Proyecto Final Fase 01 - Data Path

**Integrantes**

Alan Sabastian Guzmán Marquez

Alcaraz Suarez Gabriel Isaí

Kevin Manuel Sanchez Agredano

Quintero Gonzalez Diego Gerardo

**Introducción**

En el desarrollo de sistemas digitales, la implementación de procesadores es fundamental para el funcionamiento de dispositivos electrónicos. Comprender los elementos básicos de un procesador, como la Memoria, la Unidad Aritmético-Lógica (ALU), y el Banco de Registros, es esencial para diseñar y construir sistemas computacionales eficientes y funcionales.

Una vez entendido el funcionamiento de estos elementos básicos, es posible avanzar en la implementación de procesadores más complejos. En esta fase del proyecto, se aborda la construcción de un "DataPath", que forma parte de una versión simplificada del procesador MIPS. El objetivo principal de este DataPath es implementar y decodificar instrucciones básicas tipo R. Las instrucciones tipo R se caracterizan por tener una estructura de 32 bits, dividida en seis elementos: OpCode, RS, RT, RD, Shamt y Function. El OpCode y el Function son campos de 6 bits cada uno. Al ingresar a los módulos correspondientes, se van separando y procesando para ejecutar la instrucción adecuada.

El objetivo principal de esta práctica es crear un DataPath capaz de ejecutar las instrucciones mencionadas. Para lograrlo, se integran los módulos utilizados previamente (ALU, BR y Memoria) con nuevos módulos que gestionarán el flujo de bits en el sistema. Estos nuevos módulos incluyen la Unidad de Control (UC), ALU-Control y Mux 2:1.

En resumen, esta práctica tiene como objetivo implementar un DataPath capaz de ejecutar instrucciones tipo R en un procesador, integrando módulos existentes con nuevos módulos de control y selección de datos. El éxito en esta implementación permitirá comprender en detalle el funcionamiento interno de un procesador y sentar las bases para diseños más complejos en el futuro.

**Objetivo**

* Objetivo General

Crear un DataPath capaz de ejecutar las instrucciones tipo R en un procesador MIPS, integrando los módulos utilizados en actividades anteriores (ALU, BR y Memoria) con nuevos módulos (Unidad de Control, ALU-Control, y Mux 2:1).

* Objetivos Particulares

1. Integrar los módulos ALU, BR, y Memoria utilizados en la actividad anterior en el nuevo DataPath.
2. Diseñar y desarrollar la Unidad de Control (UC) con las siguientes características:
   * Entrada de 6 bits para recibir los bits del Código de Operación (Opcode).
   * Tres salidas de 1 bit: MemToReg, RegWrite, y MemToWrite.
   * Salida de 3 bits, ALUOp, conectada a la ALU-Control para indicar la operación de la ALU.
3. Implementar la ALU-Control que:
   * Recibe una entrada de 6 bits (bits 0:5 de la instrucción) y una entrada de 3 bits de la UC.
   * Genera una salida de 3 bits para la ALU, indicando la operación que debe realizar.
4. Desarrollar el Mux 2:1 con dos entradas de 32 bits cada una y una entrada de 1 bit para seleccionar qué entrada se pasa a la salida. La salida es de 32 bits, donde se canaliza el dato seleccionado para mandar a la entrada de WriteData del Banco de Registro.
5. Crear un TestBench para verificar su funcionamiento correcto.
6. Probar el DataPath completo (DPTR) enviándole 10 instrucciones, dos de cada una de las instrucciones, y verificar que las salidas sean correctas.

**Desarrollo**

* Descripción de las instrucciones tipo R

Las instrucciones tipo R son un tipo de instrucción en lenguaje de máquina utilizadas en arquitecturas de computadoras, especialmente en procesadores tipo MIPS. Estas instrucciones involucran registros y operaciones aritméticas o lógicas. En una instrucción tipo R, los campos de la instrucción están organizados de la siguiente manera:

- OpCode: 6 bits

- RS: 5 bits (registro fuente 1)

- RT: 5 bits (registro fuente 2)

- RD: 5 bits (registro destino)

- Shamt: 5 bits (desplazamiento)

- Function: 6 bits (código de función específico de la operación)

La instrucción SLT (Set on Less Than) se utiliza para comparar dos valores y establecer un registro con el valor 1 si el primer valor es menor que el segundo, o con el valor 0 en caso contrario. Su formato en lenguaje ensamblador es:

SLT $rd, $rs, $rt

Donde $rd es el registro destino, $rs es el registro fuente 1 y $rt es el registro fuente 2. La operación SLT es útil en la implementación de estructuras de control de flujo condicionales.

* Investigación sobre la operación ternaria

La operación ternaria es una operación condicional que se encuentra en muchos lenguajes de programación. También se conoce como operador ternario. Su forma general es:

condición ? expresión1 : expresión2

Donde la condición es evaluada primero. Si es verdadera, se evalúa expresión1 y se devuelve su valor. Si es falsa, se evalúa expresión2 y se devuelve su valor. La operación ternaria es una forma concisa de escribir una estructura condicional if-else en una sola línea.

* Investigación sobre el proceso de compilación

El proceso de compilación es el proceso mediante el cual un programa escrito en un lenguaje de alto nivel se traduce a un lenguaje de máquina que la computadora puede entender y ejecutar. El proceso de compilación consta de varios pasos:

1. Análisis léxico: El código fuente se divide en tokens (palabras clave, identificadores, literales, etc.).

2. Análisis sintáctico: Se verifica la estructura gramatical del código fuente para garantizar que cumpla con las reglas del lenguaje.

3. Análisis semántico: Se verifica el significado del código para detectar errores semánticos.

4. Generación de código intermedio: Se genera un código intermedio que representa el programa de manera más abstracta.

5. Optimización de código: Se aplican diversas técnicas para mejorar el código intermedio y hacerlo más eficiente.

6. Generación de código objeto: Se genera el código objeto específico de la arquitectura de la computadora.

7. Enlazado: Se combinan los diferentes módulos de código objeto y bibliotecas para formar un programa ejecutable.

El proceso de compilación es fundamental para la programación en lenguajes de alto nivel, ya que permite que los programas escritos por los programadores se ejecuten en la computadora de manera eficiente.

* Unidad de Control (UC)

La Unidad de Control recibe una entrada de 6 bits correspondiente al Código de Operación (OpCode) y tiene tres salidas de 1 bit: MemToReg, RegWrite y MemToWrite. Además, tiene una salida de 3 bits, ALUOp, que se conecta a la ALU-Control. Esta unidad define las señales que se enviarán a los diferentes elementos del sistema dependiendo del OpCode que reciba.

* ALU-Control

Esta unidad recibe una entrada de 6 bits, que corresponde a los bits 0:5 de la instrucción, y una entrada de 3 bits que llega de la UC. Tiene una salida de 3 bits que se envía a la ALU para indicar la operación que debe realizar. La ALU-Control escucha a la UC y, dependiendo del código que reciba, hace caso o no de los bits 5:0 de la instrucción.

* Mux 2:1

Este multiplexor tiene dos entradas de 32 bits cada una y una entrada de 1 bit para seleccionar qué entrada se pasa a la salida. La salida es de 32 bits, donde se canaliza el dato seleccionado.

* TestBench

Para el TestBench del DataPath Tipo-R (DPTR), se debe hacer un test para los módulos nuevos. Se debe probar el DPTR enviándole 10 instrucciones, dos de cada una de las instrucciones mencionadas en la introducción. Se debe incluir una tabla con las instrucciones en formato ensamblador y otra con el código máquina correspondiente.

**Conclusión**

En esta práctica, logramos diseñar e implementar un DataPath capaz de ejecutar instrucciones básicas tipo R. Se integraron los módulos existentes con nuevos módulos de control, permitiendo el funcionamiento adecuado del procesador en la ejecución de las instrucciones mencionadas.

# Bibliografía

Rocha Pacheco, N. (2021). Diseño, implementación y validación de un núcleo de procesador basado en el conjunto de instrucción RISC-V. https://repositorio.uniandes.edu.co/entities/publication/ee3d1db5-1a73-4855-a3cb-c5827305bc30

Ramírez, C. A. R., Quintero, D. M. M., & Builes, J. A. J. (2021). Un Ambiente visual integrado de desarrollo para el aprendizaje de programación en robótica. *Investigación e Innovación en Ingenierías*. https://revistas.unisimon.edu.co/index.php/innovacioning/article/view/3957