



UNIVERSIDAD DE GUADALAJARA
CENTRO UNIVERSITARIO DE CIENCIAS EXACTAS E INGENIERÍAS

ARQUITECTURA DE COMPUTADORAS

A - Decodificador de instrucciones tipo MIPS

Alumno: Erick Manuel González Carrillo (219695611)

Profesor: Jorge Ernesto Lopez Arce Delgado

30 de octubre de 2025

Índice

| | |
|------------------|---|
| 1. Introducción | 1 |
| 2. Marco teórico | 1 |
| 3. Desarrollo | 1 |

1. Introducción

El lenguaje de las computadoras suele ser complejo para la comprensión del humano puesto que la única forma en que pueda entender nuestras instrucciones es por medio de códigos binarios, sin embargo si queremos realizar demasiadas instrucciones llega a ser bastante tardado y difícil de comprender.

Por eso es que los programadores construyeron un traductor que sea capaz de realizar las mismas instrucciones pero que sea mas claro de entender, entre ellas surgió el lenguaje ensamblador; este lenguaje estaba directamente escrito en la maquina, ademas que las instrucciones se pasaban directamente a binario. por ejemplo, podemos realizar una suma en ensamblador:

```
1      add A, B
```

El ensamblador traducirá esta notación a:

```
1      1000110010100000
```

Es por ello que para esta actividad nos enfocaremos en desarrollar una aplicación que sea capaz de simular un código ensamblador de la arquitectura MIPS.

Este decodificador recibirá las instrucciones en ensamblador MIPS, convirtiéndolas en código binario de 32 bits y que sean guardadas en la memoria de instrucciones.

2. Marco teórico

Antes de realizar este decodificador debemos de saber que nos basaremos en la arquitectura MIPS de 32 bits con segmentación. Este diseño es capaz de realizar instrucciones tipo R e I, es por ello se considero el como se conforman estas instrucciones.

Así pues, para esta actividad se uso el lenguaje de programación Python. Esta herramienta es fácil de comprender y programar ayudando a realizar programas que en otros lenguajes se requieren estructuras bastantes complejas.

La aplicación esta desarrollada en un entorno gráfico (GUI) para eso se uso la librería TKinter. Con esta herramienta se pueden realizar aplicaciones GUI sencillas pero funcionales.

El punto principal de esta aplicación es que reciba una instrucción ensamblador y que este sea traducido en código binario; en una parte estará la instrucción en forma de 32 bits y el otro cuadro estará la misma instrucción en bytes. Ademas de que cuenta con la capacidad de guardar la instrucción en un archivo para que la memoria de datos leerá el código y ser ejecutado.

3. Desarrollo

Antes de pasar al desarrollo de la aplicación primero debemos de saber como es que se estructuran las instrucciones de la arquitectura MIPS de 32 bits.

Esto quiere decir que la instrucción cuenta con 32 espacios donde recibirá tanto la operación como las direcciones de extracción y destino de dicha instrucción. MIPS cuenta con 3 tipos de instrucciones, las de tipo R, I y J. Las cuales describiremos solo las de tipo R y I.

La instrucción de tipo R es una instrucción que realiza operaciones lógicas y aritméticas entre registros por lo que su estructura sera mostrada a continuación:

[1]

| | | | | | |
|--------|-------|-------|-------|-------|--------|
| 000000 | 10001 | 10010 | 01000 | 00000 | 100000 |
| 6 | 5 | 5 | 5 | 5 | 6 |

Tabla 1: Estructura de instrucción MIPS

Referencias

- [1] David A. Patterson and John L. Hennessy. *Estructura y diseño de computadores: la interfaz hardware / software*. Reverte, Barcelona, 2011.