CS 2734, Computer Organization II
Fall Semester, 2003
Second Examination

1. For this problem, you are to use a xerox of Figure 5.29. (Final datapath for the *single*-cycle implementation of MIPS.) You will be tracing through the path of the **`lw`** instruction on this single-cycle model. You should use a highlighter to trace the path the instruction and associated data takes through the diagram. (Do *not* show data traveling to "dead-end" components, which will eventually have no effect.) Then write in the values for the *relevant* control signals. (Do *not* give control signals that serve to keep "dead-end" paths from having an effect.) **(30)**

   Use the following specific instruction:

   ```
   lw $t5, 92($t1)
   ```

   or in machine language form:

   ```
   0x8d2d005c                              (in hexadecimal)
   100011 01001 01101 00000 00001 011100 (fields in binary)
       35     9    13                 92 (fields in decimal)
   ```

   Don't forget the handling of the PC by this instruction. Show what values are traveling along the different lines, assuming the following initial values:

   (a) **`$t1`** and **`$t5`** are registers numbers **`9`** and **`13`** (decimal), respectively.

   (b) The contents of register **`9`** is **`200`** (decimal).

   (c) The PC has value **`16`**.

---

2. For this problem, you are to use a xerox of Figure 5.33. (Final datapath for the *multi*-cycle implementation of MIPS.) You will be tracing through the path of the **`beq`** instruction on this multi-cycle model. You should use several colors of highlighters to trace the paths the instruction and associated data takes through the diagram. (Or you can trace these paths using more than one diagram.) Do *not* show data traveling to "dead-end" components, which will eventually have no effect. **(30)**

   For this diagram, you do *not* need to give the values of control signals.

   Below the diagram, or in some other way, carefully identify *which cycle* (or step) of handling the instruction belongs to each part of the highlighted datapath (just for data, not control). Thus you should identify *Cycle 1*, *Cycle 2*, *Cycle 3*, and perhaps *Cycle 4* and *Cycle 5* (if the instruction uses Cycles 4 and 5).

   Use the following specific instruction:

```
beq   $t2, $t5, LabelA
```

or in machine language form:

```
0x114d0004                              (in hexadecimal)
000100 01010 01101 00000 00000 000100 (fields in binary)
     4    10    13                  4 (fields in decimal)
```

Start at the left side, showing the PC value coming in, and assume this instruction is read from the Instruction Memory. Show what values are traveling along the different lines, assuming the following initial values:

(a) **`$t2`** and **`$t5`** are register numbers **`10`** and **`13`** (decimal), respectively.

(b) Assume that the contents of each of these registers is **`5234`**, so you should assume that the branch is taken.

(c) Assume the PC has value **`20`** (decimal) initially. On the proper line, give the *final* PC value, assuming the branch is taken. Don't forget to highlight the parts related to the PC as well as the rest of the instruction. *Be sure to identify the different cycles.*

---

3. This problem deals with the *Hamming Code*. **(20)**

(a) Suppose you have the following six *data bits*: **`0 1 1 0 0 1`**. Insert the proper values for the check bits in the correct positions. (Do not use position **`0`**.)

(b) Suppose the bit in position **`6`** is transmitted in error. Show how the Hamming code will be able to correct this error.

---

4. This question is concerned with *exceptions* and *interrupts*, as described in the text for the multi-cycle implementation. **(20)**

(a) Does MIPS use vectored interrupts? (Just "Yes" or "No.")

(b) Suppose an *undefined instruction* exception occurs. Itentify and describe each of the following additions to the hardware that are needed to handle this exception:

i. EPC. (What is it and what is it for?)

ii. Cause. (What is it and what is it for?)

iii. Control lines going to EPC and to Cause. (What are they for?)

iv. Extra option on the line leading back to the PC. (What does it do?)

v. Additions to the finite state machine for control. (Describe the additions in general terms, without saying exactly what they are.)