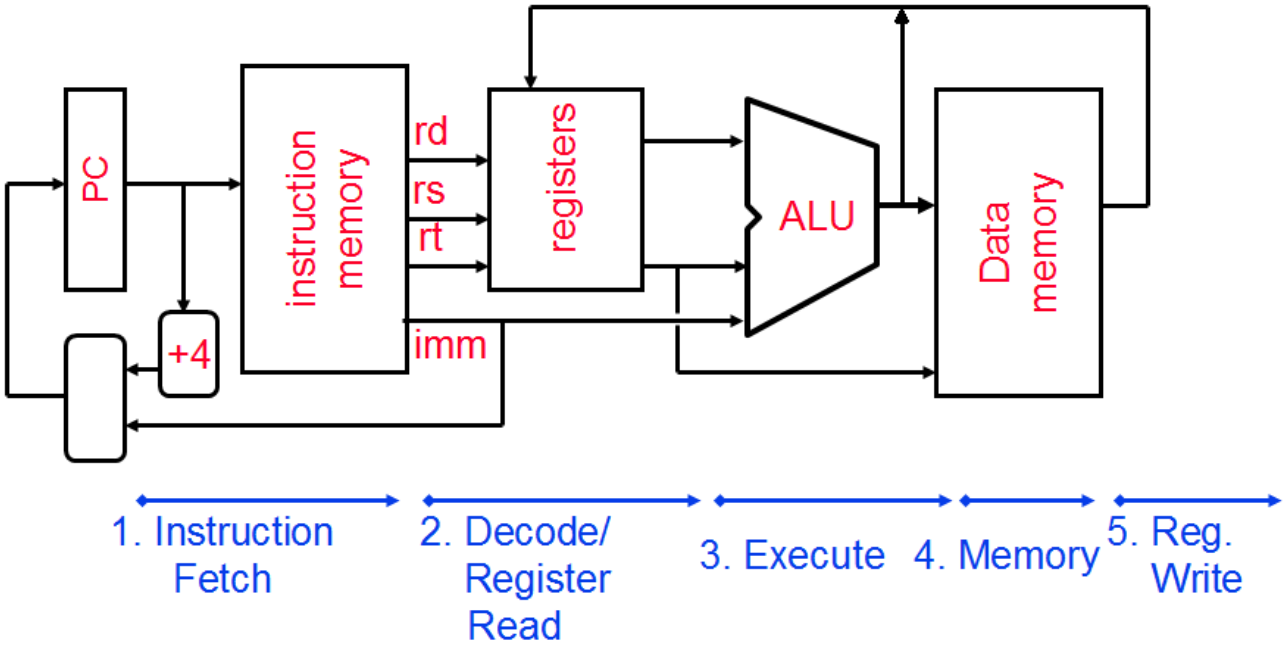
CPU Design

Here is the basic datapath as discussed in lecture, shown in simplified format.



`rd`, `rs`, and `rt` are 5 bit wires, `imm` is a 16 bit wire. All other wires are 32 bits wide.

Register Transfer Language(RTL)

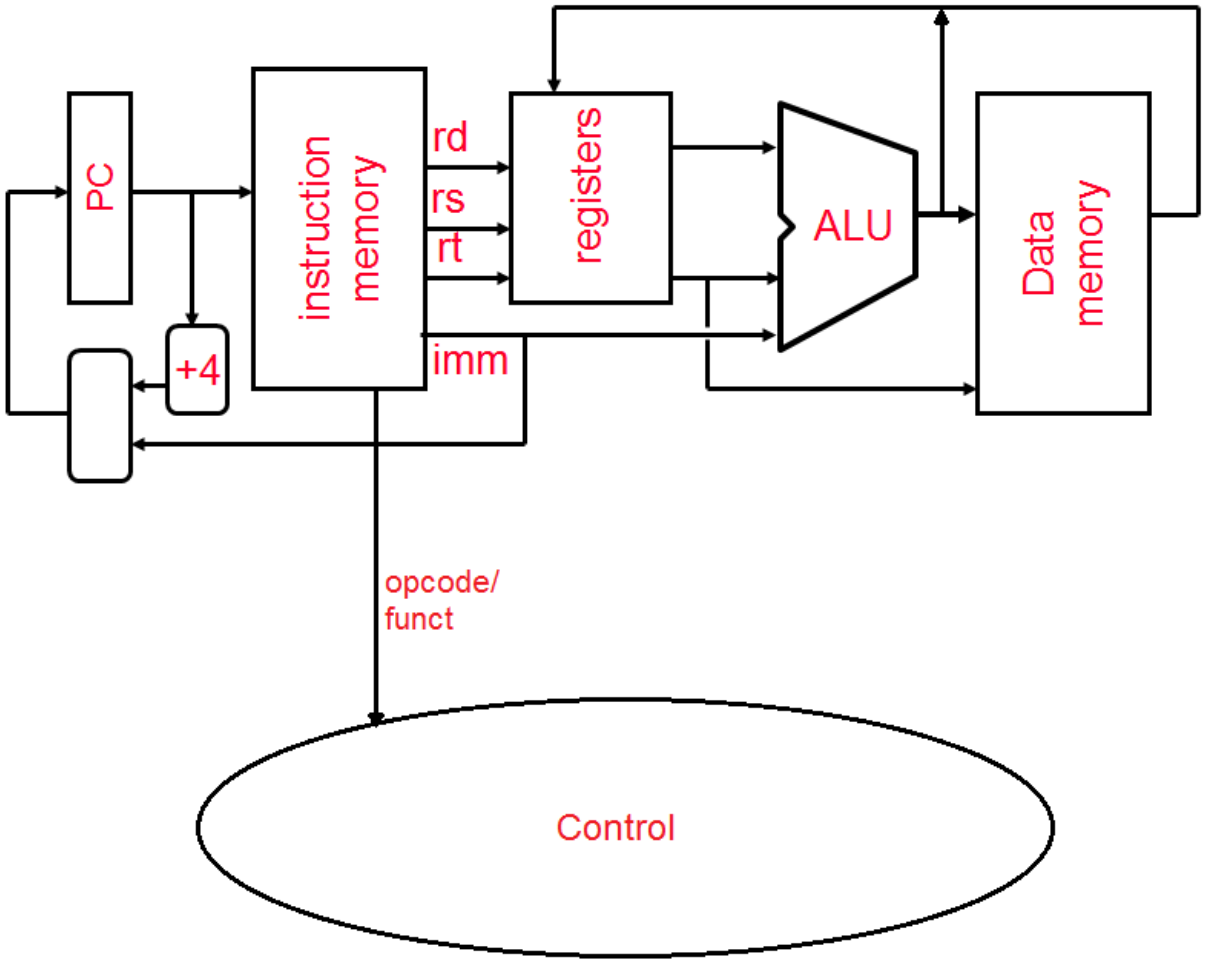
- Use to describe flow of data: *dest* ← *src*
- Each line happens in parallel (at the same time): b ← c, a ← b
- In MIPS, use R[x] for register x, and Mem[y] for memory at y. Similar to array syntax.

Exercises

For the following exercises, assume that the ALU can output an "equals" signal, which is high/on/one when its two inputs are equal.

1. Label the unlabelled wires in the diagram above, describing what data is on each line. For example, one of the outputs of the registers block could be R[rs].

2. Add control signals and missing elements (such as multiplexers) to the diagram below so that the datapath can execute the following instructions: `add, lui, sw, bne, j`.



3. Fill out the values for the control signals from question 2 (Write the names of your control signals in the second row):

Instrs.	Control Signals							
`add`								
`lui`								
`sw`								
`bne`								
`j`								

4. Suppose you wanted to add a new instruction, `beqr`, which will be used like this: `beqr $x, $y, $z` will branch to the address in `$z` if `$x` and `$y` are equal, otherwise continue to the next instruction. Show any changes that would need to be made to the datapath above to make this instruction work.