وَمَا أُوتِيتُمْ مِنَ الْعِلْمِ إِلَّا قَلِيلًا

Ain Shams University – Faculty of Engineering – ECE Dept. – Integrated Circuits Lab.
Dr. Hesham Omran

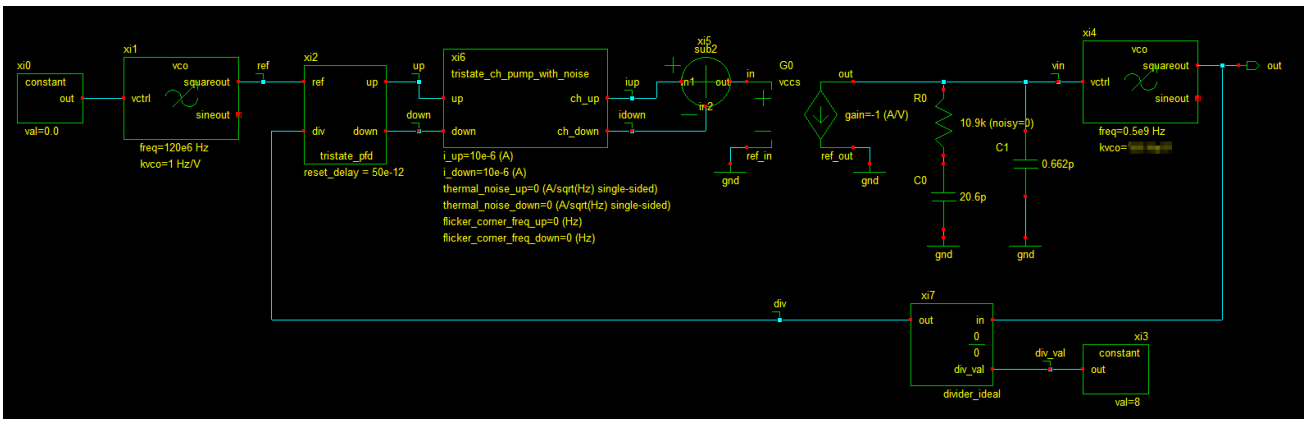
Analog/Mixed-Signal Simulation and Modeling
Lab 06
PLL System-Level Design and Simulation

Objectives

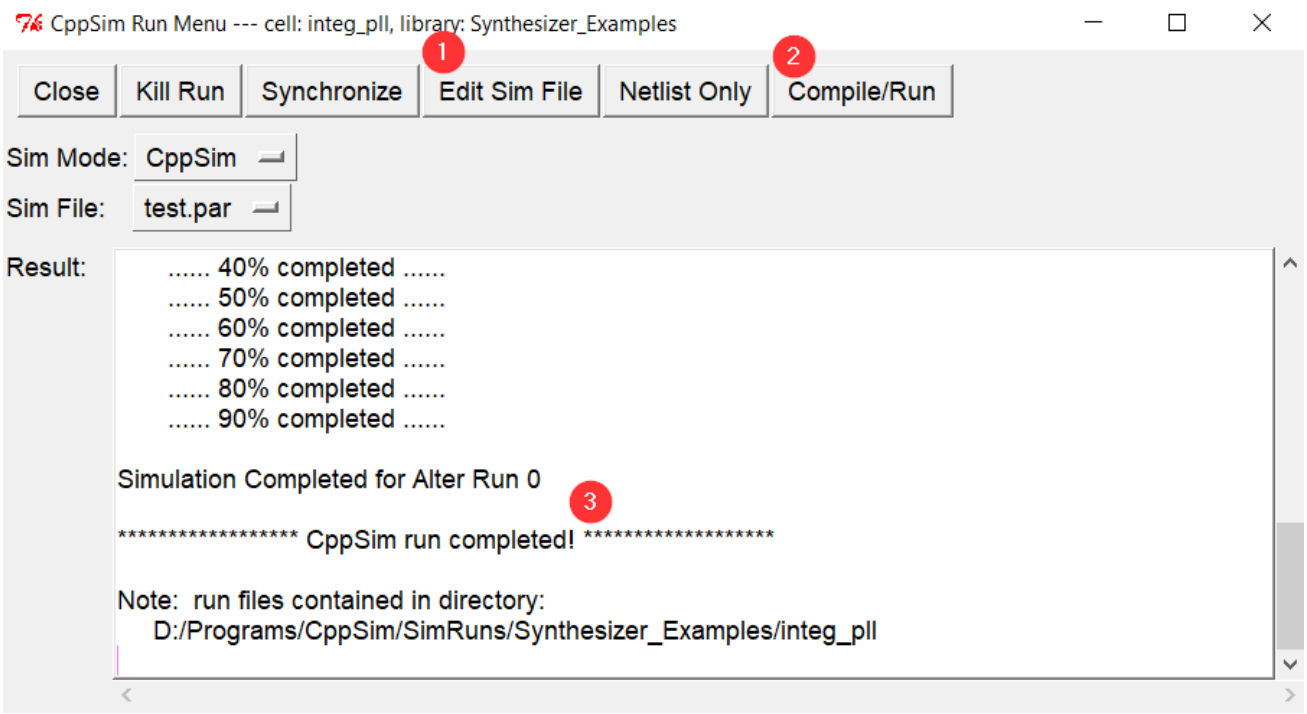
1. Learn how to use fast system level simulation tools.
2. Learn how to design and simulate a PLL system.

Instructions

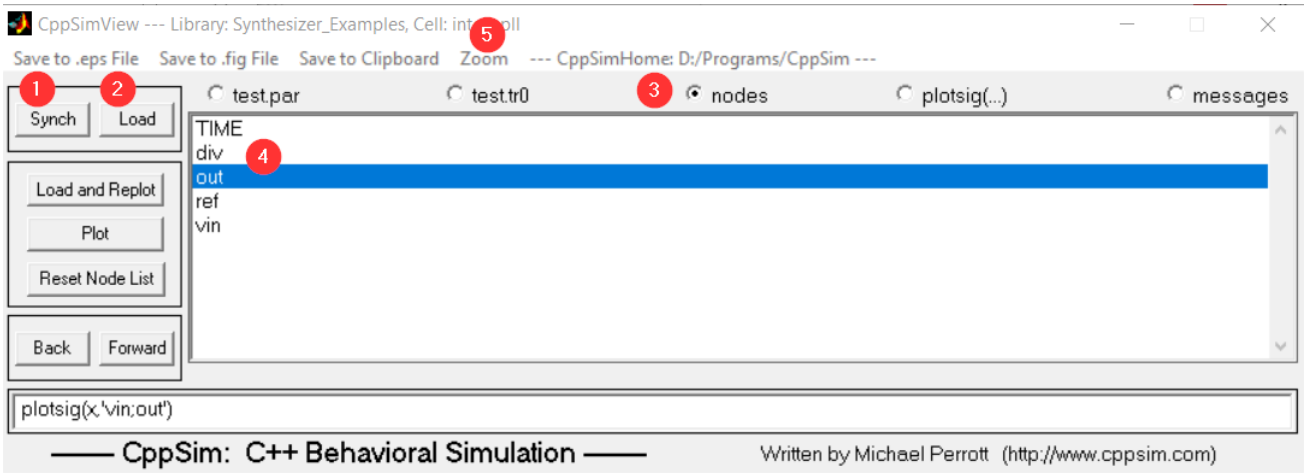
1. Download and install CppSim: https://www.cppsim.com/download.html
   Note: You must install CppSim in a path that does NOT have spaces.
2. From the Windows start menu: Run Sue2 (CppSim schematic editor).
3. Draw the schematic of an integer-N PLL similar to the one in "pll_design.pdf" and adjust all settings as shown in the figure below.
   Hint: Instead of creating a design from scratch, you may edit one of the existing Synthesizer Examples and use (File -> Save As) to save it to a new file.



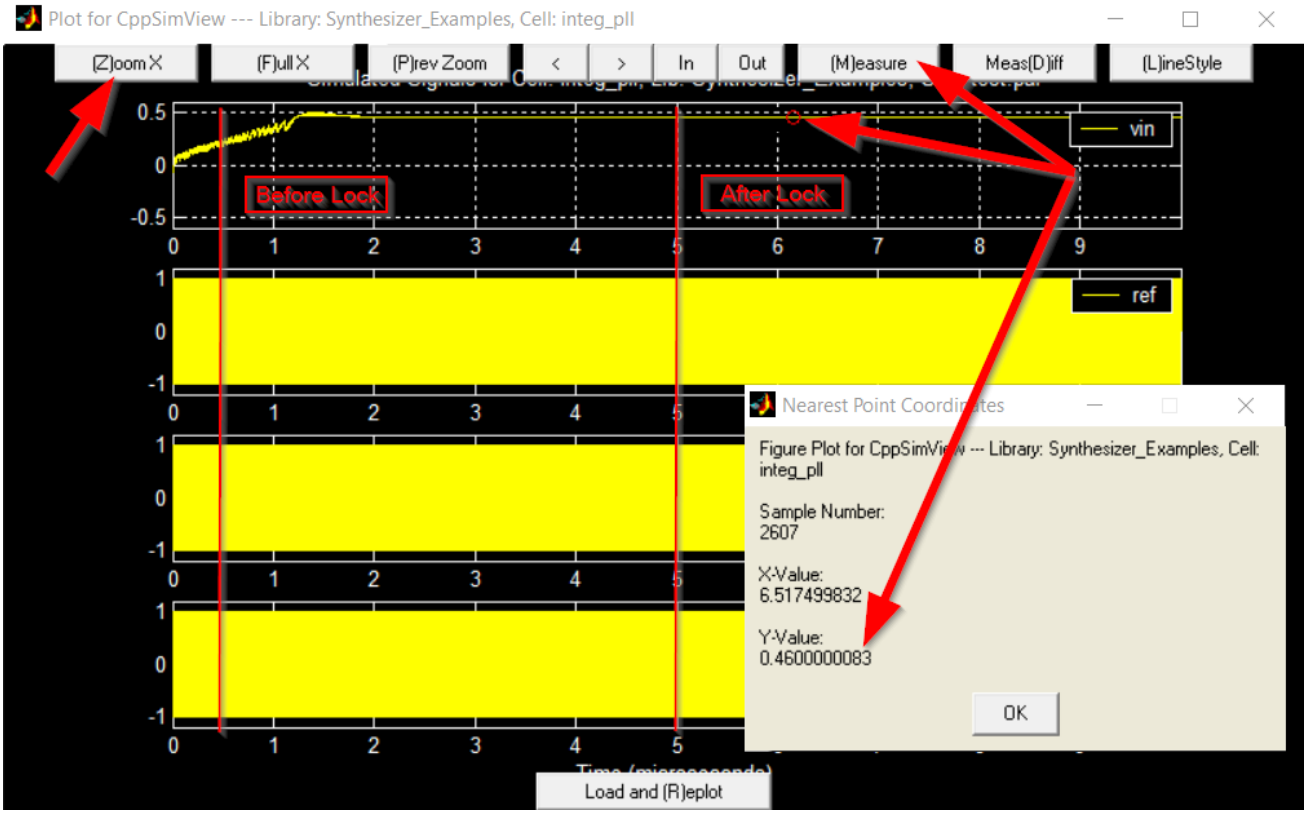
4. For the parameter kvco you will use a unique random number. To calculate your kvco, visit this random number generator website to get a random number between 50 and 400: https://www.calculator.net/random-number-generator.html?slower=50&supper=400&ctype=1&s=1922&submit1=Generate
5. Use kvco = your_random_num / 100 * 1e9 Hz/V
6. Open CppSim Run Menu (Tools -> CppSim Simulation).
7. Choose "Edit Sim File" and set the following settings:
   num_sim_steps: 1e6
   Ts: 10e-12
   probe: ref div vin out
   global_nodes: gnd=0.0 avdd=1.2 dvdd=1.2

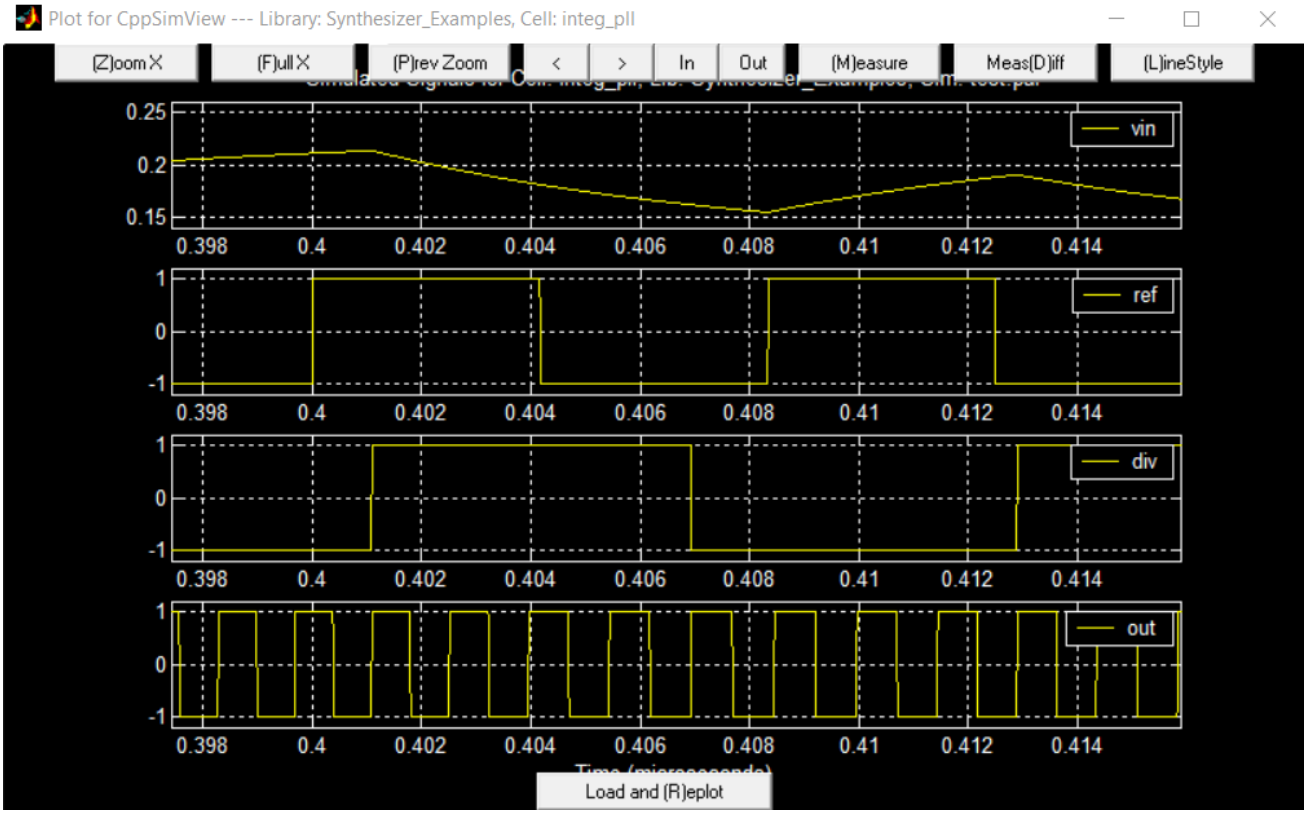


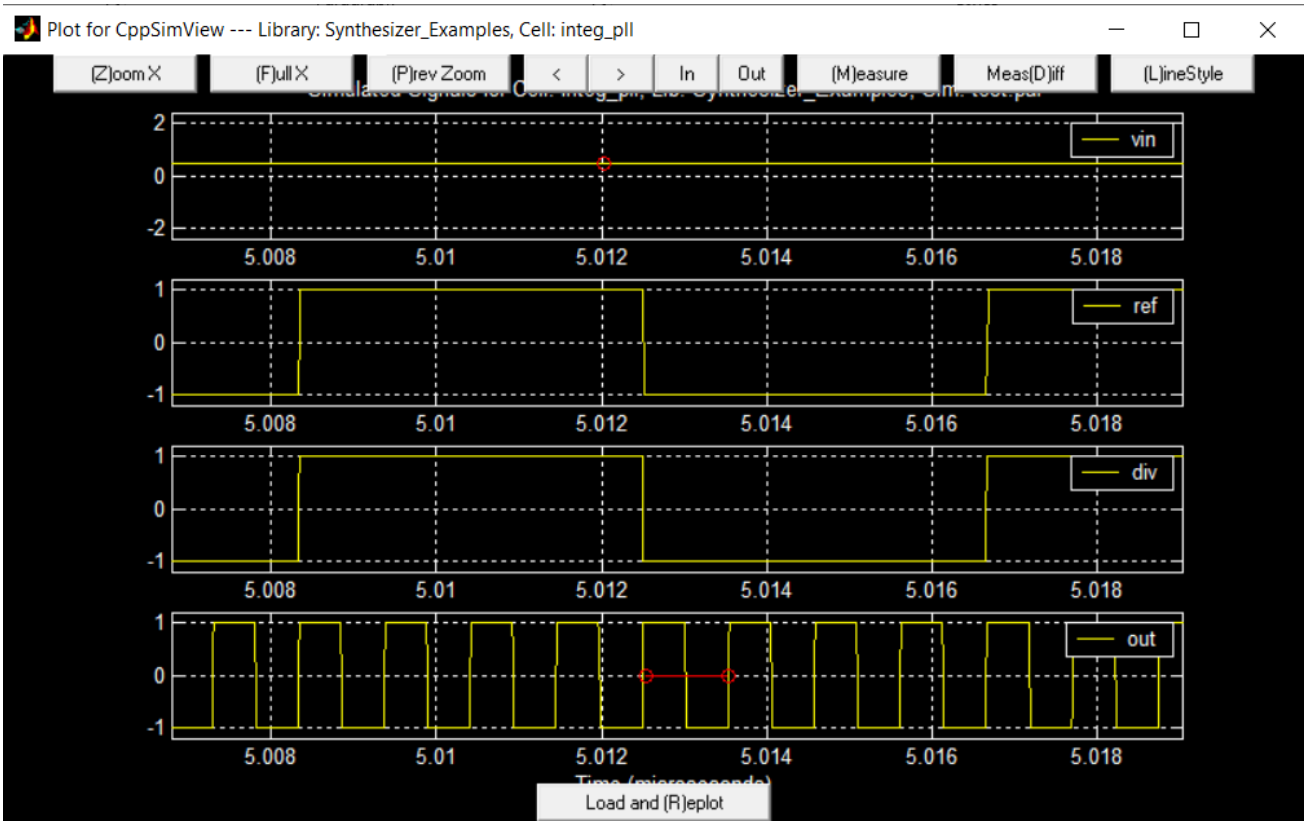
8. Choose “Compile/Run”. Make sure the simulation is completed without errors as shown above. Debug the errors if any.
9. From the Windows start menu: Run CppSimView (the waveform viewer).
10. Click “Synch”, then “Load”, then “nodes”, then double click each of the four nodes in the following order (vin, ref, div, vout). Then click “Zoom”.



11. Add a cursor using “Measure” to show the value of vin at lock clearly as shown below. Use “Zoom X” to zoom in the area before lock and after lock. Report snapshots to clearly show that ref and div are out of phase then in phase as shown below.


Plot for CppSimView --- Library: Synthesizer_Examples, Cell: integ_pll
(Z)oom X
(F)ull X
(P)rev Zoom
<
>
In
Out
(M)easure
Meas(D)iff
(L)ineStyle
vin
Before Lock
After Lock
ref
Nearest Point Coordinates
Figure Plot for CppSimView --- Library: Synthesizer_Examples, Cell: integ_pll
Sample Number:
2607
X-Value:
6.517499832
Y-Value:
0.4600000083
OK
Load and (R)eplot


Plot for CppSimView --- Library: Synthesizer_Examples, Cell: integ_pll
(Z)oom X
(F)ull X
(P)rev Zoom
<
>
In
Out
(M)easure
Meas(D)iff
(L)ineStyle
vin
ref
div
out
Load and (R)eplot



Part 1

Index	Deliverable
1.	Schematic of the PLL drawn in Sue2.
2.	Snapshot of the random number generation website showing your random number. Calculate kvco.
3.	Given kvco, analytically calculate vin at which the PLL will achieve lock.
4.	Snapshot of CppSim Run dialog showing successful simulation completion.
5.	Snapshot of the simulation results with a probe showing vin value as shown above.
6.	Compare the simulated vin value with the analytically calculated value. Comment.
7.	Clear Zoom X snapshot before lock as shown above.
8.	Clear Zoom X snapshot after lock as shown above.

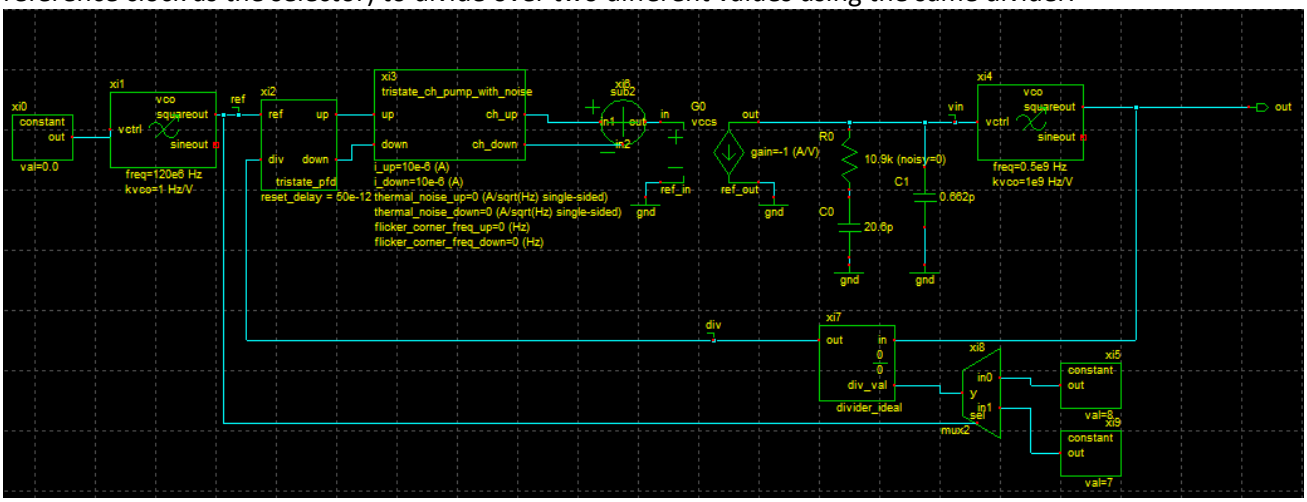
Part 2

Add a name to the charge pump output (output of xi5 sub2 in the above schematic). Use “Edit Sim File” to probe (up,down,chp_out) signals.

Index	Deliverable
1.	Plot (ref,div,up,down). Report clear Zoom X snapshot showing the PFD operation.
2.	Reset node list. Plot (vin,chp_out).
3.	[Optional] In Sue2 click (Doc -> PLL Design Assistant Manual). From the Windows start menu (or the Desktop shortcut), run PllDesign tool. Demonstrate the procedure and the results for a PLL design example using this tool. Try to match the design in the "pll_design.pdf" document as much as possible. Enjoy 😊

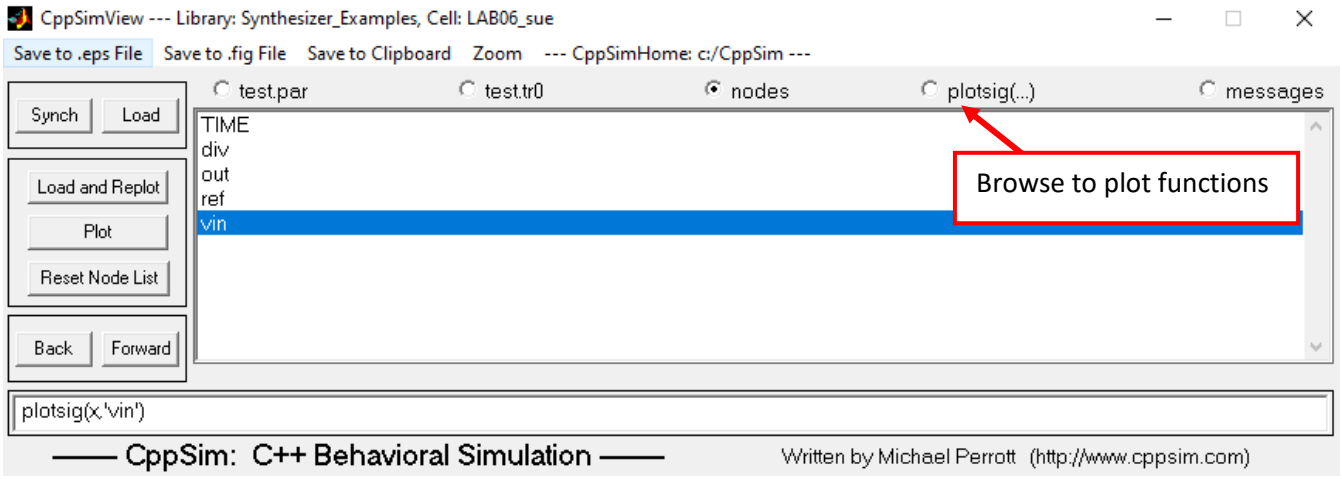
Part 3

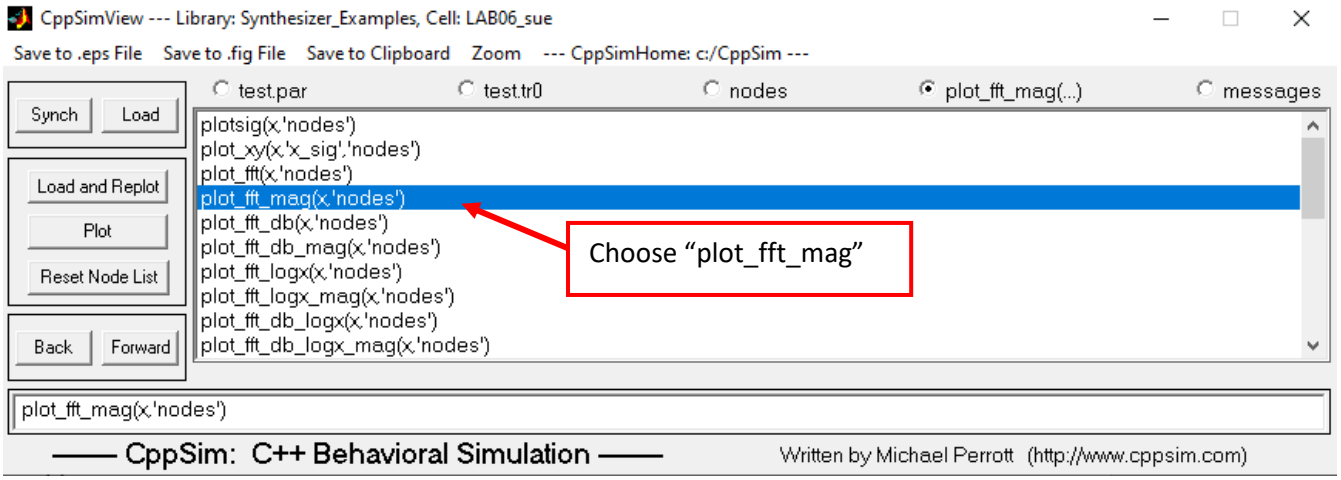
Modify your PLL to obtain a 900MHz clock after locking. Using multiplexer-2 from CppSimModules (use reference clock as the selector) to divide over two different values using the same divider.

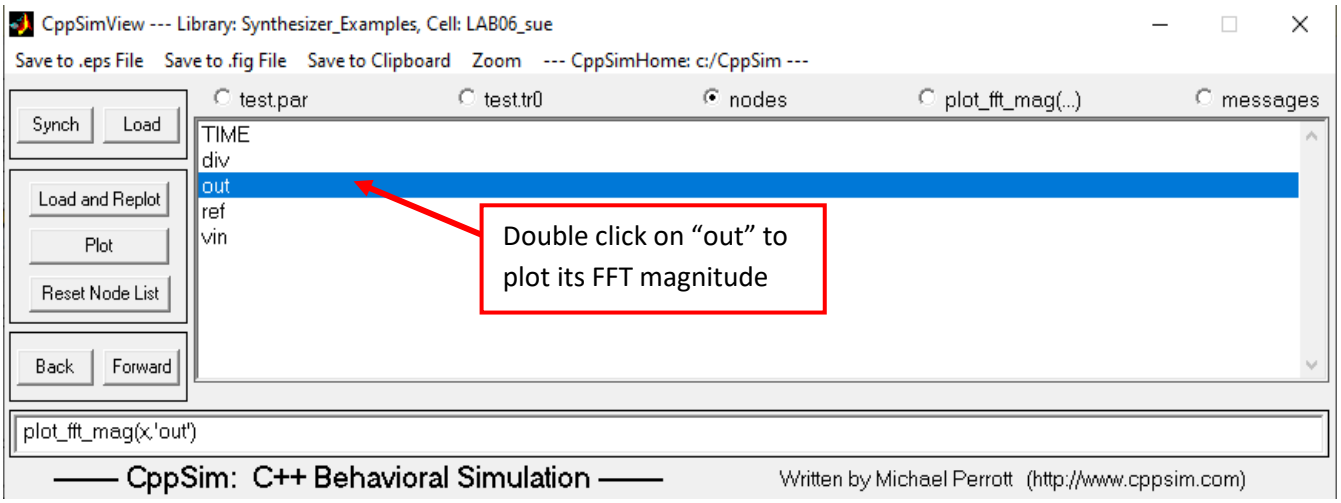


Index	Deliverable
1.	Using kvco from Part 1 and the new divider value, analytically calculate vin at which the PLL will achieve lock.
2.	Snapshot of the simulation results showing vin waveform after locking. Comment.
3.	Use CppSimView plot functions to plot the output spectrum (FFT). Comment. For help, in Sue2 click (Doc -> CppSim/VppSim Primer).

Hint: How to plot FFT







Thanks to all who contributed to these labs. If you find any errors or have suggestions concerning these labs, please contact Hesham.omran@eng.asu.edu.eg.