EE 3755 Homework 2 Due: 15 October 2001

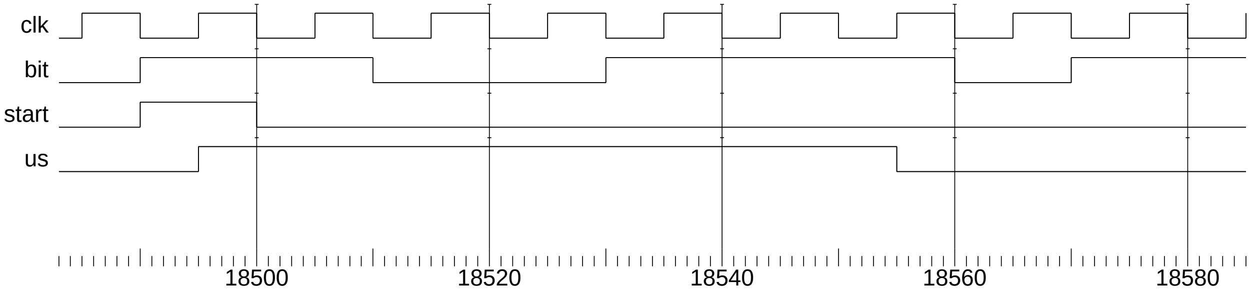
Solve this problem by modifying a copy of `http://www.ece.lsu.edu/ee3755/2001f/hw02.html` *which can also be found in* `/home/classes/ee3755/com/v/hw02.v`*. See* `http://www.ece.lsu.edu/ee3755/proc.html` *for instructions on running the simulator. Alternate instructions can be found in Lesson 7 of the ModelSim Tutorial, linked to the references web page,* `http://www.ece.lsu.edu/ee3755/ref.html`*. The links are clickable when this assignment is viewed with Acrobat Reader. The ModelSim tutorial and other documentation can also be accessed from the Help menu on the ModelSim GUI (started by the command vsim -gui).*

Copy the homework template, `/home/classes/ee3755/com/v/hw02.v`, into a subdirectory named `hw` in your class account.

The following definition is used in the problems below: An n-bit vector is *uniformly striped* if bits 0 to $b-1$ are 1, bits b to $2b-1$ are 0, bits $2b$ to $3b-1$ are 1, etc, where bit 0 is the least-significant bit and b is a positive integer. The vector is uniformly striped even if the last run of consecutive 0's or 1's is too short. For example, the following vectors are uniformly striped: 1010101, 0110011, 1000111. The following vectors are NOT uniformly striped: 0101101, 0011001, 1100111. (Note that the rightmost bit is bit 0.)

Problem 1: Module `uni_striped_c`, which can be found in the homework template, has a 32-bit input `v` and a one-bit output `us`. Complete the module so that output `us` is 1 if `v` is uniformly striped and is 0 otherwise. The module should use procedural code and should synthesize to combinational logic (no registers). Use testbench `test_us_c` to test `uni_striped_c`.

Problem 2: Module `uni_striped_s`, also in the template, is a sequential version of `uni_striped_c`. It has three one-bit inputs, `bit`, `clk`, and `start`. At the positive edge of each `clk` a bit of a vector appears on `bit`. When `start` is 1 the first bit appears, at the next positive edge of `clk` the second bit appears, and so on until `start` is 1 again. (Start is 1 only for the first bit of the vector.) After each positive edge, `us` should be asserted if the bits received starting from when `start` was last 1 form a uniformly striped vector. When a bit is received that breaks the uniformly striped pattern `us` should be set to zero and should stay zero until `start` is asserted at which time a new vector starts. Sample timing appears in the diagram below:



Complete module `uni_striped_c` using synthesizable code (in Form 2). The module should handle vectors of at least 32 bits and should be written so that much longer vectors can be handled. The module **must not** store the vector. Use testbench `test_us_s` to test `uni_striped_s`.

Problem 3: Module `andyetanotherif`, in the template, is from Lecture Set 6. Based on a hand analysis (that is, don't run the synthesis program) using the techniques presented in class show the hardware that would be synthesized for this. The solution to the problem can be submitted on paper or electronically by placing a file named hw1.ps, hw1.pdf, hw1.gif, or hw1.png in the same directory as the completed Verilog homework file.