**Universidad Nacional de Tucumán**

**Facultad de Ciencias Exactas y Tecnología**

****

Proyecto Integrador Electrónica II - Núcleo RISC-V

**Campos Pablo Patricio**

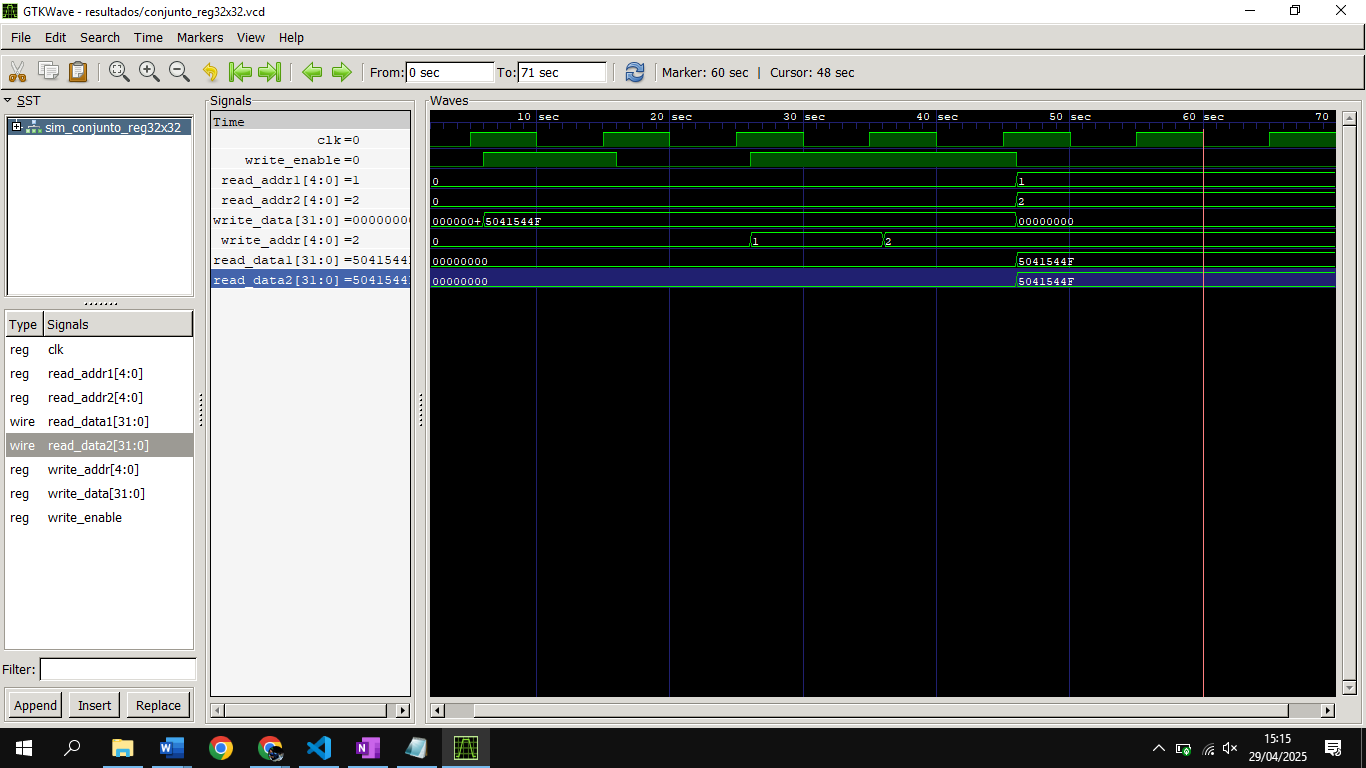
**Ingeniería Electrónica**

**Resumen**

Este proyecto consiste en el diseño, implementación y simulación de un procesador basado en un conjunto de instrucciones tipo RISC-V.

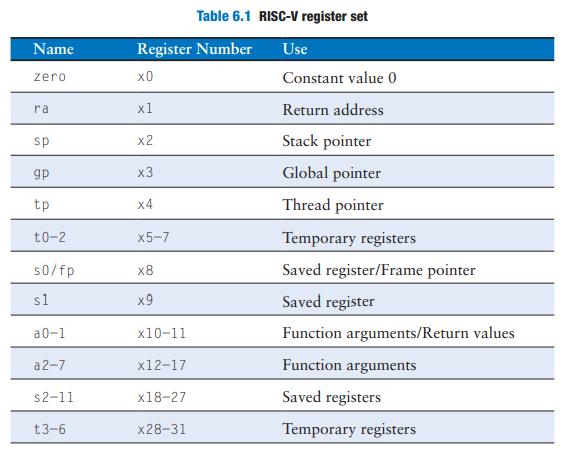
Luego se realizó un datapath multiciclo para implementar instrucciones RV32I, integrando bloques que fueron propuestos y realizados previamente.

**Conjunto de registros 32x32**



**Conjunto de registros de la arquitectura RISC-V**

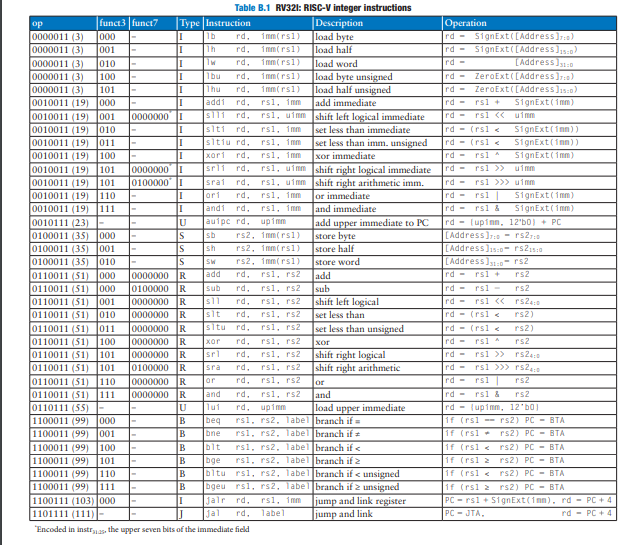
El conjunto de registros de la arquitectura RISC-V, se compone de 32 registros de propósito general. Se utilizan para almacenar datos temporales durante la ejecución de las instrucciones del programa. Cada registro se puede utilizar para operaciones aritméticas, lógicas, y almacenamiento de resultados temporales. No hay restricciones estrictas sobre qué tipo de datos pueden almacenarse en cada registro, pero algunas convenciones de la arquitectura y el sistema operativo definen ciertos registros con fines específicos.

****

*Tabla B.4 Números y nombres de registros [2].*

**Conjunto de instrucciones RV32I**

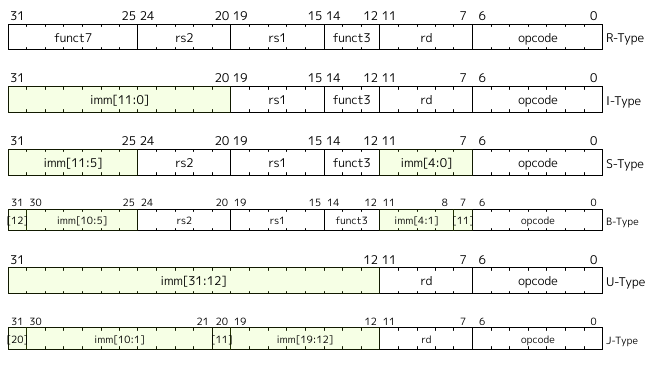
El conjunto de instrucciones de **RISC-V** es una **arquitectura de conjunto de instrucciones (ISA)** basada en el principio **RISC (Reduced Instruction Set Computer)**, lo que significa que está diseñado para ser simple, eficiente y escalable. Se organiza en un conjunto **base obligatorio** y varias **extensiones opcionales** que amplían sus capacidades. El conjunto base de instrucciones define las operaciones fundamentales que cualquier procesador RISC-V debe soportar [1].



*Tabla B.1 RV32I: RISC-V integer instructions [1].*

**Describir la estructura de las instrucciones tipo R, I, S, B, U y J**

Dependiendo el tipo de instrucción que sea, va a tener una estructura. A continuación, se adjunta en la siguiente imagen sobre la estructura de las instrucciones tipo R, I, S, B, U, J [2].



*Estructura de las Instrucciones tipo R, I, S, B, U y J [2].*

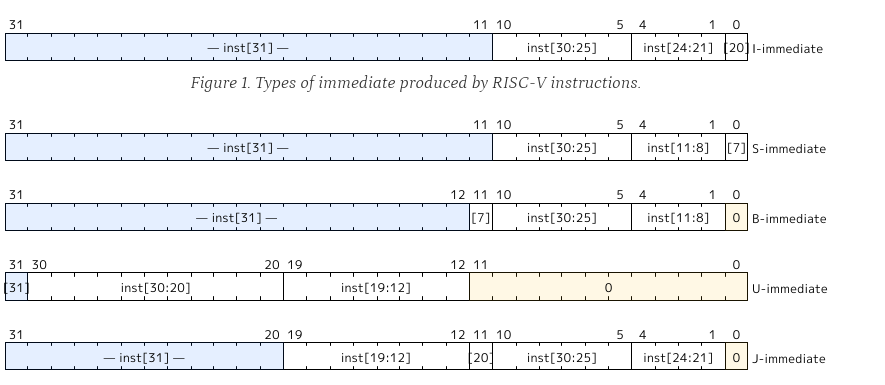
**Determinar cómo se calcula el valor inmediato para los tipos I, S, B, U y J**

Para determinar cómo se calculan los valores inmediatos, se usó como guía la tabla, donde indica como está compuesta el valor inmediato en cada instrucción.

En las instrucciones de tipo I y S, el inmediato está formado por 12bits, donde al bit más significativo inst[31] lo extendemos 21 veces para formar un valor de 32bits.

En las instrucciones de tipo J el inmediato está formado por 21bits, dividido en 4 partes como se pude ver en la siguiente imagen.

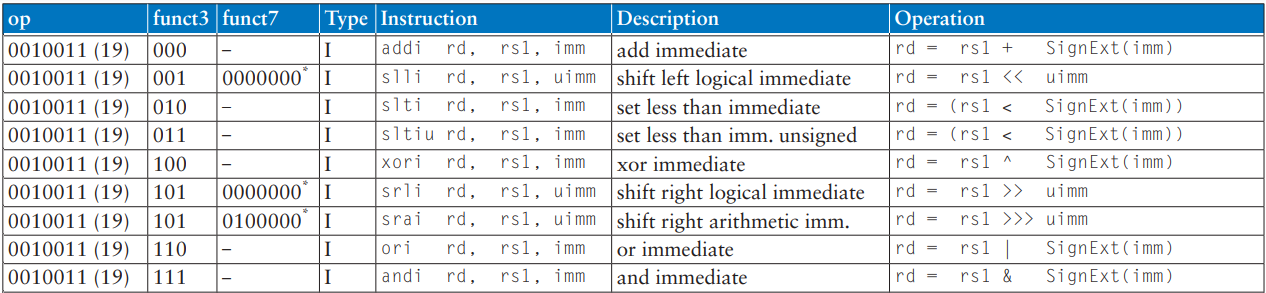
En las instrucciones de tipo B, el inmediato está formado por 13bits, dividido en 4 partes como se puede ver en la siguiente imagen.



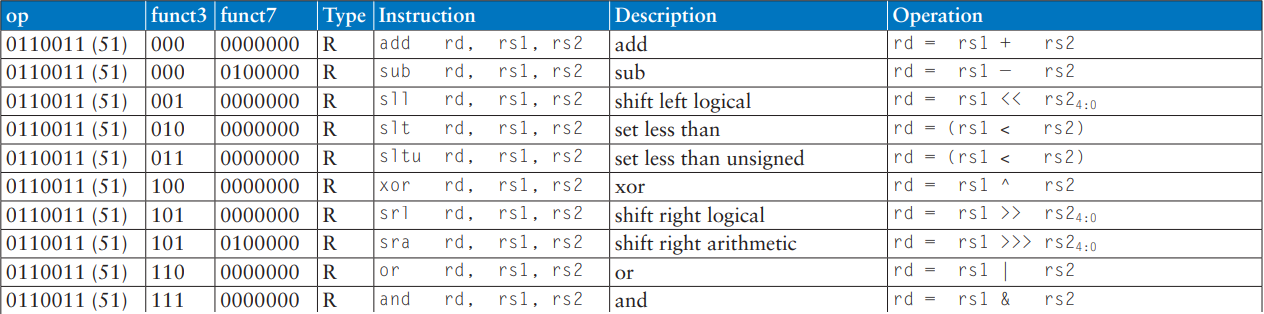
*Codificación de inmediato para tipo R, I, S, B, U y J [2].*

**Analizar las instrucciones de código de operación decimal 19 y 51, encontrar la manera en que se codifica la operación matemática de la ALU**

Las instrucciones tanto la 19 como la 51, el campo funct3 define la operación básica y el funct7 nos sirve para distinguir entre variantes de una operación. En el caso del funct7 nos fijamos en el bit 5. A continuación están las tablas que se usaron para determinar cómo se codifican dichas instrucciones.

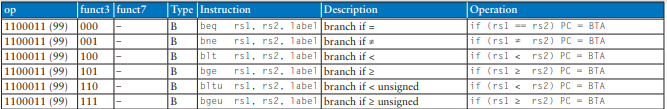
****

*Tabla B.1 RV32I: RISC-V integer instructions [1].*



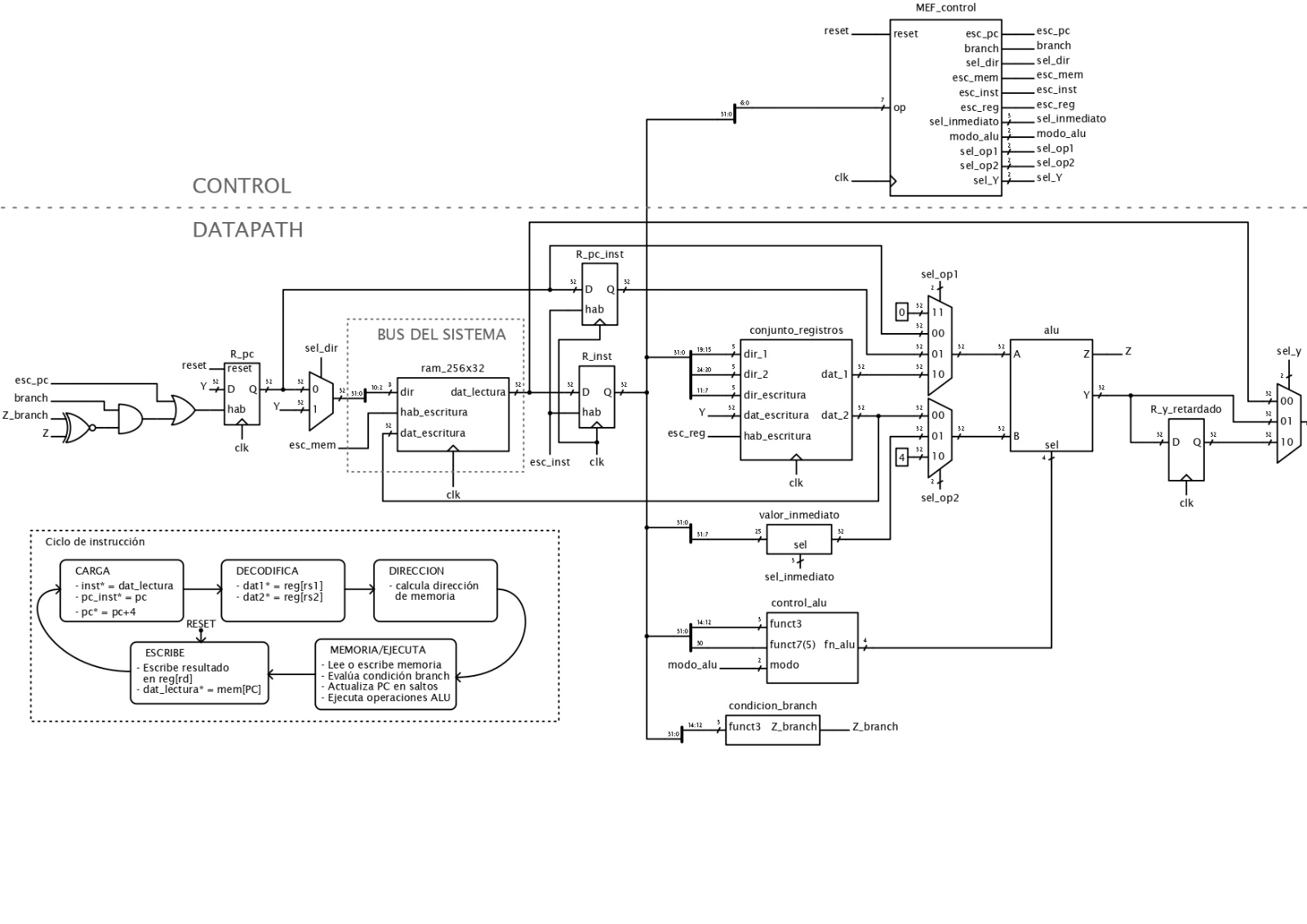
*Tabla B.1 RV32I: RISC-V integer instructions [1].*

**Analizar las instrucciones de código de operación decimal 99 (saltos condicionales), determinar la forma en que se codifica la operación de la ALU y la condición de cero necesaria para tomar el salto**

****

*Tabla B.1 RV32I: RISC-V integer instructions [1].*

**Resultados**

****

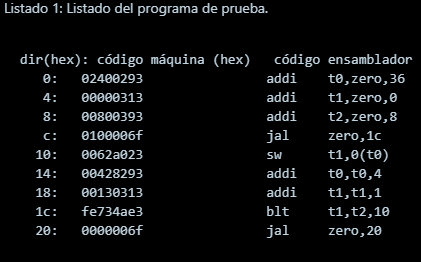
Componentes desarrollados:

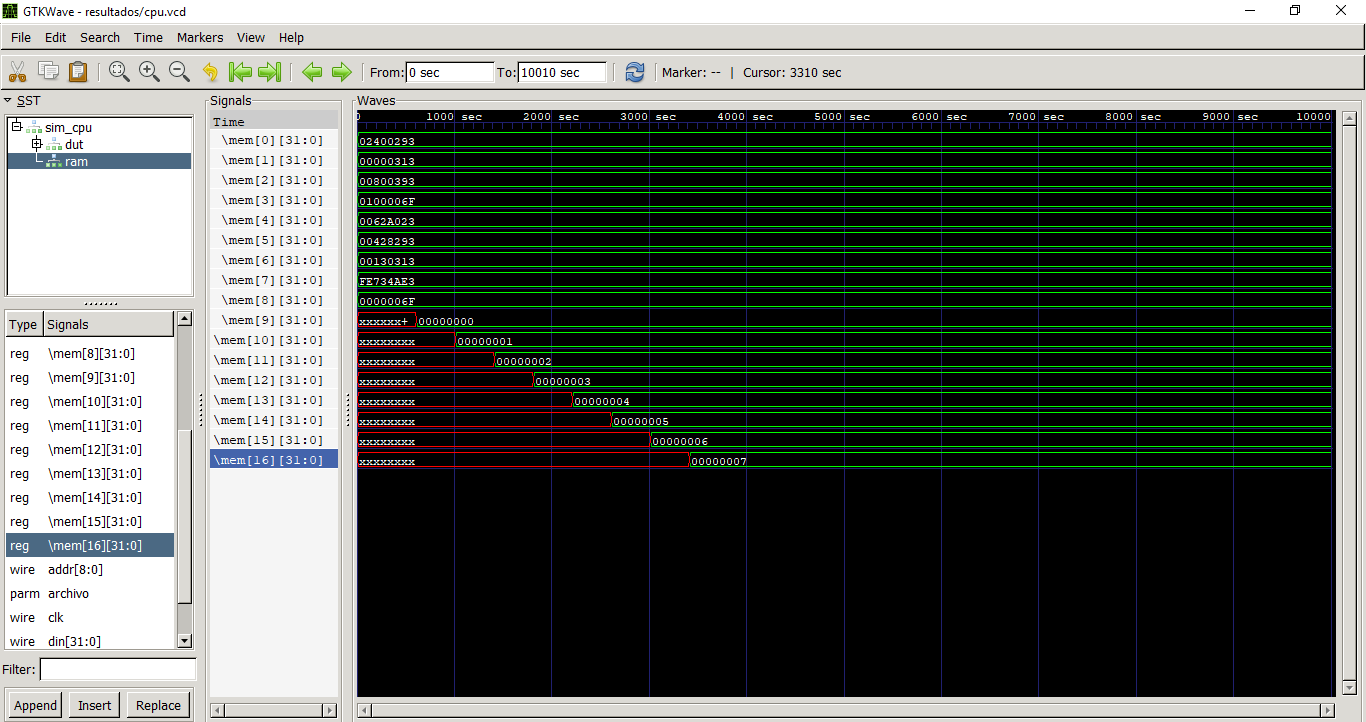
1. Función CERO.
2. Función AND.
3. Función OR.
4. Función XOR.
5. Función SUMA/RESTA.
6. Función MENOR QUE.
7. Función desplazamiento a la izquierda.
8. Función desplazamiento a la derecha.
9. Unidad aritmética lógica de 32bits.
10. Registros de 32bits.
11. Memoria RAM.
12. Conjunto de registros.
13. Valor inmediato.
14. Control ALU.
15. Condición Branch.
16. Unidad de control (MEF).

Las simulaciones de cada componente se encuentran en el repositorio.

**Simulación del CPU**

Para evaluar el correcto funcionamiento del procesador, se realizó la respectiva simulación con un programa de prueba, dado en código ensamblador y código máquina. A continuación, el código que se utilizo y una captura de la simulación del procesador en funcionamiento.





*Ventana programa GTKWave simulación procesador.*

**Biografía**

[1] Sarah L. Harris, David Money Harris, “Digital Desing and Computer Architecture RISC-V Edition”, 2021.

[2] The RISC-V Instruction Set Manual Volumen I, “Unprivileged Architecture”, 2024.