

Exercise 5 : Data Path

In this exercise, we will create a multi-cycle MIPS data path (this figure corresponds to figure 5.28 in the 3rd edition textbook, 5.33 in the 2nd edition, but omitted in the 4th edition and beyond).

Please see [here](#) for the difference between single cycle and multi cycle.

In this issue, we will combine the 32-bit ALUs that we have created so far, as well as the circuits necessary for processors such as register files, and wire the data.

- It is not necessary to enter the control part (the part written in light color in the textbook's drawing) this time. Leave the input pins such as clock enable (ce) of fdce, control inputs of multiplexer, ALU and other input pins such as WriteEN of register file empty. Of course, simulation is impossible in this state, so after the next time, create a control circuit and wire the control system before doing.
- Please use RegFile in u-aizu library for register file. An explanation of this register file is [here](#) (mipsreg.html).
- Instruction registers, PC, ALUOut, A, B, memory data registers are all 32-bit registers. Please extend fdce of xc 4000 to 32 bits by instance duplication. For clarity, let's add comments to each register with Add (Create) -> Note.
- As for the multiplexer as well, since the input/output is 32 bits, this also needs to be instantiated to 32 bits (for the multiplexor of the register file write register number, it is expanded to 5 bits).
- For 3 input multiplexer, there are hands to make by yourself, but you can substitute for 4 input type. Let's input a 32-bit zero value for the fourth input not to be used.
- Please use mips_mem in the u-aizu library for memory. "Write data" is indicated as DI, "memory data" as DO, "read address" as ADR.
- For the clock signal, make an input pin so that a signal can be applied from the outside and connect all clocks to it. The same applies to the clear signal.
- About address of jump instruction ([jump_note.html](#))

Supplement to describe the circuit in Cadence

- Sign extension method ([sign_ext.html](#))
- 2 bit left shift method ([shift.html](#))
- Supplement for wire name ([wire_name_note.html](#))
- How to enter 4 ([input_four.html](#))