

# Proyecto Final: Arquitectura de Computadoras

Jose Angel Preciado Delgadillo  
Carlos Damian Contreras

## Introducción

El presente proyecto tiene como objetivo diseñar y desarrollar un pipeline basado en la arquitectura MIPS32 de 5 etapas, una de las arquitecturas RISC más utilizadas en el ámbito académico y en aplicaciones embebidas.

A lo largo del documento se describen las etapas esenciales del diseño, incluyendo la definición del conjunto de instrucciones, el entendimiento de la ruta de datos, la implementación de la unidad de control, y las consideraciones necesarias para asegurar un funcionamiento eficiente y correcto del procesador. Este proyecto permite integrar conocimientos teóricos y prácticos, promoviendo una comprensión sólida de cómo se construyen y operan los procesadores modernos.

## Objetivo general

Comprender a profundidad el funcionamiento interno de una CPU MIPS32 de 5 fases aplicando el diseño de hardware usando Verilog para implementar dicho CPU. También entender como el CPU es capaz de entender las instrucciones que se precargan y entender a nivel bajo lo que ocurre con las computadoras, para así mejorar nuestras habilidades como programadores.

## Objetivos particulares

Al implementar dicho CPU nosotros mismos podemos saber como funciona todo por dentro. Al mismo tiempo escribiremos un compilador que traduzca las instrucciones que nosotros escribamos para programar un algoritmo no trivial para comprobar el funcionamiento del CPU implementado.

## Desarrollo

Estas son las instrucciones que el procesador soportará ordenadas por tipo:

Tipo R (\$, \$, \$)	Tipo I (\$, \$, #)	Tipo J (#)
ADD	ADDI	J
SUB	ANDI	
AND	ORI	
OR	XORI	
SLT	SLTI	
	BEQ	
	LW	
	SW	

## Proceso de Compilación (VASM compiler for MIPS32)

VASM compiler for Verilog es un compilador para MIPS32 robusto que nosotros desarrollamos a lo largo del semestre, el compilador antiguo era rudimentario y no permitía la escalabilidad que se requería, la interfaz gráfica usa el frontend del compilador VASM para que se encargue del proceso de compilación por separado. Los siguientes pasos son los que sigue VASM para poder compilar el ensamblador:

### Archivo:

- Lectura y carga del archivo.
- Ruptura del archivo en líneas y posteriormente en tokens.

### Sintaxis:

- Análisis de cada token por cada línea (Uso correcto de los signos de registro, de número inmediato y comas).
- Traducción de la instrucción hacia código binario empezando por la instrucción, luego los demás argumentos.
- Las instrucciones comparten una sola función por tipo de instrucción para el análisis sintáctico, entonces lo único que hay que hacer para agregar una instrucción es la traducción en binario del opcode, la instrucción en sí y su tipo, esto lo hace bastante escalable para implementar todas las instrucciones que soporta la arquitectura MIPS32.

### Manejo de errores:

- Si no se cumplen las reglas del análisis sintáctico, se levantará un error de compilación que informará al usuario de la línea y la estructura de la instrucción según el tipo de instrucción que se escribió de manera incorrecta.

## Interfaz de usuario gráfica

Para el desarrollo de la interfaz de usuario se usó la librería tkinter de python3, lo que nos permite que el usuario tenga una experiencia de desarrollo más amena, asimismo nos permite que el usuario seleccione el archivo a través de una ventana en vez de copiar y pegar la ruta del archivo que se quiere compilar, también permite al usuario código de manera inmediata usando el espacio de texto.

## Algoritmo de demostración

Se va a buscar un número en la memoria de datos.

## Conclusión

El desarrollo del CPU MIPS32 acompañado de su compilador y una interfaz gráfica representa un proyecto integral que combina conocimientos de arquitectura de computadoras, diseño de hardware, sistemas embebidos y desarrollo de software. La implementación del procesador nos permitió comprender a profundidad el funcionamiento interno de una arquitectura tipo RISC, desde el ciclo fetch hasta la gestión de señales de control y la interacción entre sus módulos funcionales.

La creación del compilador VASM consolidó el entendimiento del proceso de un compilador y el lenguaje ensamblador propio del procesador, asegurando que el hardware diseñado fuera utilizable y verificable mediante el programa que escribiremos en él.

El diseño de la interfaz gráfica nos dió mas entendimiento a cómo nosotros podemos hacer que una persona se sienta cómoda con una GUI, y darnos otra perspectiva sobre la practicidad y entendimiento que tiene el usuario sobre el programa.

## Bibliografía

*MIPS® Architecture for Programmers Volume II-A: The MIPS32® Instruction Set Manual* (6.<sup>a</sup> ed., Vol. 2). (s. f.).

<https://github.com/DamianKZK/Python> (Graphical User Interface)

<https://github.com/AnGlonchas/VASMCompilerForMIPS32> (VASM Compiler)

<https://github.com/DamianKZK/ArquitecturaDeComputadoras> (Verilog MIPS32 Implementation)