Homework Chapter 6 Part 1
Computer Organization Spring 2003

Due date: Friday, 3/7/03

Problem 1 (15 points)

Suppose you have a pipelined machine with a $k$ stage pipeline and a program with $n$ instructions whose dependencies are such that the pipeline does not stall.

If each stage of the pipe takes 1 cycle what is the speedup gained by pipelining compared to execution of the program on the same machine without exploiting the pipeline when $k = 5$ and $n = 1000$? What about when $k = 5$ and $n = 15$? (You may assume that there is no extra overhead when operating the machine in a non-pipelined fashion.) Write a formula for the speedup in terms of $k$ and $n$.

Problem 2 (20 points)

Consider the following code.

```
or $5,$6,$7
lw $8,20($2)
sub $10,$9,$8
```

Draw a diagram like that of Figure 6.44 that shows any dependencies that are present in the code. Then, draw a diagram like that of Figure 6.45 that shows how the code would execute on a pipeline with forwarding. Be sure to identify any stalls or forwarding that must be done. How many cycles does the code require to complete?

Problem 3 (35 points)

Suppose the `bne` instruction is a branch on not equal. Rearrange the code below to minimize completion time on the basic MIPS pipeline where the implementation uses a single branch delay slot.

Your final code should minimize the effect of stalls, if any, and use the delay slot. Note that since you are playing the role of compiler in this question you are free to alter any constants that appear in the instructions, e.g., address offsets, and to change the order of instructions.

When giving your answer be sure to identify all stalls that you remove, why they exist in the original code, and why they do not in your altered code. You should also explain any other alteration of the code and why your choice of instruction to fill the delay slot is appropriate.
Problem 4 (30 points)

Consider Figure 6.25 on page 467 and specifically the pipeline registers that have been added to the datapath. For each pipeline register, determine the data that is loaded into each of its fields from the previous stage and how those bits are used in the next stage. Label each field and determine its length in bits. (In your answer please list the fields and lengths for each register starting at the top of the register.)

For each field of a pipeline register also indicate how the bits are used in the stage of the pipeline for which the register provides input.

Note that you do not need to worry about control signals in the pipeline registers. Consider only datapath signals.