Computer Organization / Professor H. Roumani

Lab-M

The Question

1. Create a working directory for this exercise and copy to it the file `cpu.v` that you created in this lab.
2. Recall that the `yEX` module in your `cpu.v` has the following header:

   ```
   module yEX(z, zero, rd1, rd2, imm, op, ALUSrc);
   ```

   where `zero` is a 1-bit zero flag.
3. In this exercise you are asked to modify this module (and any module on which it depends) so that `zero` becomes 2-bit wide instead of 1-bit. Set the least significant bit of this signal (i.e. `zero[0]`) so it acts as a *zero flag* (as before) and set its most significant bit (i.e. `zero[1]`) so it acts as an *even flag*. These two flags are defined below.
4. The zero flag should be `1` if the output `z` is zero and should be `0` otherwise.
5. The even flag should be `1` if the output `z` is even and should be `0` otherwise. Note that the number `0` is considered even.

Testing

It is strongly recommended that you test your modified `cpu.v` before submitting it by creating a `ram.dat` program image (or reuse the one in the lab) and a testing module `Test.v` (similar to `LabM10.v`) that issues the appropriate control signal for the program image; executes it; and then determines if everythingg is working correctly. To compile, use:

```
iverilog Prog.v cpu.v
```

and run with `ram.dat` in the same directory. This is in fact how your submitted module will be marked.

Submitting Your Work

1. Issue the following command from the working directory:

   ```
   submit 2021 LabM_Tue cpu.v
   ```

2. Note that every submission you make **overwrites** the content and the timestamp of any previous one; i.e. we only keep the very last file you submit.

3. This test is **40 minute** long.