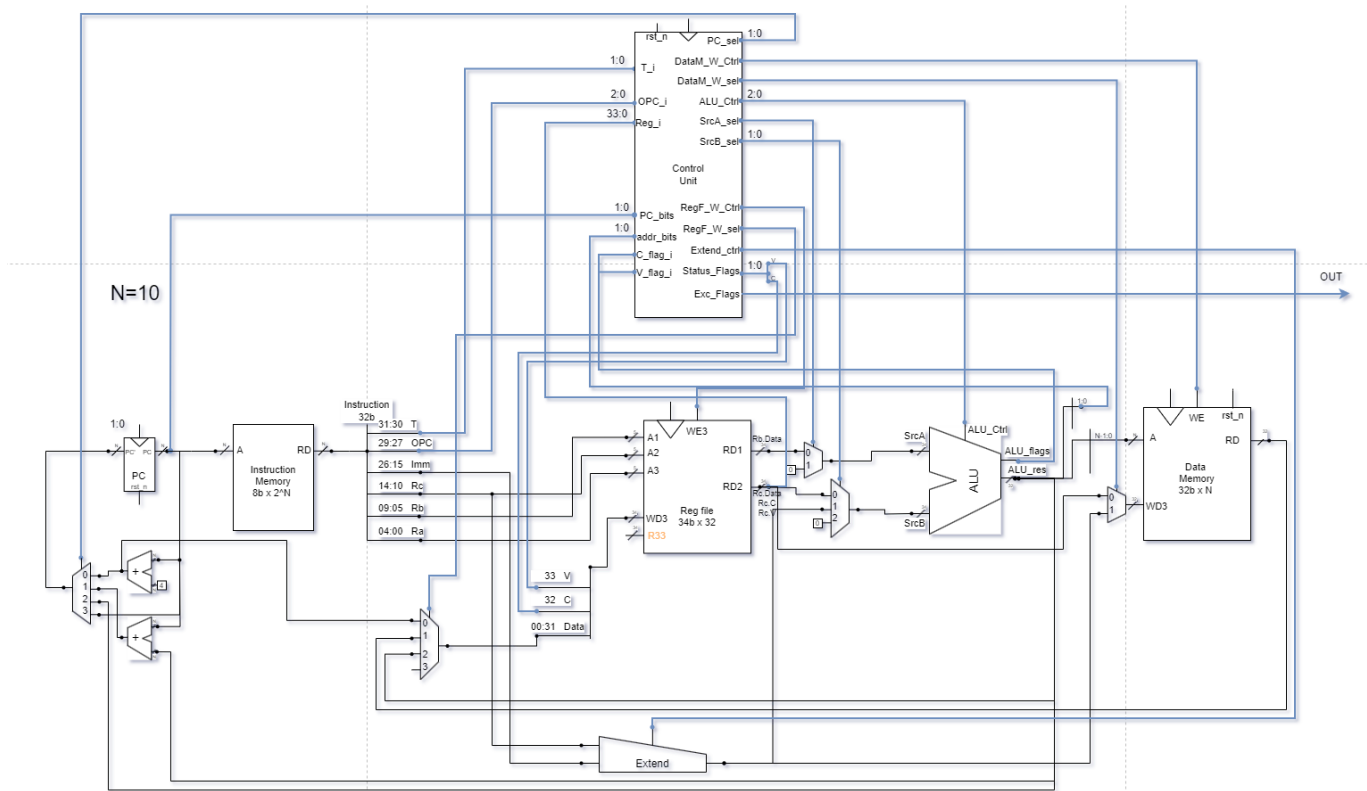


## RISC\_One\_Cycle - Especificaciones del Diseño

### Resumen

**RISC\_One\_Cycle** contiene la implementación de un procesador RISC One Cycle que cumple con las especificaciones proporcionadas por la guía **RTL Exercise Training** de Allegro.

### Diagrama



### Módulos

- **top\_module.sv**: Módulo principal. En este módulo se instancian y conectan el resto de los módulos.
- **control\_unit\_mod.sv**: Unidad de control. Este módulo genera las señales de control necesarias para el funcionamiento del procesador, decodificando las instrucciones y activando las señales y excepciones correspondientes.
- **data\_memory\_mod.sv**: Memoria de datos.
- **register\_file\_mod.sv**: Memoria de registros. Contiene los 32 registros del procesador.
- **instruction\_memory\_mod.sv**: Memoria de instrucciones. Almacena el conjunto de instrucciones que serán ejecutadas por el procesador.
- **program\_counter\_mod.sv**: Contador de programa. Registro de la dirección de la siguiente instrucción a ejecutar.
- **extender\_mod.sv**: Extensor de señal. Extiende a 32 bits el parametro Immediate o su concatenacion con el parametro Rc.
- **alu\_mod.sv**: Unidad Aritmético-Lógica. Realiza operaciones aritméticas y lógicas sobre los datos proporcionados por los registros o la memoria.

### Entradas

- **clk**: Señal de reloj.
- **rst\_n**: Señal de reinicio activa en bajo. Cuando está en bajo (0) resetea el módulo al estado "A".

## Salidas

- **exception\_flags\_o**: Señales de bandera de excepción de 3 bits: Instrucción inválida, dirección de memoria inválida y dirección de program counter inválida.
- **alu\_result\_o**: Resultado de 32 bits de la operación realizada por la ALU.
- **reg\_file\_r\_data1\_o**: Datos de 34 bits leídos del primer registro de la memoria de registros.
- **reg\_file\_r\_data2\_o**: Datos de 34 bits leídos del segundo registro de la memoria de registros.
- **data\_mem\_r\_data\_o**: Datos de 32 bits leídos de la memoria de datos.
- **pc\_o**: Contador de programa de N bits.
- **inst\_read\_data\_o**: Datos de 32 bits leídos de la memoria de instrucciones.
- **alu\_C\_flag\_o**: Señal de bandera de carry de la ALU. Indica si hubo un acarreo en la última operación aritmética.
- **alu\_V\_flag\_o**: Señal de bandera de overflow de la ALU. Indica si hubo un desbordamiento en la última operación aritmética.

## Estructura del proyecto

El proyecto se encuentra dividido en 5 carpetas:

- Modules: Aquí se encuentran los módulos que conforman procesador.
- testbenchs: Contiene los testbenchs correspondientes cada uno de los módulos.
- Assembler: Aquí puede encontrarse el compilador de assembler desarrollado en python, junto con el programa utilizado y un txt con su explicación.
- Memory files: Contiene los datos a precargar en las memorias para el target testbench.
- Documentación: Aquí se almacenan todos los archivos relevantes para la documentación.

```

RISC_One_Cycle
├── testbenchs
│   ├── alu_tb.sv
│   ├── top_module_target_tb.sv
│   ├── control_unit_tb.sv
│   ├── data_memory_tb.sv
│   ├── instruction_memory_tb.sv
│   ├── program_counter_tb.sv
│   ├── register_file_tb.sv
│   ├── extender_tb.sv
│   └── top_module_predictor_tb.sv
├── Assembler
│   ├── RISC_compiler.py
│   ├── output.bin
│   ├── program1.asm
│   └── program1_explained.txt
├── Memory files
│   ├── data_memory.hex
│   └── reg_memory.bin
└── Modules

```

```
|  |  | alu_mod.sv
|  |  | control_unit_mod.sv
|  |  | register_file_mod.sv
|  |  | data_memory_mod.sv
|  |  | extender_mod.sv
|  |  | instruction_memory_mod.sv
|  |  | program_counter_mod.sv
|  |  | top_module.sv
|  |  | Documentation
|  |  | One_cycle_diagram.png
|  |  | RISC_One_Cycle_Specs_Hipperdinger.pdf
```

### Simulación online

<https://www.edaplayground.com/x/7vE8>