Exercise 8 Cadence Filter **(HW2 prob 5)**:

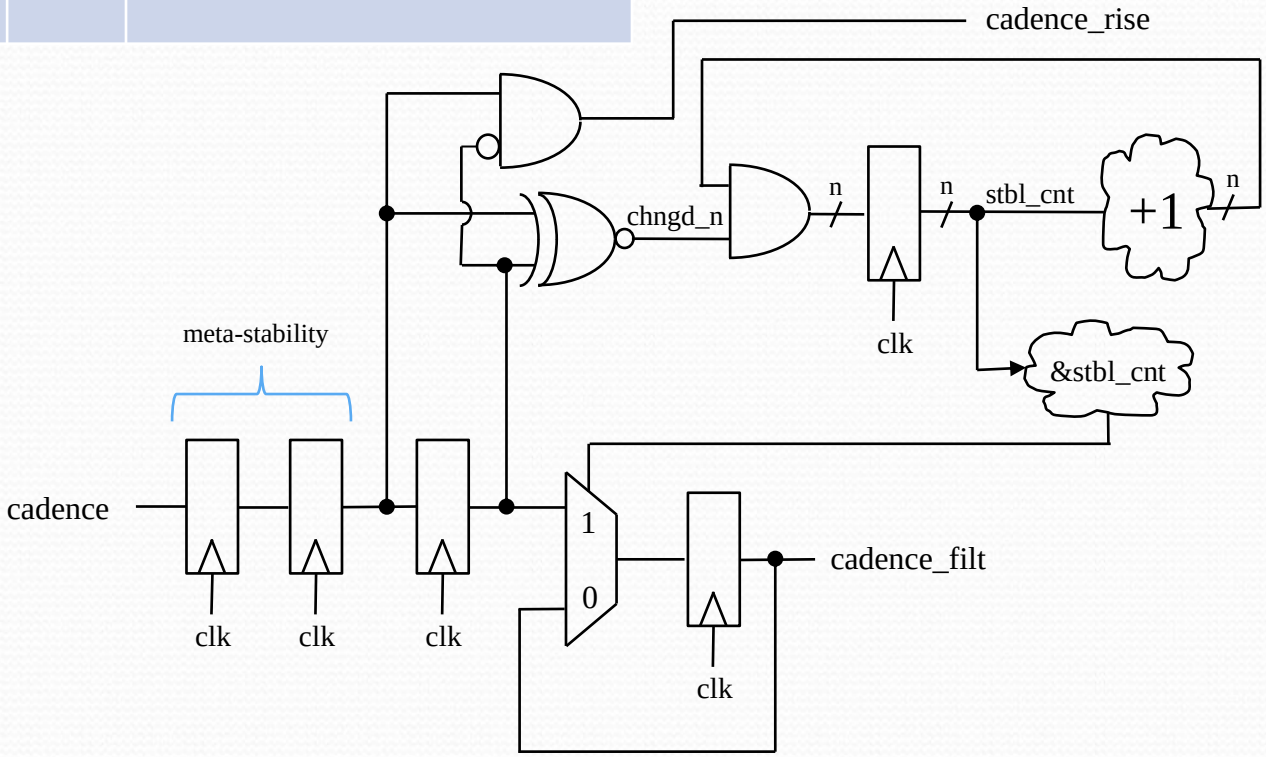
This problem will be a bit of a stretch of your current knowledge since it will be your first time inferring flops & counters. In this problem you are implementing a block needed for the project, so you want to do a good job. I suspect, however, later in the semester *(when you are more learned in the ways of Verilog)* you will look back at the code you wrote here and say: "yowzah that is garbage". Just give it your best "college try" for now.

The sensor that senses how fast the rider is pedaling is called a cadence sensor. It gives out 32 digital pulses per rotation of the cranks. The problem is the sensor is noisy as all heck. I think it is built with a rotating mechanical slip ring. All I know is it can give many false up and down pulses at the transition points, and that the output seems very low impedance and cannot be filtered via an analog low pass filter. We are going to have to build a digital filter to filter it.

Exercise 8 Cadence Filter (HW2 prob 5):

Signal:	Dir:	Description:
clk	in	50MHz clock
rst_n	In	Asynch active low reset signal
cadence	in	Raw cadence signal (bouncy and asynch)
cadence_filt	out	Filtered meta-stability free version of cadence signal
cadence_rise	out	Rise edge detect of cadence

Our filter is just going to look for greater than 1ms of stability in the signal before allowing the filtered version to be updated with the current signal. See the following diagram:



Exercise 8 Cadence Filter *(verbal description)* **(HW2 prob 5)**:

Of course, the incoming signal from the sensor is asynchronous to our clock domain so has to be double flopped to eliminate meta-stability. I am sure you would have thought of that, so sorry to mention it. An XNOR gate can look across the synchronized signal vs what the signal was 1 clock ago to see if it changed. If it did change our stability counter should be knocked down to zero. I will let you figure out how wide (**n**) the counter needs to be to measure at least 1ms of stability (our clock frequency will be 50MHz). If the stability counter gets up to a full count it means the signal has been stable for at least 1ms so the filtered version of the signal is allowed to take on the current version of the signal. Yes..this filter does introduce more than 1ms of delay to the signal and limits the frequency of what can come through. However, think of the application…does it matter? Even Greg LeMond never pedaled that fast. Code **cadence_filt.sv** with the following interface and submit it to the dropbox.

A self checking testbench (**cadence_filt_tb.sv**) has been provided. Use it to check your design. **Submit** a capture of the transcript window showing it ran successfully.