Práctica 7

La ruta de datos

monociclo

Jordi Blasco Lozano

Arquitectura de comuputadores

Grado en Inteligencia Artificial

## Indice:

[Indice: 2](#_Toc158036761)

[1. Actividad 1 3](#_Toc158036762)

[3](#_Toc158036763)

## 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.

A screenshot of a computer

Description automatically generated

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

A screenshot of a computer

Description automatically generated

**¿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 intruccions lw y sw usan de la misma forma la carga y la escritura datos en la memoria, por lo que ambas calculan las direcciones de memoria.

## 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

final: addi $a0, $t1, 0 8

li $v0, 1 9

syscall 10

li $v0, 10 11

syscall 12

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Núm** | | **Etiqueta** | | **Valor durante**  **ciclo 5** | | **Valor durante**  **ciclo 6** | | **Valor durante**  **ciclo 7** | |
| 1 | | **a** (en hex) | |  | |  | |  | |
| 2 | | **b** (hex o dec) | |  | |  | |  | |
| 3 | | **c** (en dec) | |  | |  | |  | |
| 4 | | **d** (en dec) | |  | |  | |  | |
| 5 | | **e** (en dec) | |  | |  | |  | |
| 6 | | **f** (en dec) | |  | |  | |  | |
| 7 | | **g** (en dec) | |  | |  | |  | |
| 8 | | **h** (en hex) | |  | |  | |  | |
| 9 | | **i** (hex o dec) | |  | |  | |  | |
| 10 | | **j** (en dec) | |  | |  | |  | |
| 11 | | **k** (en dec) | |  | |  | |  | |
| 12 | | **m** (en hex) | |  | |  | |  | |
| 13 | | **n** (en dec) | |  | |  | |  | |
| 14 | | **p** (hex o dec) | |  | |  | |  | |
| 15 | | **q** (en dec) | |  | |  | |  | |
| 15 | | **r** (en hex) | |  | |  | |  | |
| 17 | | **s** (hex o dec) | |  | |  | |  | |