Suppose that we have implemented a (0,3) GPR machine, using a Superscalar Speculative Tomasulo (SST) architecture that executes the MIPS ISA. For each of the questions in this section, write the letter T next to the statements below that are True, and briefly explain why the others are false.

The goal here is to make grading easy for me, not to trick you with bad wording. So, if you prefer, you may write E and explain your interpretation of the statement.

1.1 The SST uses static scheduling to maximize the amount of ILP that can be exploited at execution time.

1.2 An instruction will remain in the reorder buffer of the SST until all instructions issued during the same clock cycle have committed.

1.3 The Ideal CPI of the SST is less than 1.

1.4 The SST compiler uses a branch delay slot to deal with control hazards.

1.5 Speculative execution permits the SST to issue several instructions in a single clock cycle.
Problem 2: What’s My Line? (16 pts)

Now, we turn to control hazards, and see what you’ve learned to protect yourself from them. As with the previous quiz, the correct answer(s) come the set of terms below. For each statement, identify the “speaker” by writing the Letter(s) corresponding to the term(s) for which the statement is true.

   Potential Answers:
   D (branch Delay slots)
   F (branch Folding)
   P (branch Prediction buffer)
   T (branch Target buffer without prediction)
   All (All of the above)
   None (None of the above)
   Other (fill in your Own answer)

1. I am a dynamic method to decrease the stalls caused by control hazards.
2. I am a static method to decrease the stalls caused by control hazards.
3. I eliminate all stalls due to control hazards.
4. I may consult a 2-bit saturating predictor.
5. I contain the the next instruction to be executed after a known unconditional branch instruction.
6. If a branch was just taken, I will predict “taken” the next time that the specific branch is fetched.
7. I am a cache that tracks branching history for every known branch in the program.
8. I contain the expected next PC value after the branch condition has been evaluated.