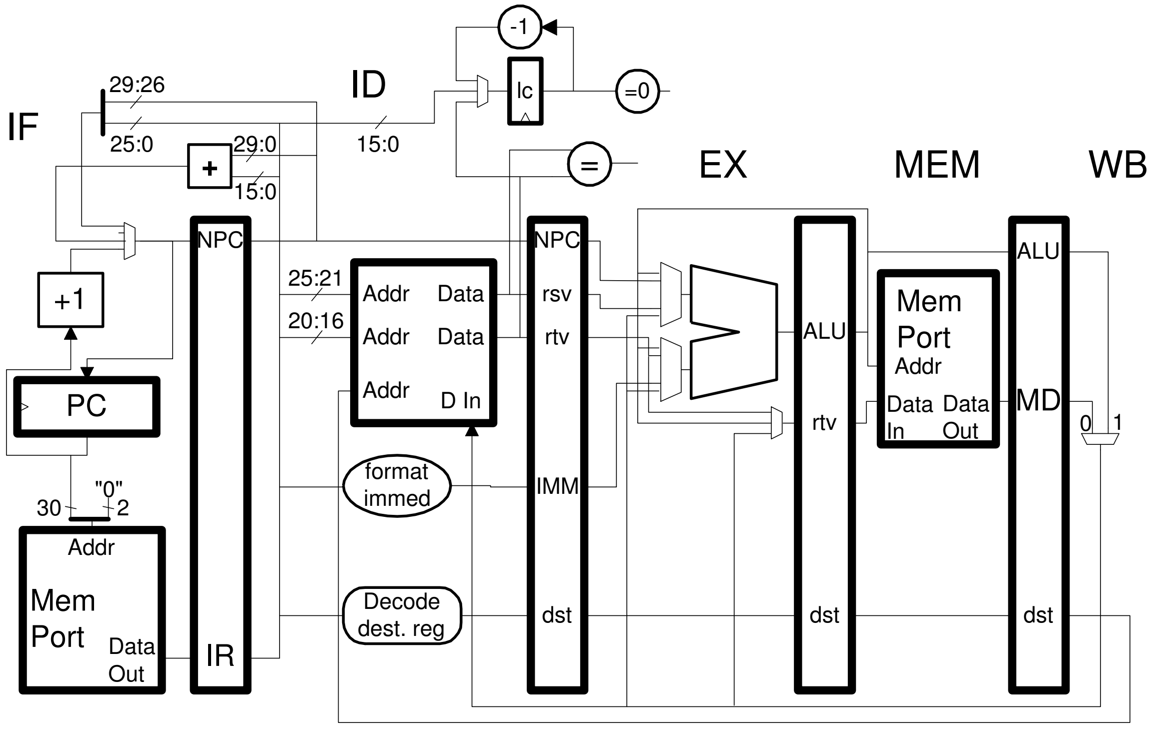
LSU EE 4720 Homework 5 Due: 30 November 2005

Problem 1: The execution of a new MIPS instruction `blcz TARG`, branch unless loop count register is zero, will result in a delayed control transfer to `TARG` unless the contents of a new register, `lc`, is zero; the target is computed in the same way as ordinary branch instructions. Execution of `blcz` will also decrement `lc` unless it is already zero. The `lc` register is loaded by two new instructions `mtlc` and `mtlci`. The code below uses some of the new instructions and the diagram shows a possible implementation.

```
 mtlc 100         # Load lc register for a 101-iteration loop
LOOP:
 sw r0, 0(r1)
 blcz LOOP        # If lc is not zero branch to LOOP, lc = lc - 1.
 addiu r1, r1, 4
```



(*a*) Re-write the code above using ordinary MIPS instructions and write it so that the loop uses as few instructions as possible. *Hint: A three-instruction loop body is possible.*

(*b*) Using pipeline execution diagrams determine the speed of the sample program and your program from the previous part. Only use bypass paths that have been provided.

(*c*) Unless the control logic is appropriately modified the implementation above may not realize precise exceptions for all integer instructions. In fact, the problem could occur in the example program. Explain what the problem is and show a pipeline execution diagram in which the control logic insures that execution proceeds so that exceptions will be precise. *Hint 1: The exception does not occur in any of the new instructions. Hint 2: One of the two remaining instructions in the example can not raise an exception so it must be the other one.*

(*d*) Modify the implementation so that precise exceptions are again possible for all integer instructions (while retaining the loop count instructions) without sacrificing performance.