Write-up

The machine language for this primitive processor is made up of instruction words that are used to control its operations. Each instruction word is composed of multiple fields, each with a specific purpose. The first 3-bit field is the opcode field, which specifies the operation to be performed by the ALU. This field is responsible for determining the type of operation that the processor should perform, such as addition, AND, OR, and subtraction. The first bit of the opcode field is the Immediate bit, which toggles whether data goes into the "bottom" of the ALU from the register file or directly from the RA2 field of the instruction word.

The next field is the Write Address (WA) field, which specifies the register where the result of the operation will be stored. The bit width of this field will depend on the number of registers in the Register File. For example, if the Register File has 8 registers, the WA field will require 3 bits to specify which register to store the result in.

The next two fields are Read Address 1 (ra1) and Read Address 2 (ra2) fields. These fields specify the registers that hold the operands for the operation. Like the WA field, the bit width of these fields will also depend on the number of registers in the Register File.

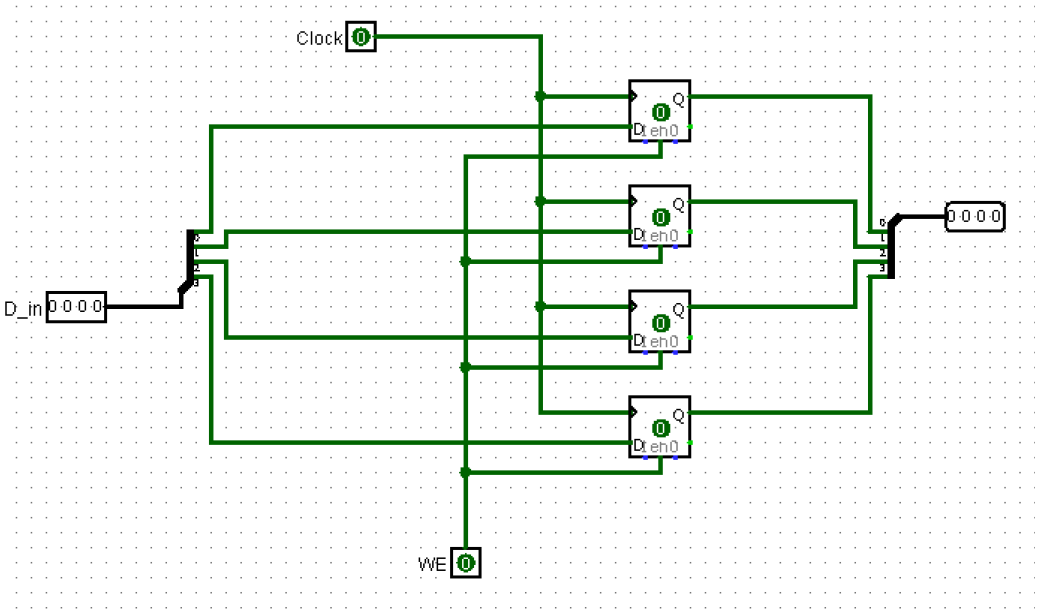
With the instruction format, the instruction word is created with a total bit width of 3+3+3+3. The instruction word format for immediate instructions requires an extra bit to indicate whether the operation is an immediate instruction or not.

The largest immediate value that can be put into a register via ADDI is 7, and the smallest immediate value is -8. This is because the RA2 is 3 bits and the I is 1 bit. together after sign extension when they are given as input, they act as a simple 4-bit number. This range may not be useful for certain operations and can be improved by increasing the number of bits allocated for the immediate value in the instruction word. However, keep in mind that adding more bits to the immediate field will increase the size of the instruction word, which can in turn decrease the amount of instruction words that can be stored in memory. Additionally, it will also affect the performance of the processor as it need more time to process the instruction with larger bit width.

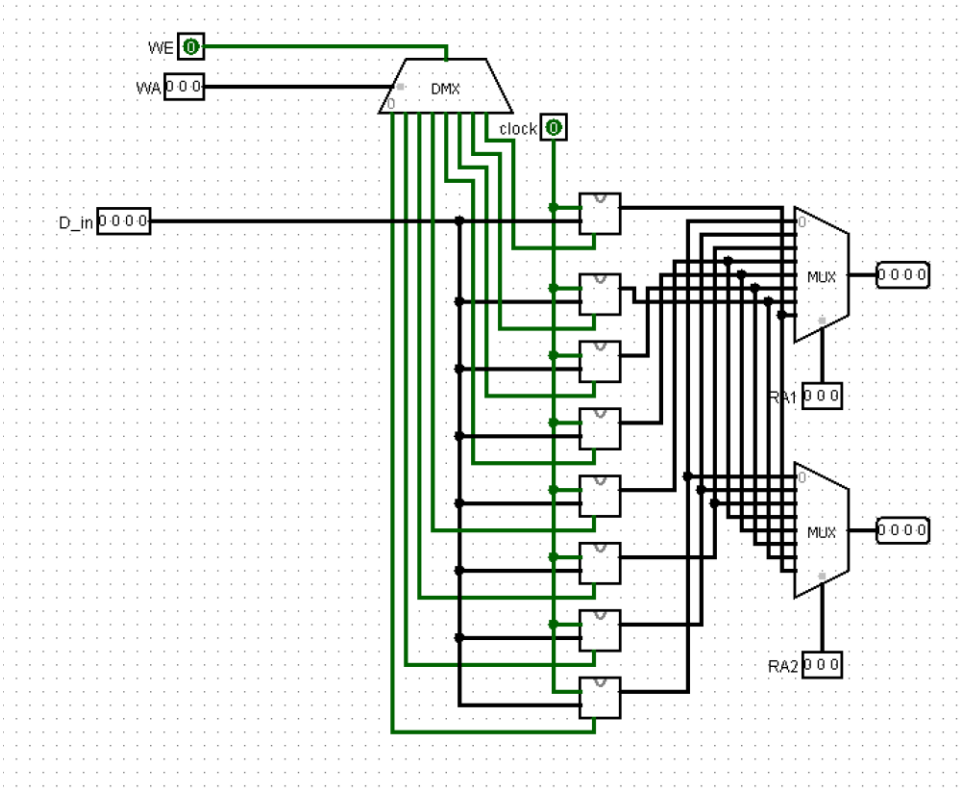
The missing bit that is mentioned in part 6 is the Sign Extender which is used to extend the bit width of the immediate value. This allows for greater immediate value range and allows more complex instructions to be executed by the ALU.

Testing

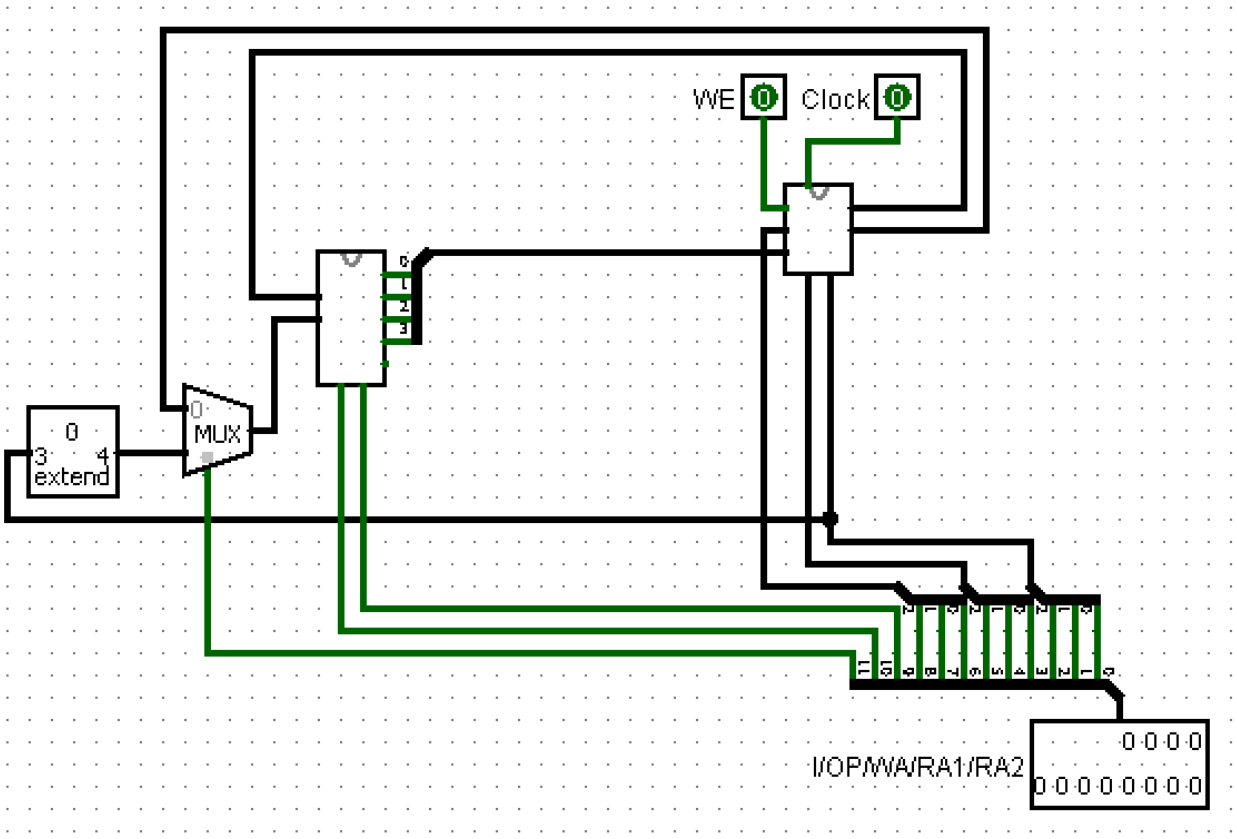
4-bit register



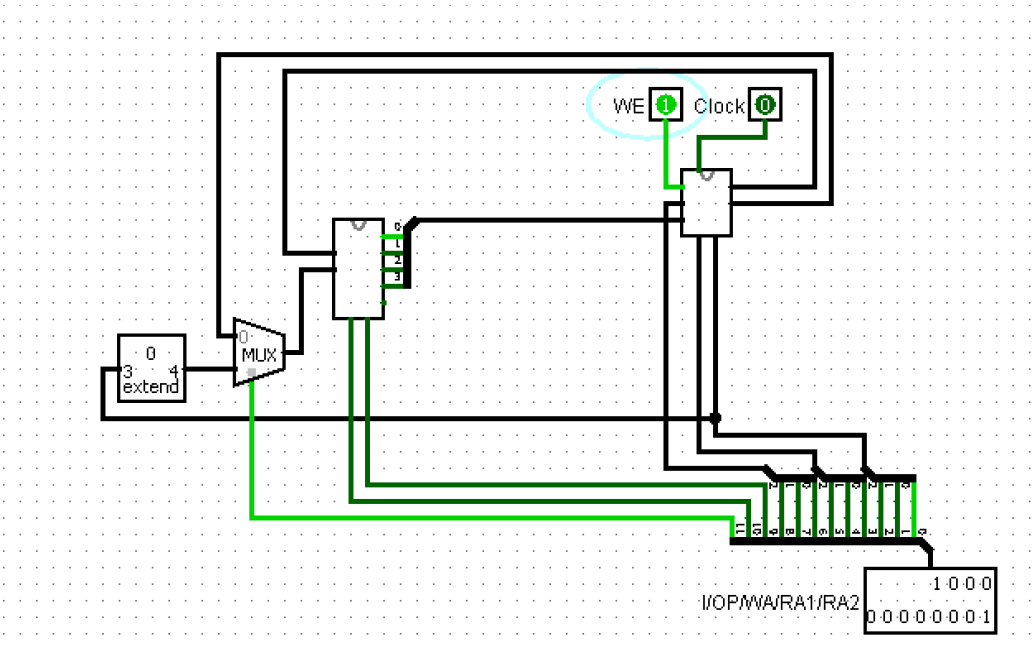
Register File



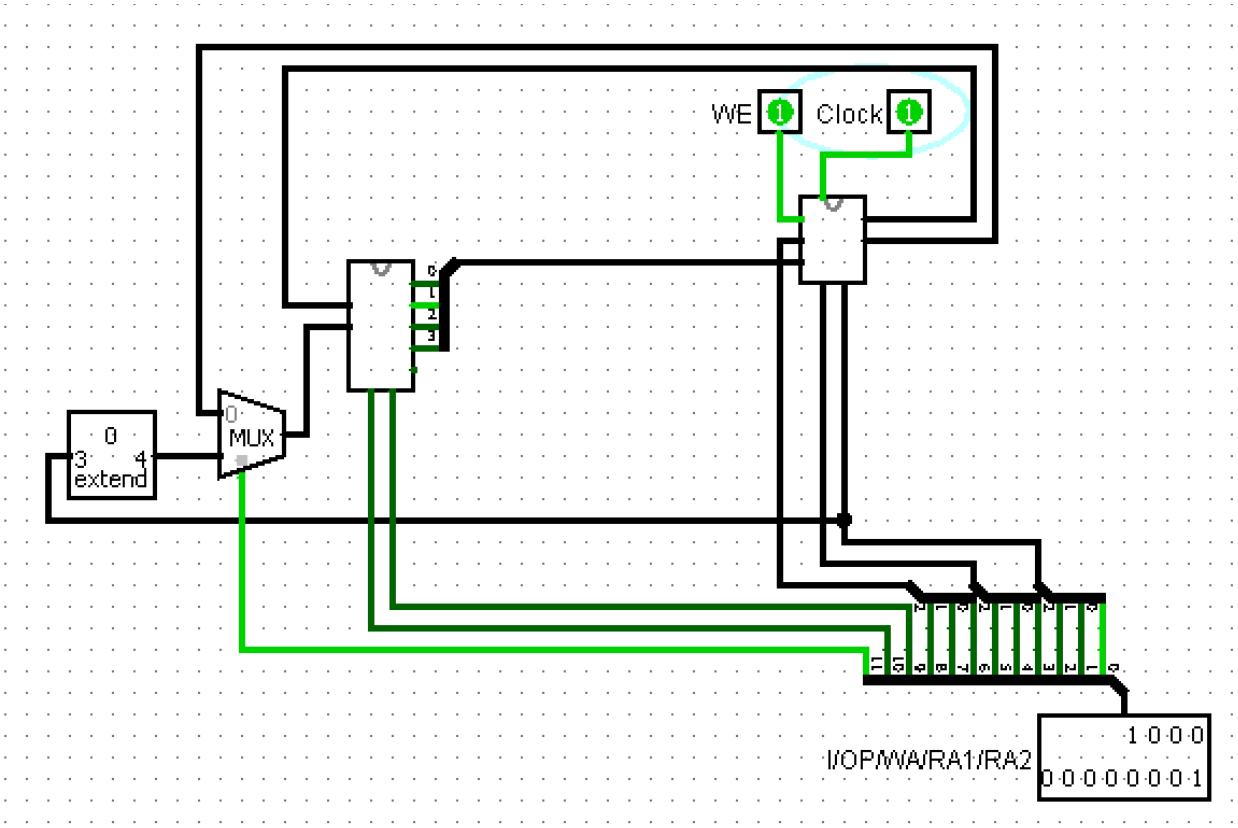
Register File + ALU



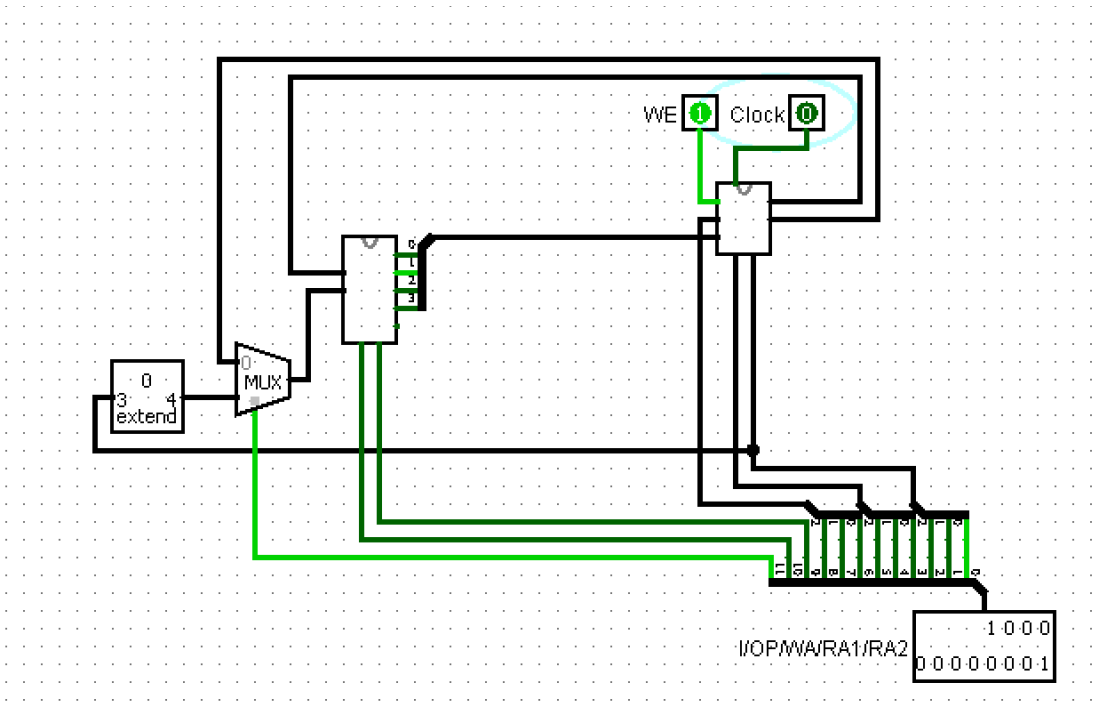
1) ADDI $0 $0 1 (100000000001): Before turning on the clock



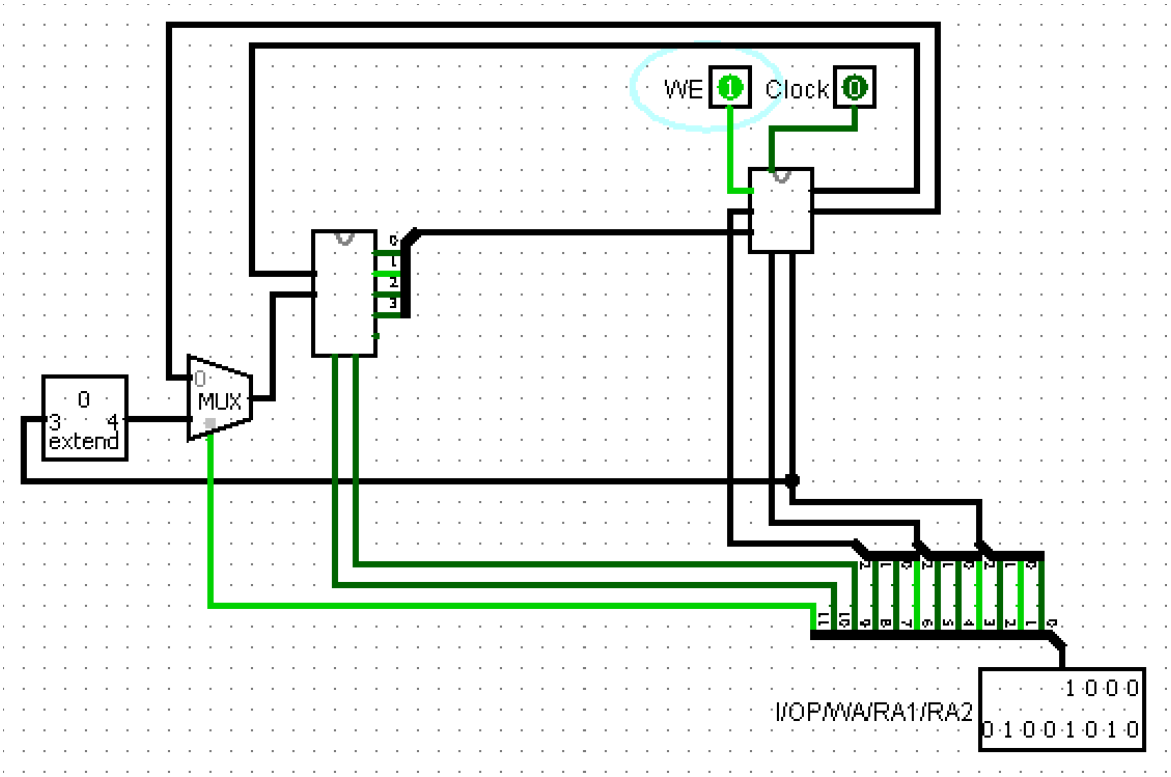
ADDI $0 $0 1 (100000000001): The clock is on



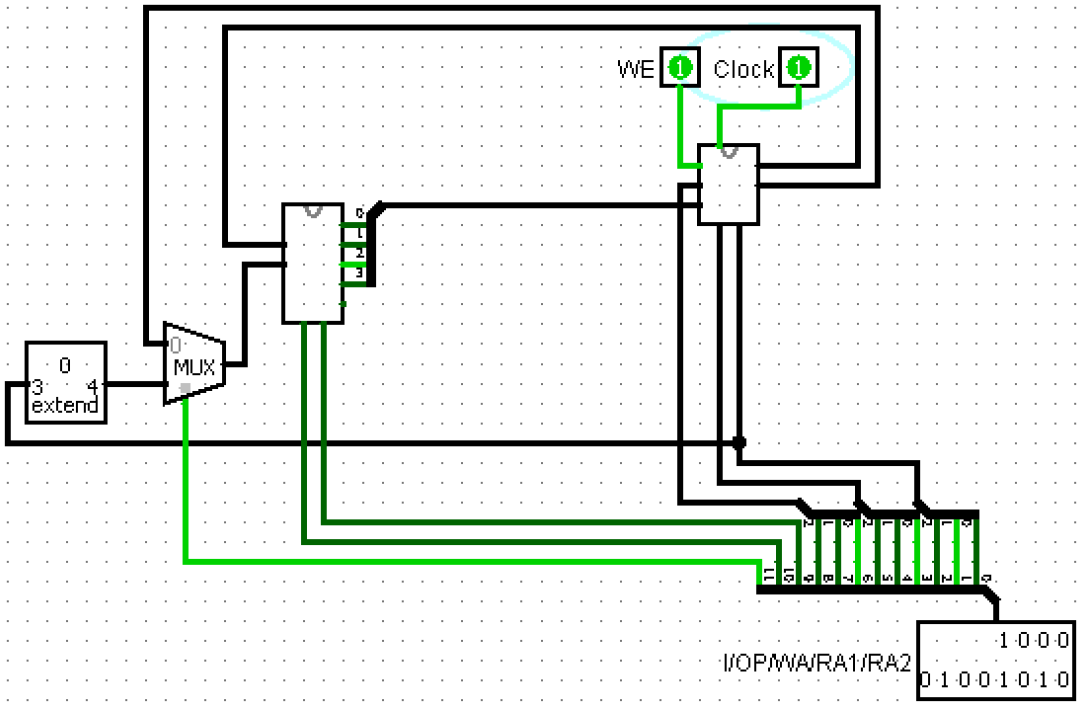
ADDI $0 $0 1 (100000000001): The clock is off.



2) ADDI $1 $1 2 (100001001010): Before turning on the clock



ADDI $1 $1 2 (100001001010): the clock is on



ADDI $1 $1 2 (100001001010): the clock is off.