2. What conflicts must be resolved in running the program in Question 4 of Section 2.3 on a pipeline machine?

Suggested approach:

In question 4 of Section 2.3, we see the following:

Suppose the Vole memory cells at addresses 0xF0 to 0xF9 contain the bit patterns given in the following table:

Address	Contents
0xF0	0x20
0xF1	0xC0
0xF2	0x30
0xF3	0xF8
0xF4	0x20
0xF5	0x00
0xF6	0x30
0xF7	0xF9
0xF8	0xFF
0xF9	0xFF

If we start the Vole with its program counter containing 0xF0, what does the machine do when it reaches the instruction at address 0xF8?

Based on the definition of the instructions (see Appendix C), analyse and number the instructions seen in the above table:

Address	Contents	Instruction number	Meaning
0xF0	0x20	I1	**LOAD** the register R with the bit pattern XY. 0x20C0: load the value 0xC0 to register 0.
0xF1	0xC0		
0xF2	0x30	I2	**STORE** the bit pattern found in register R in the memory cell whose address is XY. 0x30F8: load the value at register 0 at address 0xF8
0xF3	0xF8		
0xF4	0x20	I3	**LOAD** the register R with the bit pattern XY. 0x200: load the value 0x00 to register 0.
0xF5	0x00		
0xF6	0x30	I4	**STORE** the bit pattern found in register R in the memory cell whose address is XY. 0x30F9: load the value at register 0 at address 0xF9
0xF7	0xF9		
0xF8	0xFF		Data location
0xF9	0xFF		

Register 0 is loaded with a number which is then stored onto a memory location. The operations is repeated twice.

Answer:
Given the pipeline structure considered before, this seems to be self-modifying code as it stores bytes in memory locations that will be read from memory and fetched as instructions.

They will be read before I4 will have written the second byte at address 0xF9, leading to the execution of a wrong instruction.

Therefore, address locations 0xF8 and 0xF9 must be read **after** they will be written by I2 and I4.