EE 4702 **Homework 5** **Due: 19 April 2000**

Solution templates can be found in `/home/classes/ee4702/files/v` *and will be linked to the web page. Put your solution in a file named hw05sol.v. Soon after the time the assignment is due your directory tree will be searched for files named hw05sol.v and the most-recently modified one will be copied. If no such file is found an attempt will be made to copy a file using a guessed name, but this is not something to be relied on. Give the file the correct name.*

Solutions to the problems below should be synthesized for the following technology: ASIC (type of target), Sample (manufacturer [usually]), XCL05U (technology family). Do not specify any optimization or other synthesis options. View the RTL schematic to check your solutions. (Under the Tools menu or using the toolbar button.) Leonardo is started by typing `leonardo &` *in a shell. To work around a cosmetic stdout bug start Leonardo by typing* `leonardo > /dev/null &` *. Additional instructions on running Leonardo will be posted later.*

The assignments will be graded under the assumption that the schematic was viewed; a substantial number of points will be deducted for solutions that do not synthesize correctly.

Problem 1: Complete the Leonardo-synthesizable Verilog description of an ALU module shown below. The module has three inputs, `a`, `b`, and `op`. Inputs `a` and `b` are each 8 bits and hold unsigned integers. Input `op` specifies an operation to perform; the coding is given by the parameters. The module has two outputs, `res` and `err`. Output `res` is 8 bits and is the result of performing the operation; output `err` is one bit and is 1 if `res` cannot hold the result of `op`. That is, `err` is one if the sum is more than eight bits or the difference is negative; it is zero otherwise.

Write the description using behavioral code and synthesize it for the target specified above. The synthesized module should be combinational—no latches allowed. The module should perform only the three operations indicated, **don't** add your own.

```
module alu(res,err,a,b,op);
   input a, b, op;
   output res, err;

   parameter op_add = 0, // Addition.
             op_sub = 1, // Subtraction
             op_and = 2; // Bitwise and.

   // Insert solution here.  It's okay to delete this comment.

endmodule // alu
```

Problem 2: Complete the design of a Leonardo-synthesizable Verilog module with four 1-bit outputs and six one-bit inputs with the following behavior when synthesized:

- Output `w` is equal to the value input `d` had at the last negative edge of `clk`. (In other words, `w` is set to `d` at the negative edge of `clk`.)
- Output `y` is equal to the value input `a` had at the last positive edge of `clk`.
- Output `z` is equal to the last value input `c` had when both `clk` was high and `d` = `b`.
- Output `x` set to `b` at positive edge of `clk` if `a`=1. Output `x` is set to 1 if `clk`=1 and `d`=`a`.
- All outputs are set to zero when input `r`=1 (and will remain zero until set to a new value as described above).

If might be helpful to figure out what kinds of flip-flops are needed for each output and then check if Leonardo chooses the correct one (or something equivalent).

```
module latch_thing(w,x,y,z,a,b,c,d,r,clk);
   input a, b, c, d, r, clk;
   output w, x, y, z;

   // Insert solution here.  It's okay to delete this comment.

endmodule // latch_thing
```