LSU EE 4755 Homework 6 Due: 1 Dec 2024

Student Expectations

To solve this assignment students are expected to avail themselves of references provided in class and on the Web site, such as for Verilog programming and synthesis examples, and to seek out any additional help and resources that might be needed. (Of course this doesn't mean asking someone else to solve it for you.) It is each student's duty to himself or herself to resolve frustrations and roadblocks quickly. (If you get stuck *just ask for help!*)

This assignment cannot be solved by blindly pasting together parts of past assignments. Solving the assignment is a multi-step learning process that takes effort, but one that also provides the satisfaction of progress and of developing skills and understanding.

Collaboration Rules

Each student is expected to complete his or her own assignment. It is okay to work with other students and to ask questions in order to get ideas on how to solve the problems or how to overcome some obstacle (be it a question of Verilog syntax, interpreting error messages, how a part of the problem might be solved, etc.) It is also acceptable to seek out digital design resources for help on Verilog, digital design, etc. It is okay to make use of AI LLM tools such as ChatGPT and Copilot to generate sample Verilog code. (Do not assume LLM output is correct. Treat LLM output the same way one might treat legal advice given by a lawyer character in a movie: it may sound impressive, but it can range from sage advice to utter nonsense.)

After availing oneself to these resources **each student is expected to be able to complete the assignment alone**. Test questions will be based on homework questions and **the assumed time needed to complete the question will be for a student who had solved the homework assignment on which it was based**.

Problems start on next page.

Problem 1: Show the hardware that will be synthesized for the posted solution to Homework 5. The solution (with fewer comments than the posted version) is shown below.

```
module dot_seq_2 #( int w = 5, wi = 4 )
   ( output logic [w-1:0] dp,          output logic [wi-1:0] first_id, last_id,
     input uwire [w-1:0] a[2], b[2],   input uwire [wi-1:0] in_id,
     input uwire reset, first, last,   input uwire clk );

   logic [w-1:0] pl_a[1:1][2], pl_b[1:1][2];  // Arriving vector elements.
   logic [w-1:0] pl_prod[2:2][2];             // Vector products.
   logic [w-1:0] pl_sum[3:3];                 // Dot prod of 2-element segment.
   logic [wi-1:0] pl_id[1:3];                 // ID.
   logic [1:0] pl_fl[1:3];                    // The first and last signals.
   logic [wi-1:0] acc_id;
   logic [w-1:0] acc_sum;

   always_ff @( posedge clk ) begin
      /// Stage 0
      pl_a[1] <= a;  // This copies both elements of a.
      pl_b[1] <= b;
      pl_id[1] <= in_id;
      pl_fl[1] <= reset ? 2'b0 : {last,first};

      /// Stage 1
      for ( int i=0; i<2; i++ ) pl_prod[2][i] <= pl_a[1][i] * pl_b[1][i];
      pl_id[2] <= pl_id[1];
      pl_fl[2] <= reset ? 2'd0 : pl_fl[1];

      /// Stage 2
      pl_sum[3] <= pl_prod[2][0] + pl_prod[2][1];
      pl_id[3] <= pl_id[2];
      pl_fl[3] <= reset ? 2'h0 : pl_fl[2];

      /// Stage 3
      begin
         automatic logic s3_first = pl_fl[3][0]; // For readability.
         automatic logic s3_last =  pl_fl[3][1]; // For readability.
         automatic logic [w-1:0] s3_sum = s3_first ? pl_sum[3] : pl_sum[3] + acc_sum;

         acc_sum <= s3_sum;

         if ( reset ) begin
            first_id <= 0;
            last_id <= 0;
         end else begin
            if ( s3_first ) acc_id <= pl_id[3];
            if ( s3_last ) begin
               first_id <= s3_first ? pl_id[3] : acc_id;
               last_id <= pl_id[3];
               dp <= s3_sum;
            end
         end
      end
   end
endmodule
```

Problem 2: Solve 2023 Final Exam Problem 2, which asks for a cost and delay analysis of a word count module.