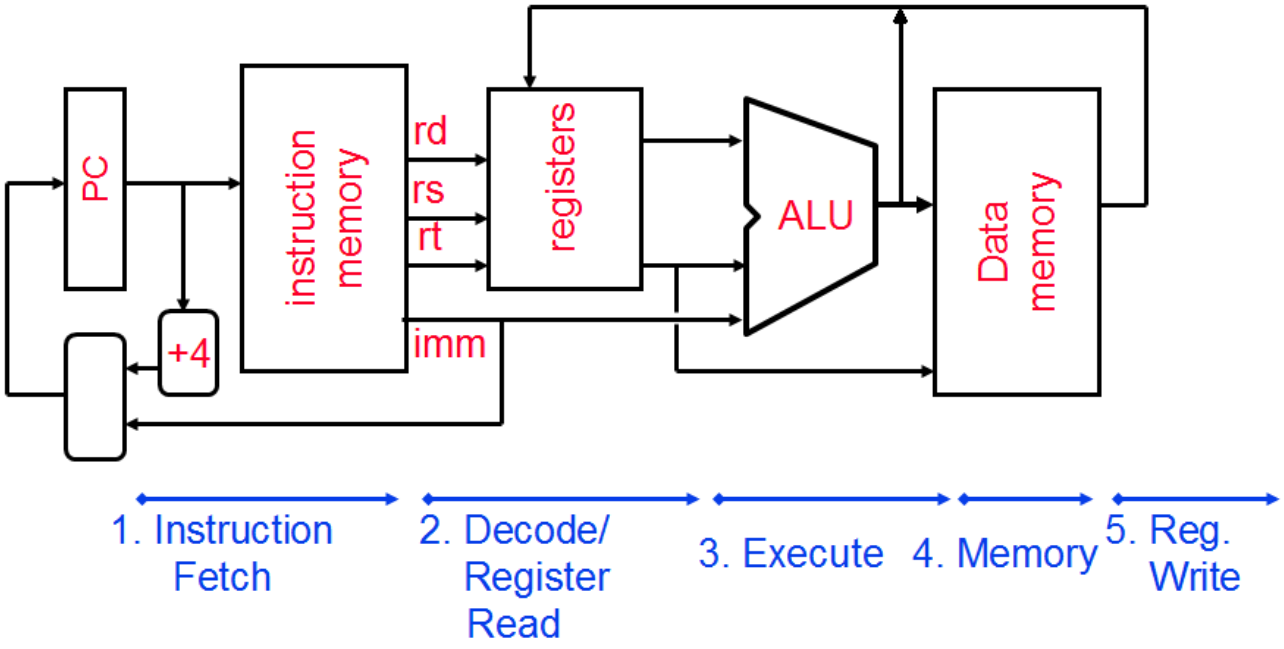
CPU Design

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)

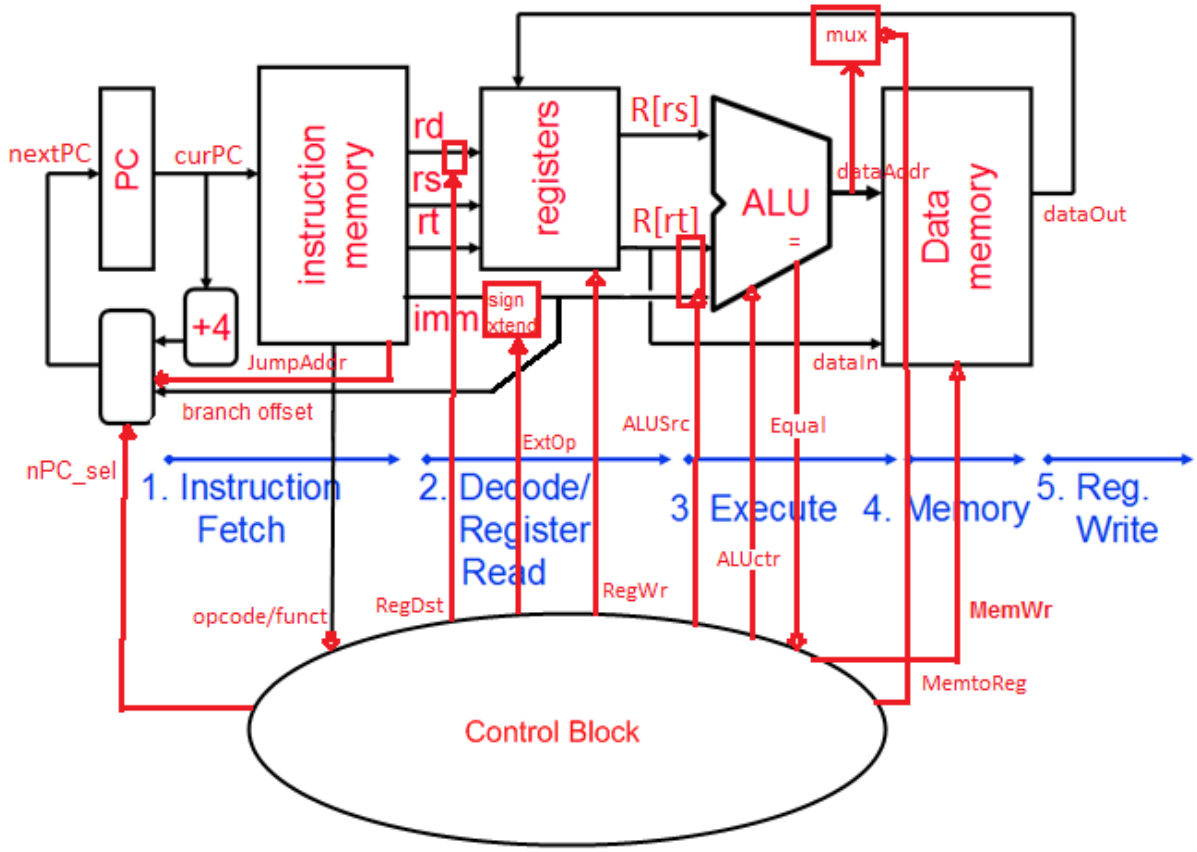
Use to describe flow of data: *dest* *src*
Each line happens in parallel (at the same time): b c, a b
In MIPS, use R[x] for register x, and Mem[y] for memory at y. Similar to array syntax.

Exercises

For the following exercises, assume that the ALU can output an equals signal, which is on 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							
	nPC_sel	RegDst	RegWr	ALUSrc	ALUctr	MemtoReg	MemWr	ExtOp
add	PC+4	rd	1	rt	add	0	0	X
lui	PC+4	rt	1	imm	shiftx16	0	0	Zero*
sw	PC+4	X	0	imm	add	X	1	Sign
bne	branch	X	0	rt	X	X	0	Sign**
j	jump	X	0	X	X	X	0	X

*Doesn't really matter since ALUctr is shiftx16

**Branch also multiplies immediate by 4 at some point

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.

The diagram is already crowded, so I will explain the changes. Since we're reading three registers at once here, we need to add a third read port to the register file, and correspondingly, a third read address port. We can reuse beq's datapath to compare two registers, rs and rt, so rd would contain the address to jump to. We would then connect R[rd] to the "address logic" module (which is essentially a mux).