Massachusetts Institute of Technology
Department of Electrical Engineering and Computer Science

6.976
High Speed Communication Circuits and Systems
Spring 2003

Homework #6: Phase-Locked Loop Circuits

Reading: PLL Design Assistant Manual (found at
http://www-mtl.mit.edu/research/perrottgroup/tools.html).
Chapter 15 of Thomas H. Lee's book.

---

1. This problem focuses on a simplified method of representing a phase-locked loop circuit with a given noise profile for system level simulations. We will focus on deriving the appropriate noise parameters to achieve a given level on noise performance, and then investigate simulation using CppSim. In all cases, assume that K_v = 20 MHz/V, K_1 = -95 (dBc/Hz), K_2 = -60 (dBc), f_p = 100 kHz, f_s = 90 kHz, and that the carrier frequency f_o = 200 MHz.

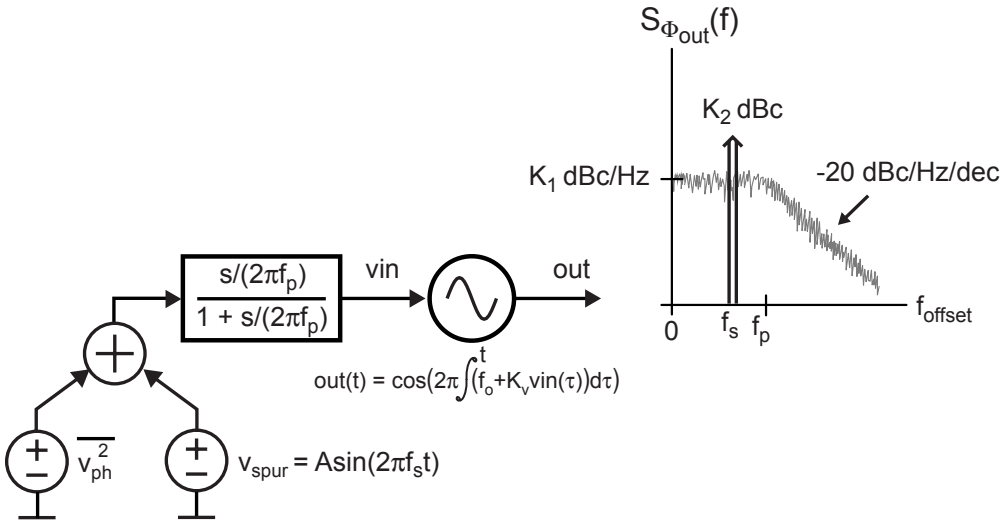

Figure 1: Simplified frequency source simulation implementation with given phase noise and spurious noise.

   (a) Draw the block of the system shown in Figure 1 in which all blocks are cast as transfer functions (i.e., the VCO is an integrator block).

   (b) Based on your drawing in part (a), compute the variance of the phase noise source, $\overline{v_{ph}^2}$, to achieve the specified phase noise profile at low frequencies, K_1.

(c) Compute the amplitude of the spurious noise source, $v_{spur}(t)$, to achieve the specified spurious noise at frequency f_s, K_2.

(d) Simulate the system in CppSim to verify your calculations using a Gaussian noise source (noise module), a sine wave source (signal_source, gain modules), a summer (add2 module), an appropriate filter (must create), and a VCO (vco module). Use a simulation sampling frequency of 2 GHz, and simulate over 1,000,000 sample points. To plot the phase noise, use the Matlab script comp_psd.m located in /mit/6.976/CppSim/MatlabCode (be sure to change the value of K_v in the script to match the value assumed here).

(e) You should notice that the plot given by the comp_psd.m script matches the phase noise (assuming you have done things correctly), but not the spurious noise. Measure the spurious noise by taking the fft of the VCO sine wave output and compare the spur directly against the carrier tone. Submit resulting phase noise plot and also plot of fft showing the carrier and spur tones.

(f) Why is the spurious noise improperly scaled with the comp_psd.m script? What does this exercise teach with respect to proper measurement of phase noise and spurious noise in simulations?

2. The following problem focuses on the design of an integer-N frequency synthesizer as depicted in Figure 2. Assume that a tristate phase detector is used, $K_v = 20$ MHz/V, the desired nominal output frequency is 200 MHz, the desired frequency resolution is 2 MHz, and that the desired phase noise profile should closely match that in Problem 1. Also, assume that the loop filter will be designed such that the PLL will be Type I, second order, and have a bandwidth of 100 kHz with Butterworth shape.

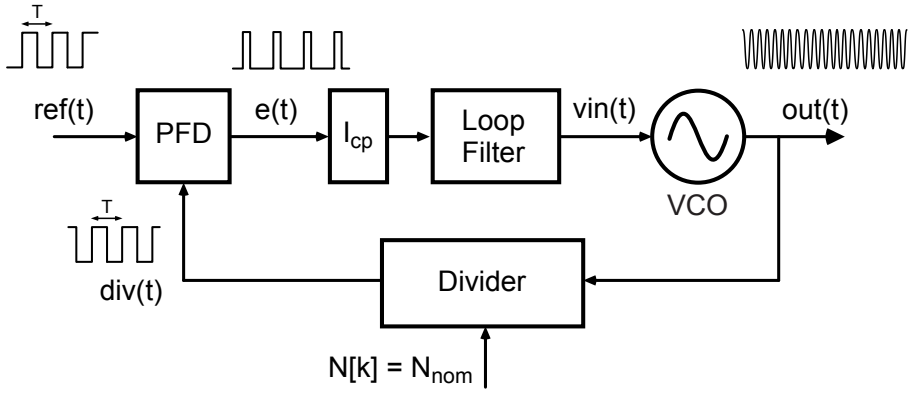

Figure 2: An integer-N frequency synthesizer.

(a) Given the assumptions stated above, draw a transfer function level block diagram of the synthesizer shown in Figure 2.

(b) Download the PLL Design Assistant program available at http://www-mtl.mit.edu/research/perrottgroup/tools.html (note that it only runs in Windows). Use the program to calculate the required loop filter parameters

to achieve the required PLL dynamics (also specify the nominal divide value and reference frequency). At this point, perform your calculations based on the assumption that $I_{cp} = 100\ \mu$A.

(c) Simulate your design using CppSim and verify that the (small signal) step response of the system matches that estimated by the PLL Design Assistant program. Note that you can simply modify the sd_synth schematic in the CppExamples library to implement the system — you will need to implement the tristate detector and a new version of the loop filter. Keep the noise sources in (just set their value to zero), and replace the constant input to the reference vco module with a step input to examine the (small signal) step response. Submit Cadence schematics of PLL and tristate pfd and also the code for modules that was added to modules.par file.

(d) Verify that the PLL relocks even if frequency lock is lost by stepping the divide value by a value of 1 and a value of 10. If it fails to lock in either of these cases, explain why that is the case. How could you modify the circuit so that it would unconditionally lock? Submit Matlab plots for the synthesizer responses for both divider steps.

(e) Compute the value of charge pump noise parameters in the gated_noise module to achieve a low frequency phase noise of -95 dBc/Hz, as shown in Figure 1. For simplicity, assume that varpos equals varneg. Verify your calculation by simulating the system with the appropriately set charge pump noise in CppSim, and use the comp_psd.m script to plot the resulting phase noise.

(f) Compute the value of the input-referred VCO noise variance (i.e. the var parameter in the noise module) to achieve PLL output phase noise that closely matches that in Figure 1 (i.e., -115 dBc/Hz at 1 MHz offset). Verify your calculation by simulating the system in CppSim with both charge pump noise (computed from the previous part) and input-referred VCO noise included in the simulation, with comp_psd.m being used to plot the overall output phase noise.

(g) Calculate the value of the charge pump current that would be required to achieve the charge pump noise specification computed in part (e). Assume that for both the top and bottom charge pump currents, their current noise, $\overline{i_{ch}^2}$, is related to their respective current, I_{cp}, as

$$\overline{i_{ch}^2} = 8\text{e-}19 \cdot I_{cp}\ \text{A}^2$$

(h) Given the above charge pump computation:

i. Redraw the transfer function block diagram of the system (as done in part (a)) and indicate which values are modified to accomodate the new charge pump current value while still keeping the same PLL dynamics and phase noise profile as assumed throughout this problem.

ii. Draw the loop filter circuit (including component values) — assume that it is implemented without using an opamp.

3. Redo problem (2) with the PLL being chosen as a Type II rather than Type I implementation. In other words, assume that a tristate phase detector is used, $K_v = 20$ MHz/V, the desired nominal output frequency is 200 MHz, and the desired frequency resolution is 2 MHz. The loop filter will be designed such that the PLL will be Type II, second order, and have a bandwidth of 100 kHz with Butterworth shape. In addition, assume $\frac{f_z}{f_o}=\frac{1}{10}$.

   (a) Use the PLL Design Assistant program to calculate the required loop filter parameters to achieve the required PLL dynamics (also specify the nominal divide value and reference frequency). At this point, perform your calculations based on the assumption that $I_{cp} = 100$ μA.

   (b) Simulate your design using CppSim and verify that the (small signal) step response of the system matches that estimated by the PLL Design Assistant program. As with problem (2), keep the noise sources in (just set their value to zero), and replace the constant input to the reference vco module with a step input to examine the (small signal) step response. Submit Cadence schematic of PLL.

   (c) Does the step response take a longer or shorter time to settle than its Type I counterpart in problem (2)? Explain. Submit simulation step response plot with explanation.

   (d) Is there a difference in the pfd output between type I and type II? If so, what cause this difference? Submit plots of the pfd output for both types to prove your point.

   (e) Verify that the PLL relocks even if frequency lock is lost by stepping the divide value by a value of 1 and a value of 10. Submit synthesizer response plots for both step values.

   (f) Draw the loop filter circuit (including component values) — assume that it is implemented without using an opamp.