

Problem 1: Without looking at the solution solve Spring 2002 Homework 2 Problem 2 parts a-c. Then, look at the solution and assign yourself a grade in the range [0,1].

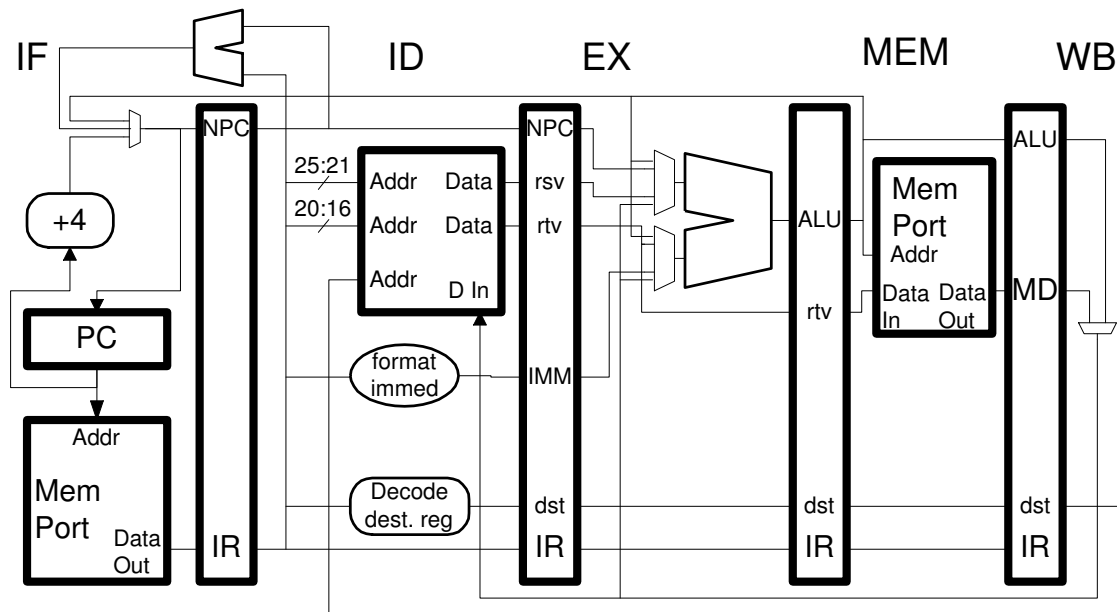
Problem 2: If the value in register r2 is not aligned (a multiple of four) the `lw` in the MIPS code below will not complete.

```
lw r1, 0(r2)
```

(a) Re-write the code so that `r1` is loaded with the word at the address in `r2`, whether or not it is aligned. For this part do not use instructions `lwl` and `lwr` (see the next part).

(b) Re-write the code, but this time use MIPS instructions `lwl` and `lwr`. *Hint: These instructions were not covered in class, try looking them up in the MIPS architecture manual conveniently linked to the <http://www.ece.lsu.edu/ee4720/reference.html> page..*

Problem 3: Consider how the `lwl` and `lwr` instructions might be added to the implementation below. There are two pieces of hardware that with minor modification would be able to merge the sub-words in a reasonable solution. Alternatively, a new piece of hardware to perform the merge can be added.



(a) Show how the hardware can be modified.

- If your solution relies on an existing component to perform the merge indicate which component and why that can be easily modified to do the merge.
- As with other MIPS instructions `lwl` must spend one cycle in each stage (except when stalling).

(b) Show how the code below would execute on your solution. **Pay attention to dependencies.** Feel free to propose an alternate solution to reduce the number of stalls.

```
add r1, r3, r4
lwl r1, 0(r2)
sub r5, r1, r6
```

Problem 4: Consider these options for handling unaligned loads in the MIPS ISA which might have been debated while MIPS was being developed.

- *Option Lean:* All load addresses aligned. No special instructions for unaligned loads (*e.g.*, no `lw1` or `lwr`).
- *Option Real:* All load addresses aligned. Special instructions for unaligned loads (*e.g.*, `lw1` or `lwr`).
- *Option Nice:* Load addresses do not have to be aligned, however warn programmers that loads of unaligned addresses may take longer in some implementations.

(a) For each option provide an advantage and a disadvantage.

(b) What kind of data would be needed to choose between these options? Consider both software and hardware data, be reasonably specific.

(c) Using made up data pick the best option. Any choice would be correct with the right data.