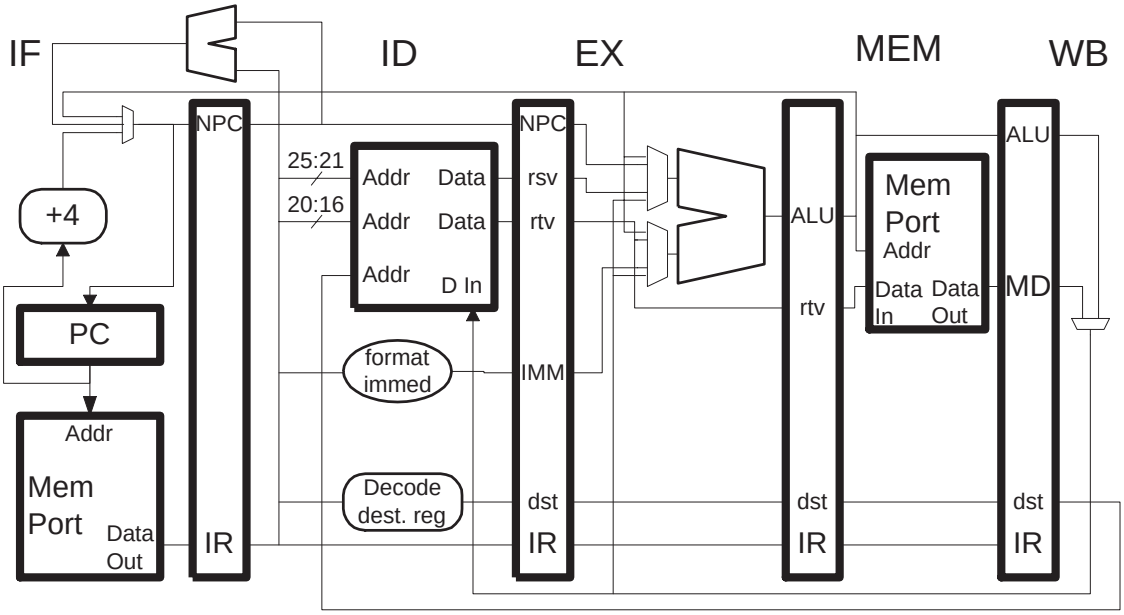
LSU EE 4720 **Homework 4** **Due: 22 March 2004**

Problem 1: Suppose code like the memory copy program above (from Homework 3) appears frequently enough in the execution of programs so that new instructions should be added to the ISA to allow improved execution. (It does and they have been.)

Following the points below devise new instruction(s) that can be used to write a new memory copy loop that would execute more efficiently than is possible with existing MIPS-I instructions. A goal is to copy at the rate of two bytes per cycle. See the subparts after the bulleted points below.

```
LOOP:
 lw $t0, 0($a0)
 sw 0($a1), $t0
 addi $a0, $a0, 4
 bne $a0, $a2 LOOP
 addi $a1, $a1, 4
```



- The instructions must use the existing MIPS formats.
- An instruction can do more than one thing (as long as it follows the points below). For example, an instruction that does more than one thing is a post-increment load. To reach the two bytes / cycle limit one might need to combine a branch with something.
- The instructions cannot use implicit registers. (A register is implicit if it does not appear in the encoded instruction. For example, register 31 is implicit in the `jal` instruction.)
- To achieve two bytes per cycle the instructions might need to do something unusual with operands. Please ask if you're not sure if something is too unusual.
- As with all other ordinary instructions, the new instructions must advance one stage per cycle (unless stalled, if so they would sit idle).

- The modified pipeline must still use the same memory port and no new memory ports can be added.
- Modifications such as bypass paths can be added to speed the instructions.

(*a*) Show an example of each new instruction and show how it is coded.

(*b*) Show how the instructions would be implemented on the pipeline.

(*c*) Write a memory copy program using the new instructions.

(*d*) Show a pipeline execution diagram for the memory copy code.

(*e*) A two bytes per cycle solution would require doing something interesting for the branch. Explain what that is and show a pipeline execution diagram for the memory copy loop finishing a copy (where the interesting stuff would be done).