Estructura de los computadores

Prácticas 1 a 3

Práctica 1:

Actividad 5:

Volved a escribir el programa cambiando addi o addiu y dad como valor inicial de $t0 el positivo más grande posible ($t0 = 0x7FFFFFFF) y ejecutadlo observando el contenido de $t1 en hexadecimal y en decimal. ¿Qué ha ocurrido?
Si el programador considera que está operando con número naturales, el resultado que hay en $t1 sería correcto? ¿Cuál sería su valor en decimal?

(/pr1/ac5.asm)

```
.text 0x00400000
addiu $9,$8,25
addiu $10,$8,5
```

Name	Number	Value
$zero	0	0x00000000
$at	1	0x00000000
$v0	2	0x00000000
$v1	3	0x00000000
$a0	4	0x00000000
$a1	5	0x00000000
$a2	6	0x00000000
$a3	7	0x00000000
$t0	8	0x7fffffff
$t1	9	0x80000018
$t2	10	0x80000004
$t3	11	0x00000000
$t4	12	0x00000000
$t5	13	0x00000000
$t6	14	0x00000000
$t7	15	0x00000000
$s0	16	0x00000000
$s1	17	0x00000000
$s2	18	0x00000000
$s3	19	0x00000000
$s4	20	0x00000000
$s5	21	0x00000000
$s6	22	0x00000000
$s7	23	0x00000000
$t8	24	0x00000000
$t9	25	0x00000000
$k0	26	0x00000000
$k1	27	0x00000000
$gp	28	0x10008000
$sp	29	0x7fffeffc
$fp	30	0x00000000
$ra	31	0x00000000
pc		0x00400008
hi		0x00000000
lo		0x00000000

Name	Number	Value
$zero	0	0
$at	1	0
$v0	2	0
$v1	3	0
$a0	4	0
$a1	5	0
$a2	6	0
$a3	7	0
$t0	8	2147483647
$t1	9	-2147483624
$t2	10	-2147483644
$t3	11	0
$t4	12	0
$t5	13	0
$t6	14	0
$t7	15	0
$s0	16	0
$s1	17	0
$s2	18	0
$s3	19	0
$s4	20	0
$s5	21	0
$s6	22	0
$s7	23	0
$t8	24	0
$t9	25	0
$k0	26	0
$k1	27	0
$gp	28	268468224
$sp	29	2147479548
$fp	30	0
$ra	31	0
pc		4194312
hi		0
lo		0

El valor supera el rango de operación en complemento a 2, si se realiza la operación con addi, no se podría realizar al pasarse del rango, pero en este caso utilizamos addiu, (suma sin signo), es decir que los resultados obtenidos en hexadecimal son correctos si no tenemos en cuenta el complemento a 2, pero ya que se tienen en cuenta al pasarlos a decimal, los valores son incorrectos.

Cuestión 6:

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

$12=5 $10= 8 $13=$12 + 10 $10=$10 - 4 $14=$13 – 30 $15=$10

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

(/pr1/cuestion6.asm)

```
.text 0x00400000
addiu $t4,$zero,5
addiu $t2,$zero,8
addiu $t5,$t4,10
addiu $t2,$t2,-4
addiu $t6,$t5,-30
addiu $t7,$t2,0

addiu $v0,$zero,1
```

Práctica 2:

Cuestión 13:

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= 0xAAAAAAAA. Ejecuta el código y estudia los resultados.

(/pr2/c13.asm)

```
.text

or $t3,$t1,$t2
and $t4,$t1,$t2
xor $t5,$t1,$t2
```

Registers | Coproc 1 | Coproc 0

Name	Number	Value
$zero	0	0x00000000
$at	1	0x00000000
$v0	2	0x00000000
$v1	3	0x00000000
$a0	4	0x00000000
$a1	5	0x00000000
$a2	6	0x00000000
$a3	7	0x00000000
$t0	8	0x00000000
$t1	9	0x55555555
$t2	10	0xaaaaaaaa
$t3	11	0xffffffff
$t4	12	0x00000000
$t5	13	0xffffffff
$t6	14	0x00000000
$t7	15	0x00000000
$s0	16	0x00000000
$s1	17	0x00000000
$s2	18	0x00000000
$s3	19	0x00000000
$s4	20	0x00000000
$s5	21	0x00000000
$s6	22	0x00000000
$s7	23	0x00000000
$t8	24	0x00000000
$t9	25	0x00000000
$k0	26	0x00000000
$k1	27	0x00000000
$gp	28	0x10008000
$sp	29	0x7fffeffc
$fp	30	0x00000000
$ra	31	0x00000000
pc		0x0040000c
hi		0x00000000
lo		0x00000000

0x55555555 = 0101 0101 0101 0101 0101 0101 0101 0101

0xaaaaaaaa = 1010 1010 1010 1010 1010 1010 1010 1010

Los operadores or, xor, y and funcionan según las siguientes tablas, comparando bit por bit de los valores de $t1 y $t2:

OR:

A	B	S
0	0	0
0	1	1
1	0	1
1	1	1

Por lo tanto el resultado obtenido es 1111 1111 1111 1111 1111 1111 1111 1111 = 0xffffffff

XOR:

A	B	S
0	0	0
0	1	1
1	0	1
1	1	0

Por lo tanto el resultado obtenido es 1111 1111 1111 1111 1111 1111 1111 1111 = 0xffffffff

AND:

A	B	S
0	0	0
1	0	0
0	1	0
1	1	1

Por lo tanto el resultado obtenido es 0000 0000 0000 0000 0000 0000 0000 0000 = 0x00000000

Cuestión 14:

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.

Es posible obtener el resultado de una forma sencilla utilizando el operador XORI, teniendo en cuenta los valores binarios:

0x0000face= 1111 1010 1100 1110

0x0000cafe= 1100 1010 1111 1110

Los valores que se van a cambiar son los marcados, por lo que según la tabla de funcionamiento de XOR del ejercicio anterior obtenemos el siguiente resultado:

0x00003030= 0011 1010 0011 1110 = 12336

Este resultado lo pasamos a decimal y escribimos la siguiente operación:

(/pr2/c14.asm)

.text

xori $t2,$t1,12336

Name	Number	Value
$zero	0	0x00000000
$at	1	0x00000000
$v0	2	0x00000000
$v1	3	0x00000000
$a0	4	0x00000000
$a1	5	0x00000000
$a2	6	0x00000000
$a3	7	0x00000000
$t0	8	0x00000000
$t1	9	0x0000face
$t2	10	0x0000cafe
$t3	11	0x00000000
$t4	12	0x00000000
$t5	13	0x00000000
$t6	14	0x00000000
$t7	15	0x00000000
$s0	16	0x00000000
$s1	17	0x00000000
$s2	18	0x00000000
$s3	19	0x00000000
$s4	20	0x00000000
$s5	21	0x00000000
$s6	22	0x00000000
$s7	23	0x00000000
$t8	24	0x00000000
$t9	25	0x00000000
$k0	26	0x00000000
$k1	27	0x00000000
$gp	28	0x10008000
$sp	29	0x7fffeffc
$fp	30	0x00000000
$ra	31	0x00000000
pc		0x00400004
hi		0x00000000
lo		0x00000000

Cuestión 16:

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

(/pr2/c16.asm)

```
.text
addi $v0,$0,5        (Lectura de un numero entero)
syscall

addi $a0,$v0,0       (Pasamos ese número al registro $a0)
addi $v0,$0,35       (35 es el código para mostrar el valor en $a0 en binario por consola)
syscall

addi $v0,$0,10       (Terminar ejecución)
syscall
```

Práctica 3:

Cuestión 8:

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

(pr3/c8.asm)

```
.text
addi $v0,$zero,12    (leer carácter)
syscall

addi $t1,$v0,32      (sumar 32 al carácter para pasarlo a minúscula, guardarlo en t1 para poder hacer salto de línea primero)

addi $a0,$zero,'\n'  (Salto de línea para que sea más fácil de ver el resultado)
addi $v0,$zero,11
syscall

addi $a0,$t1,0       (Pasar el valor de t1 a a0 e imprimirlo)
addi $v0,$zero,11
syscall

addi $v0,$zero, 10   (Terminar ejecución)
syscall
```

Cuestión 10:

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.

(pr3/c10.asm)

```
.text
eti1: addi $v0,$zero,12        (leer carácter)
syscall

addi $t1, $v0,-48              (pasar el valor del carácter a t1 y restarle 48 para que el resultado dé el numero introducido)

addi $a0,$zero,'\n'            (salto de linea)
addi $v0,$zero,11
syscall

addi $a0, $t1,0                (pasar el valor guardado en t1 e imprimirlo como entero)
addi $v0,$zero,1
syscall

addi $a0,$zero,'\n'            (salto de linea)
addi $v0,$zero,11
syscall

j eti1                         (Salto a eti1)
```