Single-cycle five stage RISC-V processor emulation in C++. The program takes in a file with one byte of machine code on each line (4 lines per instruction) and outputs the values of registers a0 and a1 (x10 and x11) to stdout after all instructions are run, as per class project spec. Currently implements 10 instructions with 8 control signals (including comparison flag set by ALU) but can be expanded to implement more functionality.

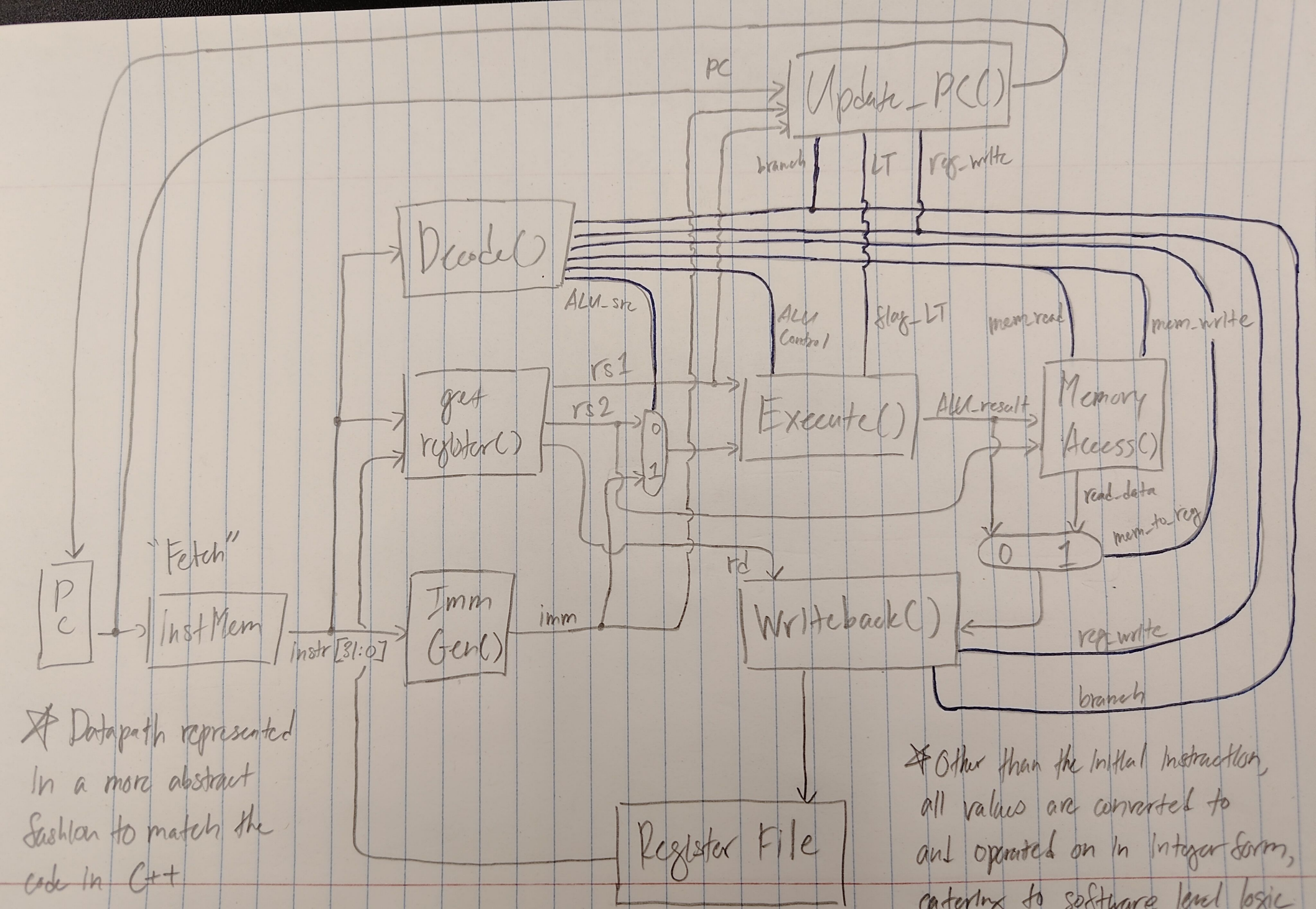
Response to Questions:

1. 13 cycles (12 + 1)
2. There are two R-type instructions, Sub (Instruction 3) and Add (Instruction 5)
3. Although the instruction memory has 14 total instructions if you go sequentially, considering branches only 12 instructions are run. Not including the zero (finish) instruction, the IPC is 1 as expected of a single-cycle processor. Including the finish, it would be 12/13.

		Control Signals						
		Branch	Mem_read	Mem_write	Mem_to_reg	Reg_write	ALU_src	ALU_control
Instructions	ADD							
	SUB							3
	ADDI							
	XOR							2
	ANDI							4
	SRA							1
	LW		1		1	1	1	
	SW			1			1	
	BLT	1						3
	JALR	1				1		

*Only non-zero control signal values are shown for better readability. All blanks are equal to 0.

Datapath + Control signal diagram is attached on the following page →



✡ Datapath represented in a more abstract fashion to match the code in C++

✡ Other than the initial instruction, all values are converted to and operated on in integer form, catering to software level logic