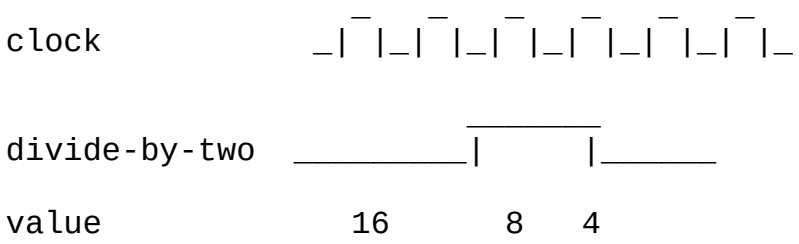
ECE 520 / ECE 464
Homework 3

Question 1

The purpose of this question is to start you using the simulation and synthesis tools. Do tutorial 1 listed with this homework on the web page.

Then do the following:

(a) Redesign this module as an 8 bit down-counter. In addition, add the input `divide-by-two`. Whenever `divide-by-two` is high and `latch` is low and `dec` is low, divide the current contents of the counter by 2 (by doing a right shift). e.g.



(b) Synthesize the fastest possible design, by taking the strategy of reducing the clock period and doing incremental compiles. e.g.

```
      /* see if the design works with a faster clock */
      create_clock -period 9 -waveform {0 4.5} clock
      compile -incremental
      report_timing
      /* if this design works, save it in case the next one did
not */
      write -f ddc -output tmp9.ddc

       /* see if the design works with a still faster clock */
      Create_clock -period 8 -waveform {0 4} clock
      compile -incremental
      report_timing
      /* if this design works, save it in case the next one did
not */

     write -f ddc -output tmp8.ddc
```

```
/* etc. */
```

You will need to iterate on this loop after your first compile but before the `Fix hold times` comment. It is best if you break the current script into 2 files; one up to the first report_timing and one after that. Enter the above commands by hand into design_analyzer or dc_shell.
When you find a clock speed that does not work, then read back in the .db file for the fastest working clock speed `(read_file -format ddc tmp?.db or read_ddc tmp?.db)` and perform the rest
of the original script.

In your solutions, please turn in the following:

- A copy of your Verilog listing (first page), as well as your test fixture (second page).
- A cut-and-paste copy of your final timing reports for both set-up and hold. Take these from the synopsys view_command.log file.

[90 points]

Use the template on the next page, as the first page of your solutions. Make sure this template is one page. Attach other documentation as needed.

Note, This is just an exercise to get you more familiar with Synopsys and with how to squeeze the timing down in an actual design. You are not expected to do it in later exercises. It is up to you if you explore this in the project. However, in general the fastest possible design does not maximize performance/area. To explore that goal, you need to look at both timing and area (report_area).

Question 2
10 points is assigned to this question. Its purpose is so you can compare and contrast the FPGA tools to the Synopsys tools. Look at Tutorial 6 on the ASIC Tutorial page about the Alterra toolset. Download the Alterra student edition and run your design example through the Alterra tool flow. It does not matter which FPGA family you use. Please note the following in any answer:

1. The Alterra synthesis flow looks after much of the scripting for you. However, did you note how the underlying scripting language is identical to the TCL language you used in Synopsys?
2. What was the speed difference between the FPGA implementation and the 45 nm library implementation? Note, that the FPGA you are using is implemented in a more advanced technology.

Note. Please use the standard cell Synopsys flow for all future exercises. However, you might find the Alterra flow useful for debugging your designs without needing campus tools.

Grading Rubric
Total grade is out of 100.

Question 1

- 30 points: Does redesigned counter look like it should work correct. Give 3if it does, give 20-29 if there are minor errors, less if there are major errors.
- 20 points. Did the test fixture look appropriate for the new counter. Does it explicitly exercise the devide_by_2 feature? 20 points if it is appropriate. Less if not.
- 10 points. Did the student include a simulation waveform showing the divide_by_2 feature working? 10=yes; 0=No.
- 20 points. Did the student seem to do the clock exercise and get a substantially faster clock? 20=yes; 10=partially; 0 = No.
- 10 points. Do the timing reports indicate a working design. 10=yes; 0=No.

Question 2

Add 10 points if the student did the Alterra exercise and commented on the two points asked about.

Name: Student ID:

I certify that I individually did the design, simulation and synthesis that lead to these solutions, did not copy the answers to this homework and did not let anyone copy my answers (sign):

Clock Period Achieved (ps):

Verilog design File (not test fixture – put that and the simulation output on the following pages):

Min and Max Timing Reports: