(25 pts)
1) Consider the following piece of old MIPS code for the unpipelined MIPS processor (machine model number 0):

```mips
loop :
   L.D F0, 0(R1) ; Load from vector A
   DIV.D F0, F0, F1 ; F1 is already initialized with “k”
   S.D F0, 0(R1) ; Store to vector A
   L.D F2, 0(R2) ; Load from vector B
   MUL.D F2, F2, F3 ; F3 is already initialized with “m”
   ADD.D F2, F0, F2
   S.D F2, 0(R2) ; Store to vector B
   DADDI R1, R1, #8
   DADDI R2, R2, #8
   DADDI R3, R3, #(-1)_10
   BNEZ R3, loop
```

Assume that this is machine model number 2 (the MIPS Int+FP pipeline).

The functional unit timings are as follows: FP operations ADD, SUB, MUL and DIV take 3, 3, 4 and 8 clock periods in EX, respectively.

The memory is an ideal memory.

Assume that R3 is two (2) initially. In which clock period, will the execution of the code be completed?

Clearly show in which clock period the execution ends as done in class, i.e. together with all the necessary forwarding and write-in-the-first-half-read-in-the-second-half cases.

Make sure you reorder the instructions so that the new code can run on this model correctly and with a minimum number of stall cycles.

Do not unroll the code!
(25 pts)
2) Consider the old original MIPS code in Question 1 again. Assume that the MIPS is implemented as the scalar hardware-speculative Tomasulo algorithm machine as discussed in class. That is, this is machine model number 4. Two (2) instructions can be committed in order per cycle provided that they are at the head of the ROB.

Assume that the functional unit timings are as given in Question 1 above; the number of FP reservation station buffers is as given in class. Assume again that R3 is two (2) initially.

The memory is an ideal memory.

In which clock period, will the execution of the code be completed? That is, what is the last clock period in which the Commit stage of the last instruction is done?

Do not show forwarding and write-in-the-first-half-read-in-the-second-half cases. Do not also show the flushed out instructions.

(25 pts)
3) Consider the old original MIPS code in Question 1 again. Assume that the MIPS is implemented as the VLIW MIPS processor: That is, this is machine model number 6. Assume that the functional unit timings are as given in Question 1 above.

Show how you would unroll and issue the instructions as done in class. Assume that there are as many iterations as needed for your unrolling.

(25 pts)
4) Consider the old original MIPS code in Question 1 again. Assume that the MIPS is implemented as the VMIPS processor: That is, this is machine model number 7.

a) Vectorize the old code in terms of VMIPS instructions as much as possible. Add comments to your code. Assume that the loop has 64 iterations and all the integer and vector registers needed have been appropriately initialized.

If you cannot vectorize a part of the original code in Question 1, indicate which part and why!

b) Show the execution timings of the vectorized code as discussed in class. Assume that there is chaining. Also assume that there are three (3) memory pipelines.