Homework Quiz (HW #5)
November 7, 2001
CS152 Computer Architecture and Engineering

This quiz covers one of the problems from homework #5.
Good Luck!

<table>
<thead>
<tr>
<th>Your Name:</th>
</tr>
</thead>
<tbody>
<tr>
<td>SID Number:</td>
</tr>
<tr>
<td>Discussion Section:</td>
</tr>
<tr>
<td>Total:</td>
</tr>
</tbody>
</table>
In problems 6.26 and 6.27, you were asked to consider a pipeline which does not support a delayed branch. Figure 6.51 from one of the printings of the book is reproduced on the previous page.

1. In this datapath, how many instructions must be “flushed” out of the pipeline when a branch is taken? Explain.

   *One. The control can know that a branch is taken when the instruction is in the ID stage. Since an instruction is being fetched in parallel, in the IF stage, that instruction must be flushed.*

2. What must be in the “Control” oval in order to support flushing (i.e. when does it decide to assert IF.Flush)? What exactly does the “IF.Flush” signal do?

   *It must prevent the instruction in the IF stage from committing its state. This means the instruction is neither allowed to write to the data memory nor the register file.*

3. Consider the following instruction sequence:

   ```
   sub $2, $4, $5
   beq $2, $3, somewhere
   ```

   Why doesn’t this code sequence work properly on this hardware (this is a bug in the book!)?

   *This pipeline does not forward back to the ID stage. Therefore, the equal comparator won’t receive the updated value of register $2.*

4. Can you fix this problem without adding hardware (i.e. only moving hardware around)? Explain. How many instructions must be “flushed” on a taken branch now?

   *There are two possible answers to this question. You can either forward to the ID stage by putting muxes in front of the comparator, or you can move the comparator into the EXE stage. In the latter case, you would have to flush two instructions each time a branch is taken.*