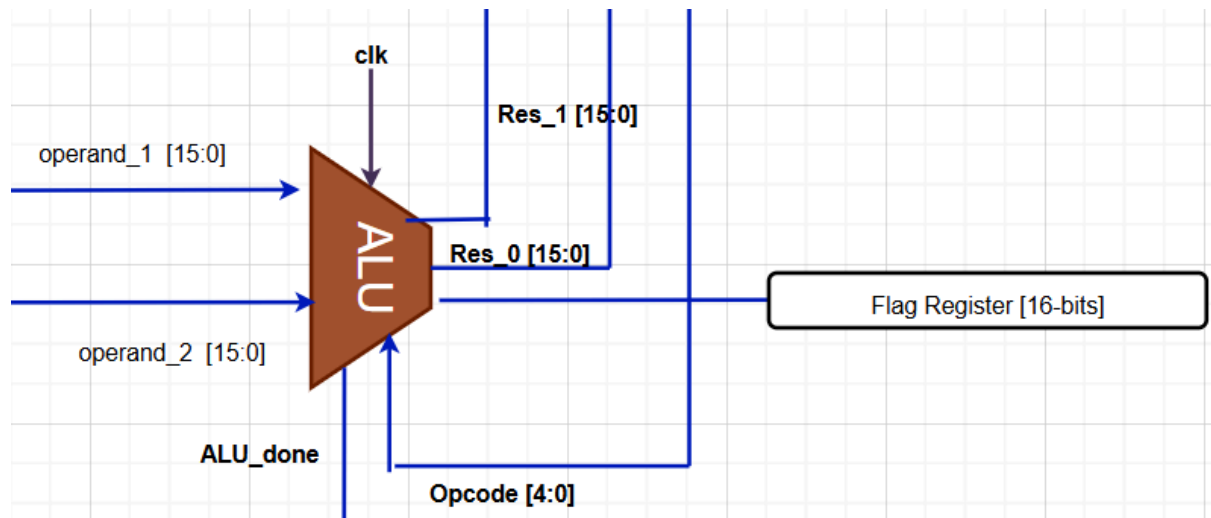


ALU



*The ALU will have a status flag (ALU_done) that will go to the control unit, to indicate the state of the operation.

Operand_1 : read_0 port of the register file

Operand_2 : read_1 port of the register file

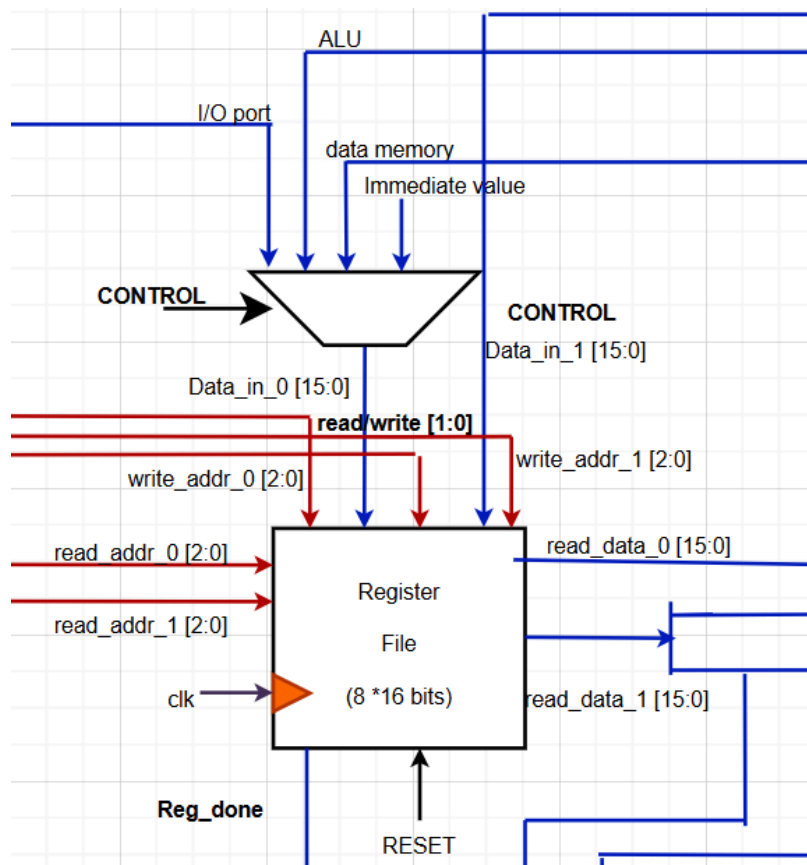
Res_0 : write_0 port of the register file

Res_1 : write_1 port of the register file

Alu_done: Goes to the control unit

*The flag register is handled by the ALU.

Register File



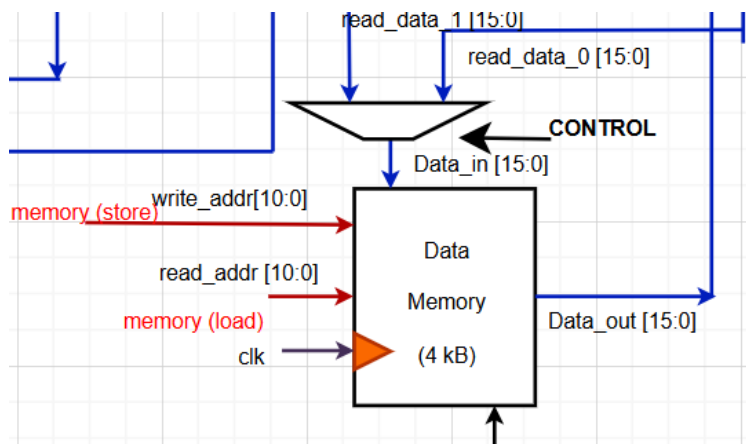
We have, 2 read ports, read_0, read_1 and two write ports, write_0 and write_1. Control signal for read/write or write_en pin.

- The read/write will be 00, for read operation.
- The read/write will be 01, for writing through write_0 port.
- The read/write will be 11, for writing through both write_0 and write_1 port.

*The Register file will have a status flag (Reg_done) that will go to the control unit, to indicate the state of the operation.

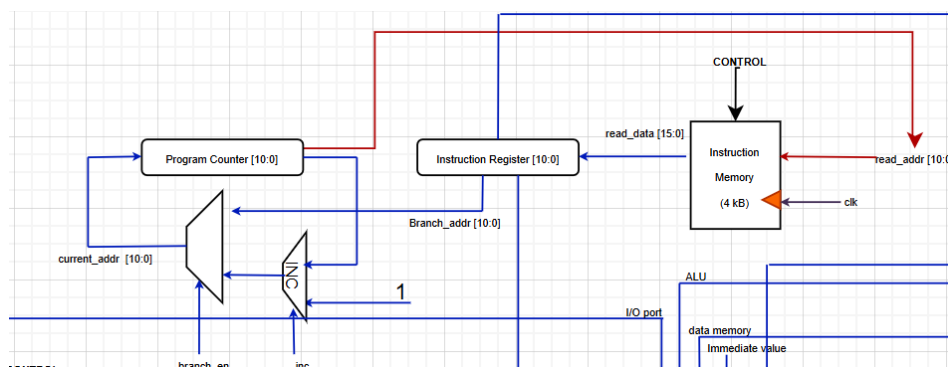
* The outputs of the register file, read_data_0 and read_data_1 will connect to either the ALU inputs or the Data memory inputs.

Data memory



- Control signal has to decide whether to accept input from read_data_0 or read_data_1 channel. The mux has these two channels of the register file as inputs.
- * The data_out connects to the mux, connected to write_0 port of the register file.
- * The reset can be removed. It is a pain.
- The inputs to the Data memory, come from the Control unit. The control unit will handle the memory instructions (load, store, load high, load low).

Instruction Memory



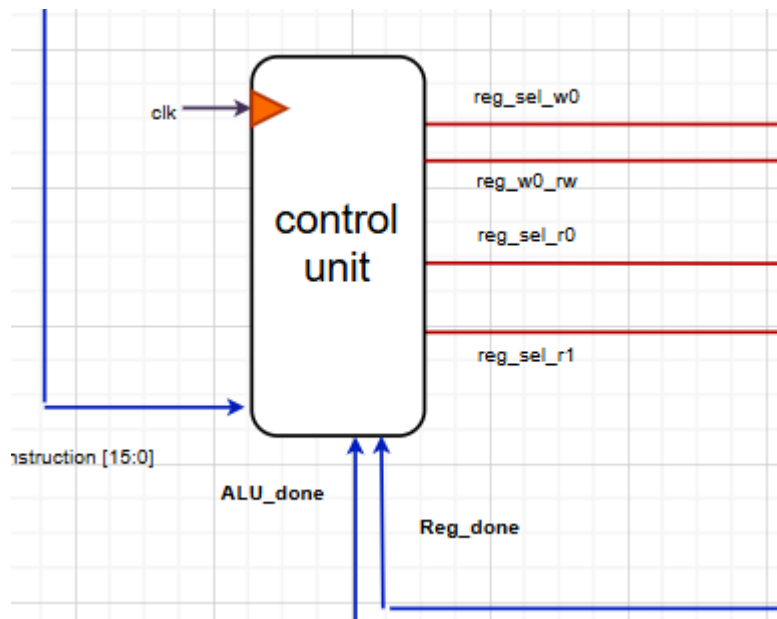
*read_data is an output port.

Control signal decides whether the next instruction (the one after the execution of the current one is over) is to be fetched or not. It should take into account the status flag of the ALU and Register file, which say whether the operation is done or not.

Read_addr will connect to the output of the program Counter.

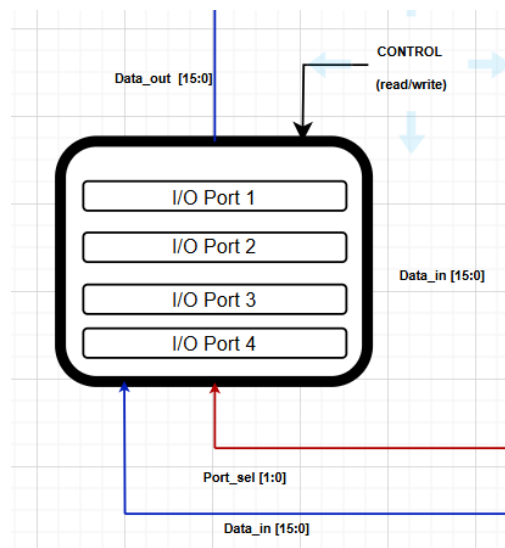
Branch_en, enables the branching. It feeds the last 11 bits of the instruction to the program counter. Inc is for regular increment.

Control unit



All memory instructions will be handled inside this. Decides when to branch, gives out the branch enable signal. Decides when to increment.

I/O Registers



We have 4 16 - bit I/O registers. Port_sel selects the I/O port, Data_in carries the data to be written to the registers. Data_out, goes to the register file.