Objectives: Learn how instruction proceed through an out-of-order (OoO) processor using a speculative version of Tomasulo's algorithm.

Assume we have a RISC-V machine that uses the speculative version of Tomasulo's algorithm. You should make the following assumptions about the machine. First, the latencies of the execution step for this machine are:

(a) FP addition and subtraction execute in 3 cycles
(b) FP division executes in 10 cycles
(c) integer operations and address addition execute in 1 cycle
(d) data can be accessed in the data cache in 1 cycle

Also, there are 2 FP adders, 1 FP multiplier/divider, 2 address adders, and 2 integer units. In a given cycle, only a single instruction can issue, only a single instruction can access the data cache, only a single instruction can write to the CDB, and only a single instruction can commit.

Fill in the table indicating the cycles in which each instruction performs each step using the approach that was described in class.

Instruction	Issues at	Executes Start	Executes End	Memory Access at	Write CDB at	Commits at
`add x3,x3,x2`	1	2	2		3	4
`fdiv.s f4,f2,f8`						
`fadd.s f6,f10,f4`						
`sub x3,x3,x4`						
`fsw f6,4(x3)`						
`flw f2,8(x3)`						
`fsub.s f10,f8,f2`						
`fadd.s f14,f6,f8`						