**EE 4720** **Homework 4** **Due: 12 March 1997**

Consider a DLX implementation fabricated based on the solution to homework 3 part 3 given in http://www.ee.lsu.edu/koppel/ee4720/1997/hw03_sol.html, and which also includes register forwarding connections so that only load instructions would stall because of RAW hazards.

**Problem 1:** Consider the fragment below.

```
 PM R4, 0xfd, 0xfa     ! Note: 0xfd = 11111101 0xfa = 11111010
 ADD R3, R1, R2
 LW  R7, 0(R20)
 SUB R6, R7, R8
```

Using a timing diagram (the table with IF, ID, etc. entries for each instruction) show what would happen when the fragment is executed with R4 both equal to zero and not equal to zero. Repeat the problem for the fragment below, in which the contents of R5 is zero.

```
 PM R4, 0x13, 0xfc
LOOP:
 ADD R3, R1, R2
 BEQZ R5, LOOP
 SUB R6, R7, R8
```

**Problem 2:** Suppose a DLX implementation was fabricated based on the solution to homework 3 part 3 given in http://www.ee.lsu.edu/koppel/ee4720/1997/hw03_sol.html, and also included register forwarding connections so that only load instructions would have RAW hazards. Users of this chip are advised never to place a control-transfer or load instruction in the eight instructions following a PM unless the corresponding bits of the execution masks for the instruction and those following are one. For example

```
 PM R4, 0xfc, 0xff      ! Note: 0xfe = 11111100 (binary)
 ADD R3, R1, R2
 SUB R6, R7, R8
 BNEQ R5, SOMEWHERE
```

is okay because the mask bits will be one for the branch and following instructions. Anyone violating these rules would have to answer to Mike. Since not everyone realizes that this threat has teeth, an illegal instruction exception (non-precise) should be generated when the rule is violated. Modify the solution to HW 3 so that a signal is generated, `PM_VIOL`, if the rule is violated. How difficult would it be to make this exception precise?

**Problem 3:** Suppose we don't want to proscribe loads and branches in the scope of a PM. Modify the solution to homework 3 problem 3 so that such loads and branches are handled properly. Note that if a branch is taken, the remaining execution mask bits would apply to the instructions at the target.