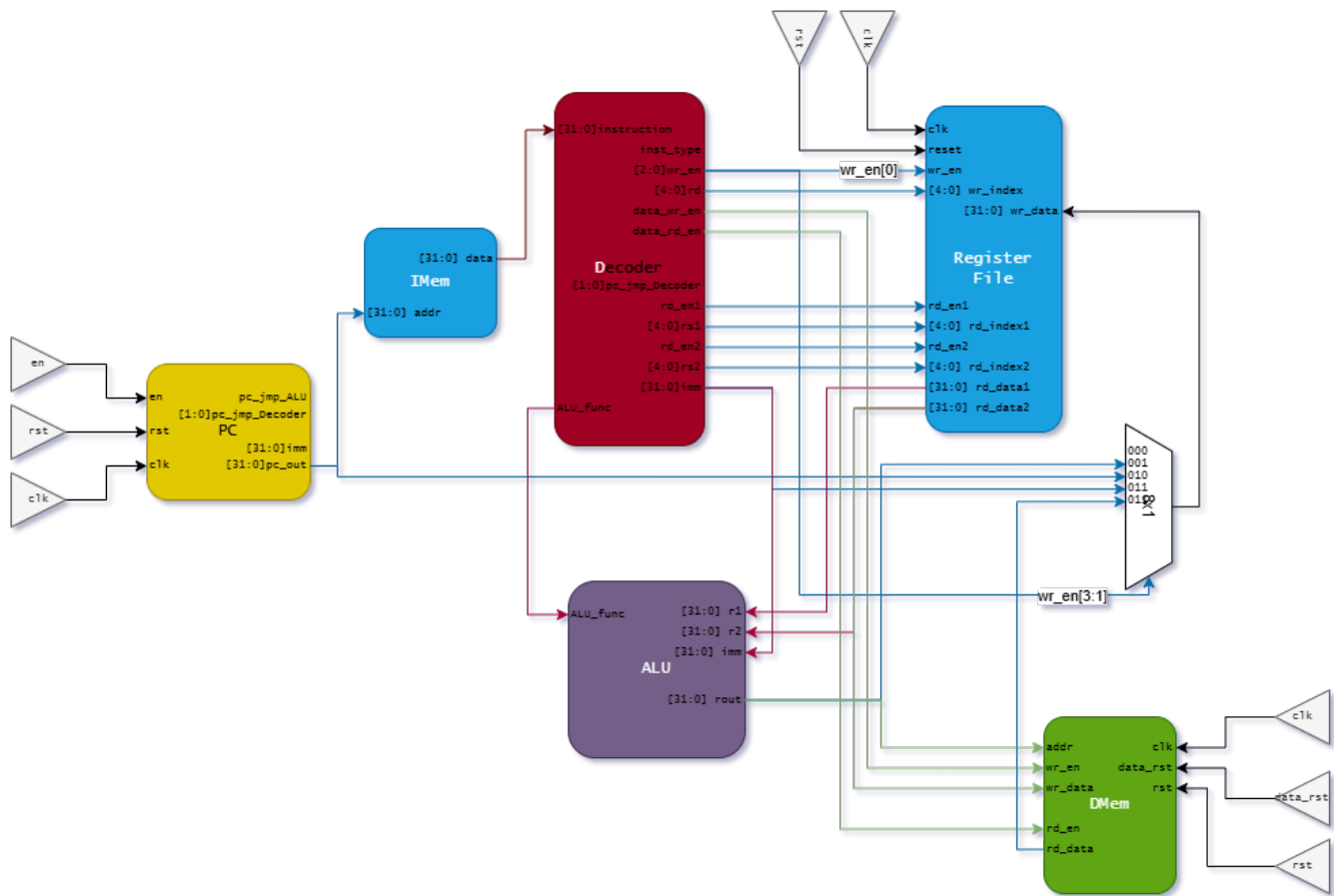


# core specifications Rev3



## 1. ISA:

- base ISA: RV32M
- modular core structure allows stripping down to RV32E if required

## 2. Execution model:

- single cycle CPU
- enable functionality included (you can stall the CPU state)

## 3. Registers:

- general purpose:
  - 32 x 32 bit registers (x0 - x31)
  - reset and enable (stall) functionality included
- flag registers:
  - 1 x 64 bit (or 32 bit) flag/control register used for controlling muxes across instruction types
  - extendable: external modules like UART can introduce new flag registers

- **not decided: do we have multiple flag registers or a singular flag register is enough?**
- instruction register (buffer):
  - 1 x 32 bit register for holding the current instruction, getting input from external instruction memory, and giving it to the decoder
  - ~~not decided: do we need multiple instruction registers or is one enough (since we are not doing pipelining)~~

#### 4. Program counter

- 32 bit width
- default increment:  $PC = PC + 4$  (byte addressable memory)
- ~~support for word addressable jumps/branches (is this required?)~~

#### 5. control unit:

- decoder:
  - generates control signals for register file, flag register and mux selection
- control mux:
  - decides what gets to write back to the register file (ALU result, immediate value, data memory output)

#### 6. ALU:

- supports all arithmetic and logical instructions, including multiply and divide

#### 7. external interface:

##### 1. memory interface:

- instruction memory: 32 bit
- data memory:
  - 32 bit, word aligned access supported

##### 2. SPI interface:

- the required SPI pins (clk, mosi, miso, ss) will be exposed from the riscv core
- **not decided: we might need a separate SPI controller block inside the core.**

##### 3. Interrupt interface:

- need to discuss on how the core will save system state, serve the interrupt request, and come back to the previous state