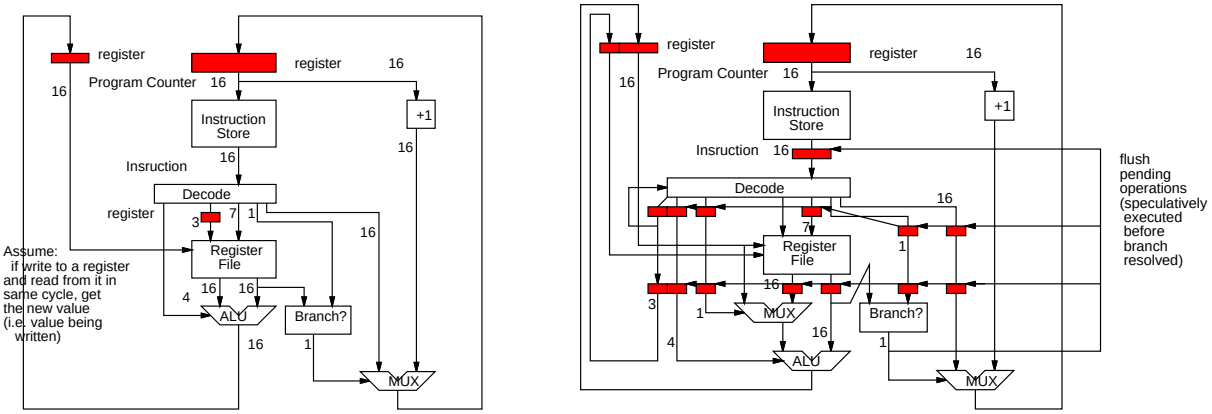
Consider the following two processor implementations



Assume:

- Writes occurring on the same cycle as reads appear to happen before the read (*i.e.*, the read gets the newly written value).
- Program Counter is 16 bits wide.
- Register File and ALU are 16 bits wide.
- There are 8 registers (R0, R1,...R7).
- Instruction store is not writable (you may consider it a ROM).

Both processors have the same semantics. The right processor achieves pipelining by:

- Bypass of register file in case where an instruction reads a value produced by the ALU on the previous cycle.
- Assuming that the branch will not be taking and speculatively issuing those instructions.
- When a branch does occur, nullifying (flushing) the instructions that should not have been issued in the branch case.

For each of the two processor:

1. How many total bits representing state?

	Left	Right
Bits		

2. How many states does each processor have?

	Left	Right
Bits		