This lab will be the first time you use clocked sequential circuits. For this lab you'll build a simple sequential system that counts down on the bar graph, and then goes into a loop where some sort of pattern is displayed on the other LEDs on the board. Other than the count down feature, this lab is essentially to “do something interesting with the LEDs.” I'm keeping that vague on purpose so you can be creative about what that means.

The required inputs to your circuit are:

- **clr**: a signal that starts your LED sequence from the beginning. This should be connected to the pushbutton on the XSA board which is active low. That is, the button provides a high signal (on pin P93 of the FPGA) until it’s pressed. When the button is pressed, that signal goes low.

- **clk**: A free running clock signal is generated by the oscillator on the XSA board and is connected to pin P88 on the FPGA. Unless someone has used the GXSSETCLK program to reprogram the oscillator, this will be running at 100MHz. That is, connecting a Xilinx input to pin P88 will give you a 100MHz clock to use in your circuit.

- You can use other buttons and switches as inputs to control your patterns if you like, but that’s not required. Remember that some of the switches interfere with some of the LEDs so if you use other switches or buttons, read the switch/LED handout carefully.

The required outputs from your circuit are:

- When the circuit is reset, all the bar graph LEDs on the XST board should be lit. Then they should start going out one at a time as a countdown to your LED patterns. When the bar graph counts down to 0 (i.e. all LEDs off), your LED pattern should start and should repeat until the reset button is pressed again. The bar graph count down should be around a half-second per tick. That is, all the LEDs start lit, and then one by one, at about a half second or so interval, they should start to go off.

- Once the bar graph counts down to 0 (all bar LEDs off), your LED pattern should start, and should repeat until the reset button is pressed. Your pattern should “do something interesting” and the definition of “interesting” is up to you. Your pattern should be at least 8 states long, but can be as long as you like. You can use any of the LED’s that are on the two boards (XSA and XST boards). You might choose to have your pattern be influenced by other switches and buttons, but that’s not required. You might make your pattern cycle at a faster rate than the countdown so that you can countdown slow and then speed things up for the “light show.” It’s up to you!
Some things to consider:

- Because the bar graph shares lines with the 7-segment display on the XSA board, that LED will light up during your countdown. That’s OK.
- If you use LED1 on the XST board (the one next to the bar graph), note that some of the segments share pins with the DIP switch on the XSA board, so you have to make SURE those switches are off (over by the numbers) to get all the segments to light up. If any of the DIP switches are ON, the segments that share those wires won’t light up. Even worse, if the switch is ON, then that wire will be pulled to ground. If the switch is trying to pull the wire to ground, and the Xilinx part is trying to drive it high, high current can result, and that can physically damage the Xilinx chip!
- If you use LED2 on the XST board, note that many segments share wires with the parallel port. Because of this, you MUST first load the dwndpa2.svf file into the CPLD (using GXSLOAD) before downloading your .bit file. This file is in the XSTOOLS installation directory in the XSA/50/LPT folder. You only need to do this once because the CPLD is non-volatile.
- Note that the system clock on pin P88 is running MUCH too fast (100MHz) to be used for a human-visible LED sequence. You must divide that down into something more reasonable to a human eye. My suggestion is to use Verilog to make a counter and pick off an output bit of the counter to use as your “slow clock” (human-time-scale clock) as described in Section 7.14.3 in your book.
- There are many ways to do this lab, and any of them are OK as long as they achieve the goal of sequencing through a set of interesting patterns on the LEDs. One way that is probably the easiest is to define a count variable that counts up on the slow-clock, and then depending on what the value of the count is, change the LED pattern.
- You might want to make your countdown slower than your pattern. You can do this a number of ways, but probably the easiest is to just make the initial countdown change after every 2nd, 3rd, or 4th (or whatever) count, and then have the pattern change on every count. You can use this trick to speed up and slow down in your pattern too.
- This shouldn’t take you very long to do! If you’re spending a huge amount of time just getting things going, you should talk to me or to a TA and make sure you understand things. The only thing that might take time is polishing your dazzling light show once you have the basic interface working. My quick (not very interesting, but working) demo took me about an hour from start to finish.
- You don’t need to hand in a simulation. You just need design documents (schematics and Verilog), and a working demo.