Tomasulo sheet as necessary to show the progress of the scoreboard; its fine to summarize several clock cycles on a single drawing when there is little change. Stop when the branch issues for the second time. Assume the branch was correctly predicted as taken. How many clock cycles does each loop iteration take?

Homework #5

Due Monday September 26 in 634 Soda at Noon

You may work on this assignment with your partner(s)

For these problems we will look at how a common vector loop runs on a variety of pipelined versions of DLX. The loop is the so-called SAXPY loop , and the central operation in Gaussian elimination The loop implements the vector operation Y = a∗X+Y for a vector of length 100. Here is the DLX code for the loop:

```
foo:    LD    F2    0(R1) ;load X(i)
        MULTD F4    F2    F0    ;multiply
        a*X(i)
        LD    F6    0(R2) ;load Y(i)
        ADDD  F6    F4    F6    ;add a*X(i)
        + Y(i)
        SD    0(R2) F6    ;store Y(i)
        ADDI  R1    R1    #8    ;increment X
        index
        ADDI  R2    R2    #8    ;increment Y
        index
        SGTI  R3    R1    done ;test if done
        BEQZ  R3    foo   ; loop if not done
```

For these problems, assume that the integer operations issue and complete in one clock cycle and that their results are fully bypassed. Ignore the branch delay. Assume the following latencies in clock cycles:

FP multiply	10
FP add	6
FP load/store	2
All integer operations	1

Assume the FP units are not pipelined unless the problem states otherwise.

1. For this problem use the pipeline constraints shown in Figure 4.38 (page 217) and assume a single issue pipeline. Show the number of stall cycles for each instruction and what clock cycle the instruction begins execution (i.e., enters its first EX cycle) on the first iteration of the loop. How many clock cycles does each loop iteration take?

2. Unroll the DLX code for SAXPY to make four copies of the body and schedule it for the standard DLX pipeline. When unwinding, you should optimize the code as we did in Section 5.1. Significant reordering of the code will be needed to maximize performance. What is the speedup over the original loop?

3 Using the DLX code for SAXPY above, assume a scoreboard with the functional units described in the algorithm for the hardware, **expanded to three integer functional units** (also used for load/store). Make as many copies of the attached Scoreboard sheet as necessary to show the progress of the scoreboard; its fine to summarize several clock cycles on a single drawing when there is little change. Stop when the branch issues for the second time. Assume the branch was correctly predicted taken and took one cycle. How many clock cycles does each loop iteration take? You may ignore any register port/bus conflicts.

4. Use the DLX code for SAXPY above. Assume Tomasulo's algorithm for the hardware using the functional-unit count shown in Section 5.2 Make as many copies of the attached