Computer Organization and Structure

Homework #5
Due: 2003/12/30

1. (20%)
Obviously either the load or the addi must occupy the branch delay slot. We can’t just put the Addi into the slot because the branch instruction needs to compare $3 with register $4 and the addi instruction changes $3. In order to move the load into the branch delay slot, we must realize that $3 will have changed. If you like, you can think of this as a two-step transformation. First, we rewrite the code as follows:

```
Loop: addi $3, $3, 4
    lw $2, 96($3)
    beq $3, $4, Loop
```

Then we can move the load into the branch delay slot:

```
Loop: addi $3, $3, 4
    beq $3, $4, Loop
    lw $2, 96($3) # branch delay slot
```

2. (20%)
The second instruction is dependent upon the first ($2). The third instruction is dependent upon the first ($2). The fourth instruction is dependent upon the first ($2) and second ($4). All of these dependencies will be resolved via forwarding.

3. (20%)
Consider each register:

- IF/ID holds PC+4 (32 bits) and the instruction (32 bits) for a total of 64 bits.
- ID/EX holds PC+4 (32 bits), Read data 1 and 2 (32 bits each), the sign-extended data (32 bits), and two possible destinations (5 bits each) for a total of 138 bits.
- EX/MEM holds PC target if branch (32 bits), Zero (1 bit), ALU Result (32 bits), Read data 2 (32 bits), and a destination register (5 bits) for a total of 102 bits.
- MEM/WB holds the data coming out of memory (32 bits), ALU Result (32 bits), and the destination register (5 bits) for a total of 69 bits.

4. (30%)

a.
In the fifth cycle of execution, register $1 will be written and registers $11 and $12 will be read.

b.
The forwarding unit is seeing if it needs to forward. It is looking at the instructions in the fourth and fifth stages and checking to see whether they intend to write to the register file and whether the register written is being used as an ALU input. Thus, it is comparing $8 = 4? \ 8 = 1? \ 9 = 4? \ 9 = 1?$

c.
The hazard detection unit is checking to see whether the instruction in the ALU stage is a lw instruction and whether the instruction in the ID stage is reading the register that the lw will be writing. If it is, it needs to stall. If there is a lw instruction, it checks to see whether the destination is register 11 or 12 (the registers being read).

5. (10%)
The situation is similar in that a read is occurring after a write. The situation is dissimilar in that the read occurs much later (in the fourth cycle for load instructions vs. in the second cycle for add instructions) and that the write occurs earlier (in the fourth cycle instead of the fifth cycle). For these reasons, there is no problem. Another way of looking at it is that both the read and write occur in the same cycle (for memory access) and thus we can’t possibly have a hazard!