(25 pts)
1) Consider the following piece of high-level code:

\[
\text{For } (i = 0 \text{ : } i < 64 \text{ : } i++) \\
K[i] = (M[i])^2 + (N[i] * Q[i]) ;
\]

Write the corresponding piece of old MIPS code for the unpipelined MIPS processor. That is a code without delayed loads, without delayed branches, without any consideration for the latencies of functional units, etc.

Assume that vectors K, M, N and Q are double-precision FP vectors and R1, R2, R3 and R4 are initialized to point at the first element of K, M, N and Q, respectively. Also assume that R5 is initialized to \( (64)^{10} \) in the beginning. Write the code in the style of class discussions and homework assignments. Add comments to your code.

(25 pts)
2) Consider the old MIPS code obtained in Problem 1 above. 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 3.

Assume that there are enough number of CDB buses to eliminate bottlenecks. In addition, assume that there is a perfect memory with no stalls and the functional unit timings are as listed on page 304 of the Hennessy book. Another assumption is that there are enough functional units for integer instructions not to cause stalls. There are separate address and branch units. A branch instruction takes two clock periods to run if its operands are ready (IF and ID stages). Otherwise, it is “issued” to the EX stage and waits there until its operands are ready. Finally, assume that only one instruction per clock period is committed from the Reorder Buffer.

For the sake of this problem, assume that the loop has two (2) iterations. In which clock period, will the second iteration of the loop be completed? That is, what is the last clock period in which the Commit stage of an instruction from the second iteration is done? Show which instructions are flushed out of the ROB.
(25 pts)
3) Consider the old MIPS code obtained in Problem 1 above again. Assume that the MIPS is implemented as the 2-way superscalar hardware-speculative Tomasulo algorithm machine as discussed in class. That is, this is machine model number 4.

Assume that instructions are issued in order. Any pair of instructions can be issued per cycle, provided that static issuing is preserved. There are enough number of CDB buses to eliminate bottlenecks. In addition, assume that there is a perfect memory with no stalls and the functional unit timings are as listed on page 304 of the Hennessy book. Another assumption is that there are enough functional units for integer instructions not to cause stalls. There are separate address and branch units. A branch instruction takes two clock periods to run if its operands are ready (IF and ID stages). Otherwise, it is “issued” to the EX stage and waits there until its operands are ready. Finally, only one instruction can be committed per cycle.

If the loop has two (2) iterations, in which clock period will the second iteration of the loop be completed? That is, what is the last clock period in which the Commit stage of an instruction from the second iteration is done? Show which instructions are flushed out of the ROB.

(25 pts)
4) Consider the old MIPS code obtained in Problem 1 above again. The processor in this question is VMIPS. That is, this is machine model number 6.

The VMIPS vector and scalar pipeline timings are as given on page G-13 and 304 of the Hennessy book, respectively. Assume that there is no chaining. Also assume that there is one memory pipeline. Finally, assume that the VM and VLR registers are already initialized.

Rewrite the old MIPS code in terms of VMIPS instructions. Show the execution timings as discussed in class. Assume that the loop has 64 iterations.

25 pts)
5) Design a hybrid processor with superscalar and VLIW execution modes. The hybrid MIPS, SVLMIPS, is in the mode of either a 5-way statically issued superscalar issuing up to five instructions if they are ordered as memory-memory-FP-FP-Integer/Branch or a 5-way VLIW issuing 5 instructions in the order of memory-memory-FP-FP-Integer/Branch. The processor decides when to switch from one mode to the other. The code that is kept on disk is the old MIPS code.

Give a sketch of the processor as drawn in class. Design it with only one pipeline used for both modes! Mention how and when the processor decides to switch the modes, how the VLIW code is generated, its quality and where it is kept, how the hybrid system can affect the execution time, the memory pressure issue, etc. To be able to issue 5 instructions in the superscalar mode, the ID stage must be broken up into several stages, such as three: ID-1 checks dependencies between the 5 instructions and instructions in the ROB, ID-2 checks dependencies among the 5 instructions and ID-3 fetches operands from registers. Use the old MIPS code in Problem 1 as an example.