

Práctica 7

La ruta de

datos

monociclo

Jordi Blasco Lozano

Arquitectura de computadores

Grado en Inteligencia Artificial

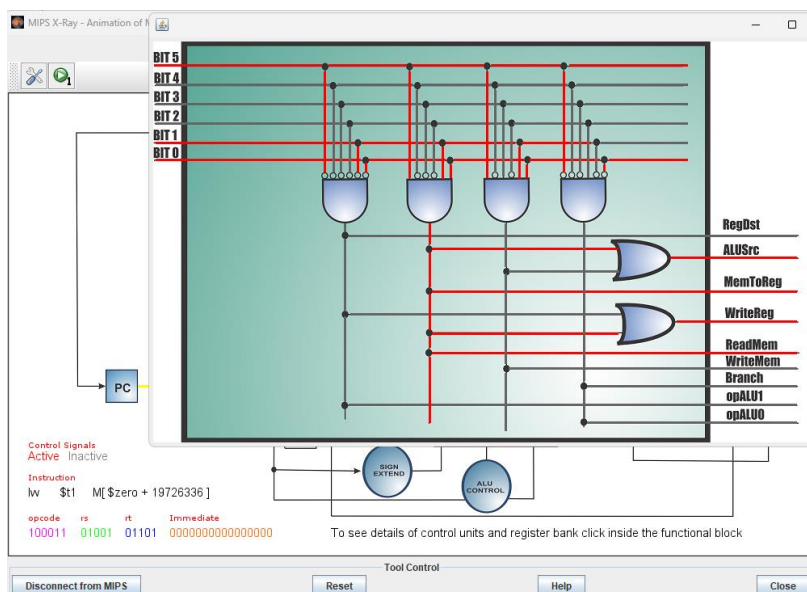
Indice:

Indice:	2
1. Cuestion 1	3
2. Actividad 2	5
3. Cuestion 2	5
4. Cuestion 3	6
5. Cuestion 4	7
6. Cuestion 5	7
7. Cuestion 6	8

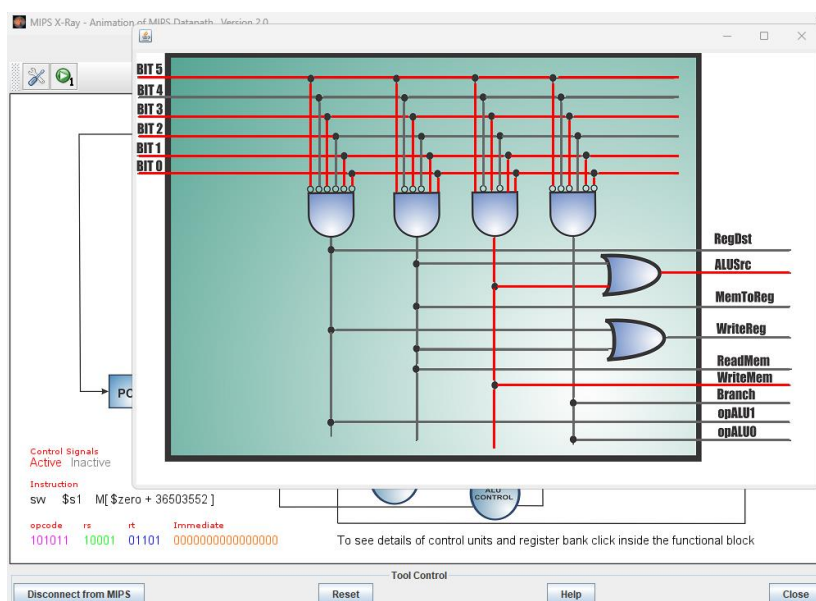
1. Cuestion 1

¿Qué señales de control se activan durante la ejecución de la instrucción lw y sw?

En lw se activaran las señales de control ALUSrc, MemToReg, WriteReg, ReadMem, por lo que lw activara la ALU para la entrada de datos, movera los datos de la memoria leyendola de alli a los registros escribiendolos en ellos.



En sw se activaran las señales de control ALUSrc y WriteMem, por lo que sw activara la ALU y escribira en la memoria.



¿Cuál es el valor de los bits de operación de la ALU (Op1, Op2 y Op3) para cada instrucción del programa? ¿Cuál es el valor de opALU1 y opALU0 para cada instrucción? ¿Por qué la señal opALU1 y opALU0 es la misma para las instrucciones lw y sw?

instruccion	Op 1	Op 2	Op 3	Op ALU1	Op ALU2
lui	1	1	0	0	1
ori	1	1	0	0	1
add	0	1	0	1	0
lw	0	1	0	0	0
addi	1	1	0	0	1
subi	1	1	0	0	1
sub	1	1	0	1	0
beq	0	1	0	0	0
sw	0	1	0	0	0
addiu	1	1	0	0	1

Las intruccion lw y sw usan de la misma forma la direcciones de memoria por lo que la carga y la escritura datos tendran la mismas señales opALU1 y opALU2

2. Actividad 2

Ejecuta el código de la actividad 2 en el simulador MARS. Si lo deseas puedes utilizar la herramienta MIPS X Ray para ver el recorrido realizado por las instrucciones en la ruta de datos.

Identifica el ciclo de reloj en el que se está ejecutando cada instrucción.

.data		ciclo:
num:		
	.word 7	
	.word 3, 8	
.text		
	addi \$t0, \$zero, 5	1
	addi \$t1, \$zero, 3	2
	la \$t4, num	3 (lui)
	la \$t4, num	4 (ori)
	lw \$t3, 4(\$t4)	5
	beq \$t3, \$t1, salto	6
	add \$t1, \$t1,\$t0	7
	j final	
salto:	add \$t1, \$t3, \$t1	8
final:	addi \$a0, \$t1, 0	9
	li \$v0, 1	10
	syscall	11
	li \$v0, 10	12
	syscall	

3. Cuestion 2

Identifica los valores que contendrán las etiquetas de la figura 2 cuando se ejecute la instrucción en el ciclo especificado. Rellena la tabla 1 con el valor de las etiquetas en formato decimal (dec), hexadecimal (hex) o en cualquiera de los dos, según se indique. Si la instrucción no atraviesa la etiqueta o no se conoce el valor, márcala como X

Núm	Etiqueta	Valor durante ciclo 5	Valor durante ciclo 6	Valor durante ciclo 7
1	a (en hex)	0x 00400010	0x 00400014	0x 00400018
2	b (hex o dec)	35 (100011)	4	0
3	c (en dec)	24 (11000)	22	22

4	d (en dec)	11 (1011)	9	9
5	e (en dec)	0 (0000 00)	0	9
6	f (en dec)	11 = d	9	9
7	g (en dec)	4 (0100)	2	0x 4820
8	h (en hex)	0x 8D8B0004	0x 11690002	0x 1694820
9	i (hex o dec)	24 = c	11	22
10	j (en dec)	4	X	0
11	k (en dec)	4 = j	X	X
12	m (en hex)	0x00400014 = a + 4	0x 00400018	0x 00400022
13	n (en dec)	0 (inactive)	0	
14	p (hex o dec)		0	0
15	q (en dec)	0 (inactive)	0	
15	r (en hex)	0x00400014	0x 00400018	0x 00400022
17	s (hex o dec)	3	X	6

4. Cuestion 3

Identifica el valor de las señales de control de la figura 2 que se activarán cuando se ejecute la instrucción en el ciclo especificado. Rellena la tabla 2 con el valor de la señal de control.

Núm	Señales de control	Valor durante ciclo 4	Valor durante ciclo 5	Valor durante ciclo 6
18	RegDst	1	0	0
19	Branch	0	0	1
20	MemRead	0	1	0
21	MemtoReg	0	1	0
22	ALUOp	1	1	0
23	MemWrite	0	0	0
24	ALUSrc	0	0	0
25	RegWrite	1	1	0
26	PCSrc	0	0	1
27	Zero	0	0	1

5. Cuestion 4

¿Cuántos ciclos de reloj tarda en ejecutarse el código completo?

Tardara en ejecutarse 72 ciclos de reloj

Si la frecuencia de reloj es de 3GHz, ¿Cuánto tiempo tarda en ejecutarse?

Tardaria $72 / 3\,000\,000\,000$ seg; es decir $2,4 \times 10^{-8}$ seg

6. Cuestion 5

Modifica el programa de la actividad 4 para que almacene en la memoria los distintitos valores de Fibonacci, para ello sobrescribe el vector A. ¿Cuántos ciclos de reloj tarda ahora en ejecutarse el código completo?

```
.data
A: .word 2, 4, 6, 8, 10, 12, 14, 16, 18
n: .word 9

.text

add $t0, $0, $zero
addi $t1, $zero, 1
la $t3, n
lw $t3, 0($t3)
la $a0, A

fib:
    beq $t3, $0, final
    add $t2, $t1, $t0
    sw $t2, 0($a0)      # guardamos el valor de Fibonacci en la memoria
    move $t0, $t1
    move $t1, $t2
    subi $t3, $t3, 1
    addi $a0, $a0, 4     # incrementamos en 4 el puntero de la memoria para seguir agregando valores
    j fib

final:
    addi $a0, $t0, 0
    li $v0, 1
    syscall

    li $v0, 10
    syscall
```

El codigo ahora tardara ahora 93 ciclos en ejecutarse

Si la frecuencia de reloj es la misma que en la cuestión 4, ¿Cuánto tiempo tarda ahora en ejecutarse el programa?

Tardaría $93 / 3\,000\,000\,000$ seg; es decir $3,1 \times 10^{-8}$ seg

7. Cuestion 6

Modifica de nuevo el programa original de la actividad 4 para que muestre por pantalla el doceavo valor de la sucesión de Fibonacci. ¿Cuántos ciclos tarda ahora en ejecutarse el programa? ¿Cuánto tiempo necesitaría para ejecutarse si la frecuencia de reloj fuera de 500MHz?

```
.data
A: .word 2, 4, 6, 8, 10, 12, 14, 16, 18
n: .word 12      # solamente cambiamos la variable n para que ahora realice 12 veces el ciclo

.text

add $t0, $0, $zero
addi $t1, $zero, 1
la $t3, n
lw $t3, 0($t3)
la $a0, A

fib:
    beq $t3, $0, final
    add $t2, $t1, $t0
    sw $t2, 0($a0)
    move $t0, $t1
    move $t1, $t2
    subi $t3, $t3, 1
    addi $a0, $a0, 4
    j fib

final:
    addi $a0, $t0, 0
    li $v0, 1
    syscall

    li $v0, 10
    syscall
```

Ahora tardaría 123 ciclos en completarse por lo que el mismo procesador tardaría $4,1 \times 10^{-8}$ seg