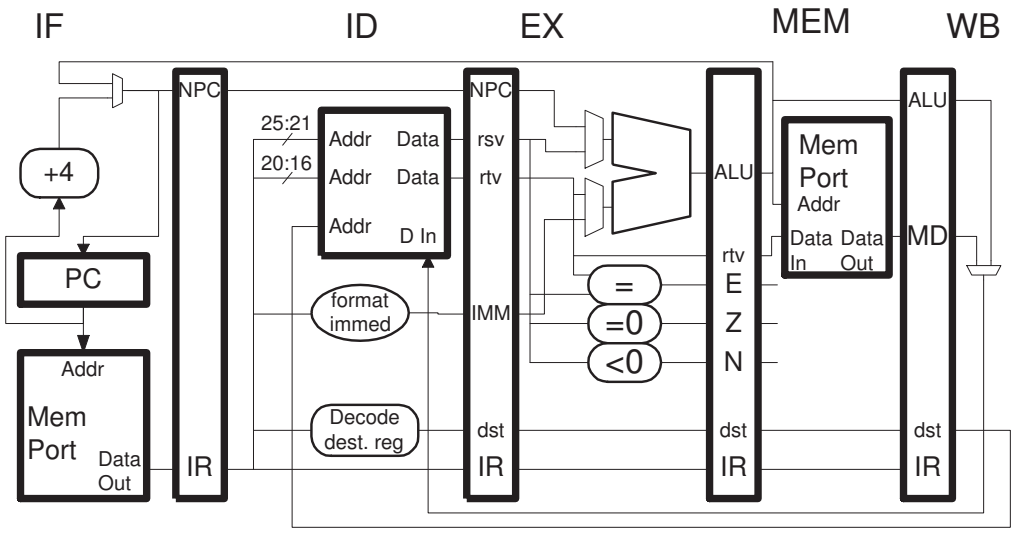
LSU EE 4720 **Homework 1** **Due: 6 February 2013**

Problem 1: The MIPS code below executes on the illustrated implementation. The loop iterates for many cycles. The register file bypasses data from the write ports to the read port in the same cycle.



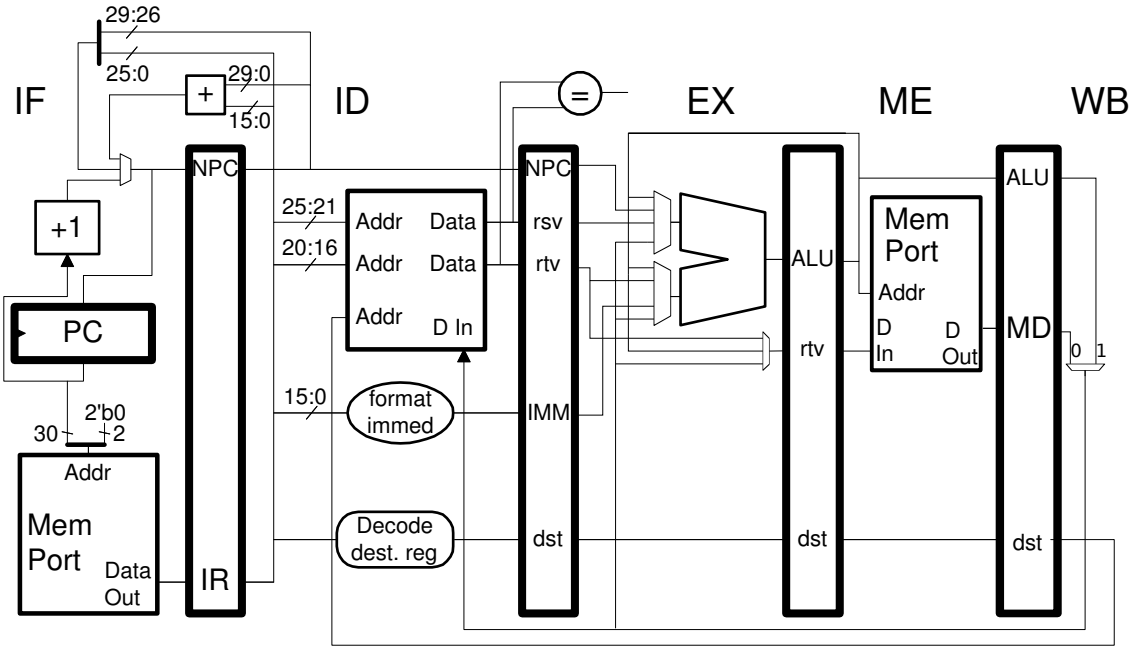

```
LOOP:
 lw r2, 0(r4)
 slt r1, r2, r7
 bne r1, r0 LOOP
 addi r4, r4, 4
 sw r4, 0(r6)
 jr r31
```

(*a*) Show the execution of the code above on the illustrated implementation up to and including the first instruction of the second iteration.

- Carefully check the code for dependencies.
- Be sure to stall when necessary.
- Pay careful attention to the timing of the fetch of the branch target.

(*b*) Compute the CPI for a large number of iterations.

Problem 2: The code fragment below is the same as the one used in the last problem, but the implementation is different (most would say better).



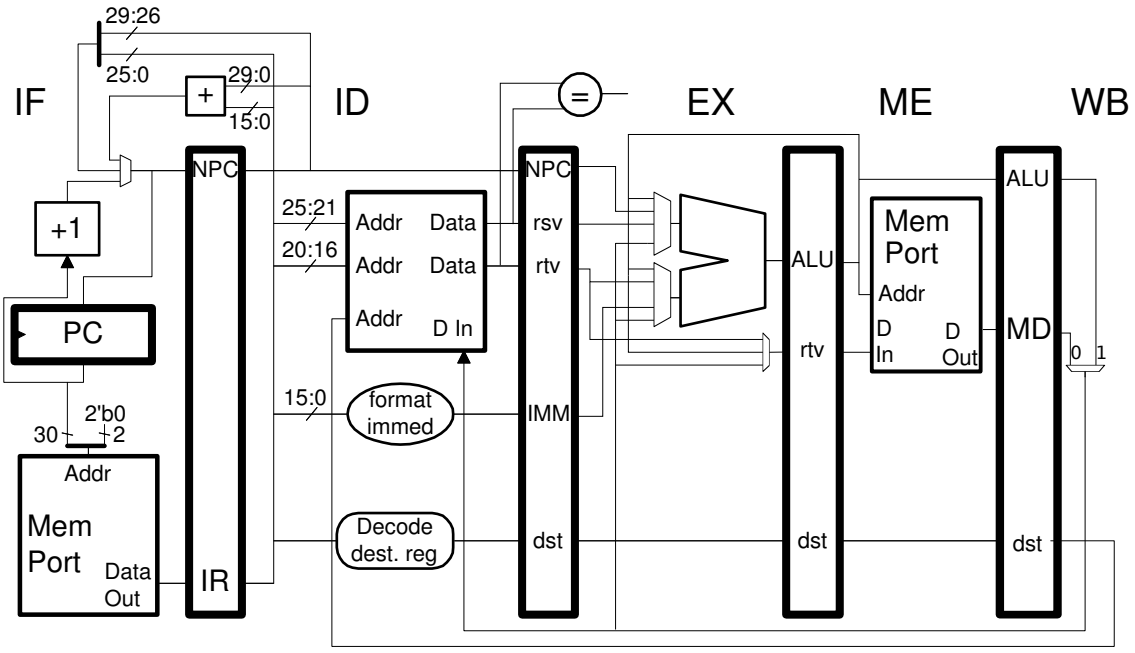

```
LOOP:
 lw r2, 0(r4)
 slt r1, r2, r7
 bne r1, r0 LOOP
 addi r4, r4, 4
 sw r4, 0(r6)
 jr r31
```

(*a*) Show the execution of the code on this new implementation.

- There will still be stalls due to dependencies, though fewer than before.

(*b*) Compute the CPI for a large number of iterations.

Problem 3: Consider once again the code fragment from the previous two problems, and the implementation from the previous problem. In this problem consider a MIPS implementation that executes a `blt` instruction, an instruction that is not part of MIPS. With such an instruction the code fragment from the previous problems can be shortened, and one would hope that the code would take less time to run. In this problem rather than hope we'll figure it out.




```
LOOP:
 lw r2, 0(r4)
 blt r2, r7  LOOP
 addi r4, r4, 4
 sw r4, 0(r6)
 jr r31
```

(*a*) Add the additional datapath (non-control) hardware needed to execute `blt`. *Hint: Just add one unit and a few wires.*

(*b*) Show the execution of the code on the illustrated implementation up until the second fetch of `lw`.

(*c*) As we discussed in class, doing a magnitude comparison in `ID` might stretch the critical path, forcing a reduction in clock frequency. Suppose the clock frequency without `blt` is 1 GHz. At what clock frequency will the `blt` implementation, the one in this problem, be just as fast as the implementation from the prior problem on their respective code fragments?

- Be sure to pick a sensible meaning of *just as fast*. **Do not** define just-as-fast in terms of CPI.

(*d*) Explain why the code fragments in these problems might exaggerate the benefit of the `blt` instruction.