Department of Electrical and Computer Engineering
University of Texas at Austin

ECE 382N, Spring 2000
Y. N. Patt, M. D. Brown
Problem Set 1b.
Due: February 11, 2000.

The purpose of this assignment is to introduce you to the Verilog hardware description language, which you will be using all term in carrying out your major design project. It is our intent to introduce these tools with an example that is small enough and familiar enough that you can focus on mastering the tools, rather than having to concentrate on the parts you are designing.

1. First extend the ALU which you designed in Problem Set 1a, as follows: Connect the output of the ALU to a 16-bit register (we will call it ALUout) made of edge-triggered D flip flops. Connect 16-bit, 2-1 multiplexors to each input of the ALU. We will call them muxa and muxb. Connect the Q output of ALUout to inputs of both muxa and muxb.

2. Translate this logic circuit into Verilog. The ALU should be constructed structurally. It can then be instantiated in a higher module, say TOP. You can use the library part dff$ to instantiate the edge-triggered D-flip flops. You are expected to design your own multiplexors, again using NAND gates only. Your ALU module may use the following port declaration:

```
module ALU (a, b, s, out);
input [15:0] a, b;
input [1:0] s;
output [15:0] out;
      :       :
```

3. Create a behavioral clock in the TOP module. Set the cycle time to 100ns. The D-flip flop is clocked with this signal. Apply the appropriate input patterns specified in the TOP module using the *initial* statement. You may assume that input changes (including the select inputs) are synchronized with the positive edge of the clock.

4. Using VCS, verify the correctness of both your ALU design and your timing calculations by applying test vectors and verifying the results. Use the "typical" delays given for the library parts to calculate the worst case delay. Use VirSim to generate the timing diagrams, and print out a hardcopy. The timing diagrams should include at least the following: the clock, the ALU control signals, the output of the ALUout register, the output of the muxes, and the output of the ALU.

Make sure that you apply the input patterns which correspond to the worst case delay for the complete feedback path (i.e., from the output of ALUout, through the mux, through the ALU, and back to the inputs of the ALUout register). All the signals along the feedback path should be included in your timing diagrams and your simulations should continue long enough for the results to arrive at the output of the register. Show that your input patterns indeed correspond to the worst case delay in the ALU. Submit more graphs if necessary.

5. Gradually reduce the cycle time. Find the minimum achievable cycle time that will NOT cause any timing problem in the circuit. Recall that the sum of the delays along the feedback path, plus the setup time of the D-flip flop, should not be greater than the cycle time. Is your calculation consistent with that observed in the simulations?

6. Prepare a short report on the work done in step 3 through step 5 that includes the following:
1) copies of the (hand-drawn) schematics of your ALU, muxes and registers

2) the calculations for the critical path length of your feedback path
3) timing diagrams with examples of each ALU function
4) timing diagrams demonstrating the smallest cycle time using critical path inputs
5) timing diagrams demonstrating the longest clock cycle that the circuit did not work properly for.