cómo se codifica la primera instrucción? Hacedlo a mano*

Si pasamos de notación Hexadecimal a binario se convierte en **0000 1001,** el código de operación al ser de 6 bits sería entonces **0000 10**.

*h) Confirmad con el programa ensamblado que el código máquina es el mismo*

*i) ¿Cómo se codifica la última instrucción de resta del código de partida que acabamos de escribir?*

*j) Confirmad con el programa ensamblado que el código máquina es el mismo*

*k) Notad que hay 64 instrucciones distintas con formato R ¿Por qué?*

*l) Escribe el código que hace estas acciones haciendo uso de las instrucciones estudiadas*

*m) Ensambla y ejecútalo, comprueba que el contenido de los registros es correcto*

*n) ¿Es necesario que forme parte del repertorio de instrucciones la instrucción* ***subi****?*

*o) Haz el código que lea dos números x e y, y obtén por pantalla el valor de la suma x + y*

*p) Haz el código que lee dos número x e y, y obtén por pantalla x - y*

*q)* *¿Podríamos utilizar la instrucción lógica ori para dar un valor inicial a un registro en lugar de la instrucción addi?*

*r)* *¿Cómo escribirías la instrucción que hace $t2=7 utilizando ori?*

*s)* *¿Sería o no ventajoso en cuanto al coste temporal el hecho de utilizar el operador lógico ori en vez de la instrucción addi?. Razónalo.*

*t)* *Reescribid el código de partida Aritmética de enteros del apartado 2 cambiando addi o addiu por ori:*

*u) ¿Cuál es el valor del código de operación de la instrucción ori?*

*v) Escribe el código que haga la operación lógica OR de $t1 y $t2 y lo guarde en $t3, la operación lógica AND de $t1 y $t2 y lo guarde en $t4, y la operación lógica XOR de $t1 y $t2 y lo guarde en $t5. Escribe en la ventana de registros, tras ensamblarlo, los siguientes valores para los registros $t1=0x55555555 y $t2= 0xAAAAAAA. Ejecuta el código y estudia los resultados.*

*w) Supón que $t1=0x0000FACE, utilizando únicamente las instrucciones lógicas de la tabla anterior, escribe el código que reordene los bits de $t1 de manera que en $t2 aparezca el valor 0x0000CAFE. Ensambla y escribe en la ventana de registros $t1=0x0000FACE. Ejecuta y comprueba que el código es correcto.*

*x)Modifica el código del último ejercicio para que aparezca en la pantalla el contenido del registro $t2 = 0000CAFE*

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