

# How to Design and Debug a Phase-Locked Loop (PLL) Circuit

By Ray Sun

## Introduction

Designing and debugging a *phase-locked loop* (PLL) circuit can be complicated, unless engineers have a deep understanding of PLL theory and a logical development process. This article presents a simplified methodology for PLL design and provides an effective and logical way to debug difficult PLL problems.

## Simulation

It is difficult to estimate the specifications of a PLL circuit without simulating it at specific conditions, so simulation should be the first step of the PLL design process. We recommend that engineers use ADIsimPLL software to run a simulation based on their system requirements, including *reference frequency*, *step frequency*, *phase noise* (jitter), and *frequency spur* limitations.

Many engineers are confused as to how to choose a reference frequency, but the relationship between the reference frequency and the output frequency step is simple. With an *integer-N* PLL, the output frequency step is equal to the frequency at the input of the *phase-frequency detector* (PFD), which is the reference frequency divided by the reference divider, R. With a *fractional-N* PLL, the output frequency step is equal to the PFD input frequency divided by the MOD value, so you can use a higher reference frequency to obtain a smaller frequency step. When deciding whether to use integer-N or fractional-N, the frequency step can be traded for phase noise, with a lower PFD frequency providing better output frequency resolution, but worse phase noise.

As an example, Table 1 shows that if the requirement calls for a fixed-frequency output with a very large frequency step, an integer-N PLL, such as the ADF4106, is preferable due to its better total in-band phase noise. Conversely, if the requirement calls for a small frequency step, a fractional-N PLL, such as the ADF4153, is preferable because its total noise is better than that of the integer-N PLL. Phase noise is an essential PLL specification, but the data sheet cannot specify performance for all possible applications. Thus, simulation followed by testing actual hardware is crucial.

**Table 1. Phase Noise Determines Choice of PLL**

	Fixed-Frequency Application	GSM1800 Application
	RF = 1.8 GHz, $f_{osc} = 13$ MHz, fixed output frequency	RF = 1.8 GHz, $f_{osc} = 13$ MHz, $f_{res} = 200$ kHz
ADF4106 Integer-N PLL	$FOM + 10\log f_{pfd} + 20\log N = -223 + 10\log 13 \text{ MHz} + 20\log 138 = -109 \text{ dBc/Hz}$	$FOM + 10\log f_{pfd} + 20\log N = -223 + 10\log 200 \text{ kHz} + 20\log 9000 = -91 \text{ dBc/Hz}$
ADF4153 Fractional-N PLL	$FOM + 10\log f_{pfd} + 20\log N = -220 + 10\log 13 \text{ MHz} + 20\log 138 = -106 \text{ dBc/Hz}$	$FOM + 10\log f_{pfd} + 20\log N = -220 + 10\log 13 \text{ MHz} + 20\log 138 = -106 \text{ dBc/Hz}$
Result	Integer-N is better	Fractional-N is better

Even when using ADIsimPLL to simulate a PLL circuit under real conditions, the results will probably be insufficient unless model files for the actual reference and voltage-controlled oscillator (VCO) are included. If not, the simulator will use an ideal reference

and VCO to do a simulation. The time required to edit a library file for a VCO and reference source will be well worth it when high simulation accuracy is required.

PLLs use a negative-feedback control system similar to that of an amplifier, so the concepts of loop bandwidth and phase margin apply here as well. Generally, the loop bandwidth should be set smaller than one-tenth of the PFD frequency, and the safe range for phase margin is from 45° to 60°. In addition, both simulation and the prototyping on a real board should be performed to confirm that the circuit will meet specifications given the parasitic elements on the PCB layout and the tolerances of resistors and capacitors in the loop filter.

Sometimes suitable values of resistors and capacitors are not readily available, so engineers must determine whether other values will work. A small function called BUILT is hidden in the Tools menu of ADIsimPLL. This function transforms the values of resistors and capacitors to the nearest standard engineering value, allowing the designer to rerun the simulation to verify the new values for phase margin and loop bandwidth.

## Registers

ADI PLLs provide many user-configurable options to enable a flexible design environment, but this introduces the challenge of determining the values to store in each register. A convenient solution is to use the evaluation software to set up the register values, even if a PCB isn't connected to the simulator. Afterwards, the setup file can be saved to an .stp file or downloaded to the evaluation board. The simulation results from ADIsimPLL, shown in Figure 1, recommend register values for parameters such as the core current of VCO.

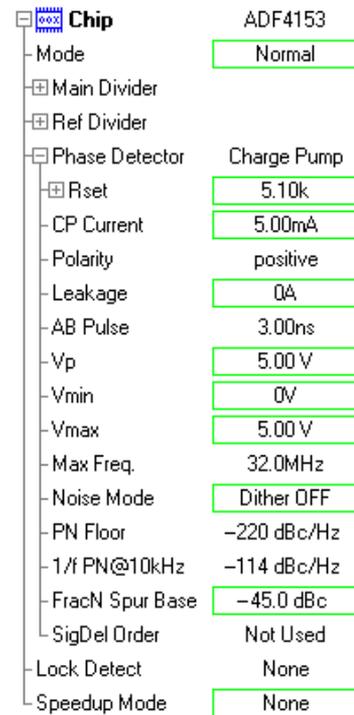


Figure 1. ADIsimPLL simulation software provides recommended values for setup registers.

## Schematics and PCB Layout

Several things should be kept in mind when designing the complete PLL circuit. First, it's important to match the impedance at the reference input port of the PLL to minimize reflections. Also, keep the capacitance in parallel with the input port small, as it will decrease the slew rate of the incoming signal and add noise

to the PLL loop. Refer to the input requirements on the PLL data sheet for more detailed information.

Second, separate the analog and digital power supplies to minimize the interference between them. The VCO supply is particularly sensitive, so spurs and noise here can easily couple into the PLL output. Please refer to [Powering a Fractional-N Voltage Controlled Oscillator \(VCO\) with Low Noise LDO Regulators for Reduced Phase Noise \(CN-0147\)](#) for additional cautions and more detailed information.

Third, the resistors and capacitors used to implement the loop filter should be placed as close as possible to the PLL chip using the values that were recommended by the simulation file. If you have difficulty locking onto a signal after you have changed the values of the loop filter components, please try the original values that were used on the evaluation board.

In terms of the PCB layout, the main principle is to separate the input from the output, making sure that the digital circuitry does not interfere with the analog circuitry. For example, placing the SPI bus too close to the reference input or VCO output will cause spurs on the PLL output when accessing the PLL registers.

From a thermal design perspective, put a thermal ground pad under the PLL chip to ensure that heat flows through the pad to the PCB and heat sink. Designers should calculate all thermal parameters of the PLL chip and PCB when used in extreme environments.

### Effective Use of MUXOUT

At the start of the debug phase, it's difficult to determine where to start when the PLL won't lock. As a first step, use MUXOUT to see if each internal function unit is operating properly, as shown in Figure 2. For example, MUXOUT can show the output of the R-counter to indicate that the reference input signal is fine and that the contents of the register were successfully written. MUXOUT can also check to see the detector's lock status and the output of the N-divider in the feedback loop. In this way, the designer can confirm whether the value of each divider, gain, or frequency is correct. This is the fundamental process of debugging a PLL.

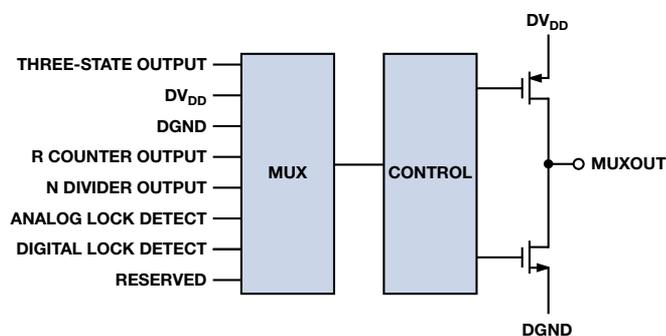


Figure 2. MUXOUT pin aids PLL debugging process.

### Time-Domain Analysis

While debugging a PLL, use time-domain analysis to demonstrate that the data written to the registers on the serial peripheral interface (SPI) bus is correct. Even though reads and writes are not done very quickly, make sure that the SPI timing meets specifications and that crosstalk between different lines is minimized.

The timing diagram of the PLL's data sheet should be consulted to determine data setup time, clock speed, pulse width, and other

specifications. Be sure to leave enough margin to ensure that the timing requirements are met under all conditions. An oscilloscope should be used to check to see that the clock and data edges are in the right position in the time domain. If clock and data lines are too close, crosstalk can cause energy from the clock to couple to the data line via the PCB traces. This coupling results in a glitch on the data line that occurs on the rising edge of the clock. Therefore, look at these two lines when writing or reading the registers, especially if register errors appear. Make sure that the voltages on the lines are satisfied as shown in Table 2.

Table 2. Logic Inputs

	Minimum	Typical	Maximum	Units
Input High Voltage, $V_{INH}$	1.5			V
Input Low Voltage, $V_{INL}$			0.6	V
Input Current, $I_{INH}/I_{INL}$			$\pm 1$	$\mu\text{A}$
Input Capacitance, $C_{IN}$		3.0		pF

### Spectrum Analysis

Issues in the frequency domain are both more frequent and more complex. With a spectrum analyzer, first check to see if the PLL output is locked, as indicated by a stable frequency peak. If not, the tips described above should be followed.

If the PLL is locked, narrow the bandwidth of the spectrum analyzer to determine whether the phase noise is acceptable or not, confirming the test result with the simulation result. Measure the phase noise at several bandwidths, such as 1 kHz, 10 kHz, and 1 MHz, for example.

If the result is not as expected, first review the loop filter design and check the real values of the components on the PCB board. Next, check the reference input to see if its phase noise is the same as the simulation. The simulated phase noise of the PLL should be similar to the real result unless the external conditions are different or the registers were written with the wrong values.

The noise from the power supply should not be ignored, even if a low-noise LDO is used, because both dc-to-dc converters and LDOs look like noise sources. The LDO data sheet usually shows a noise spectrum density that will affect noise-sensitive parts such as PLLs (see Figure 3). Choose a low-noise power source for the PLL, especially to supply the core current of the VCO.

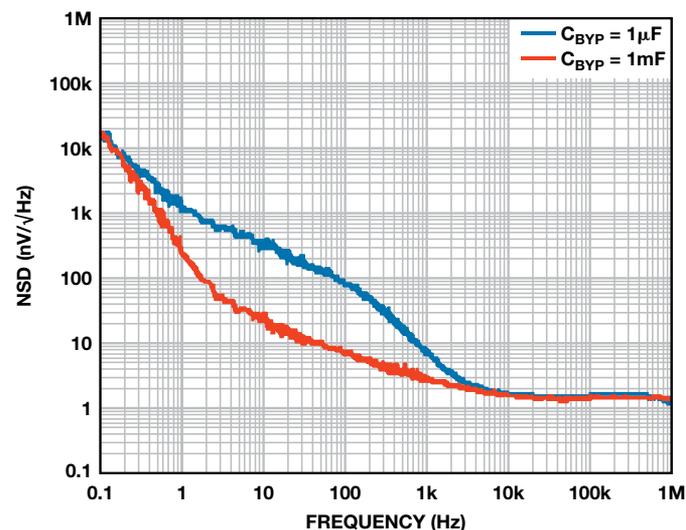


Figure 3. LDO noise spectral density.

Four types of spurs normally appear at the PLL output: PFD or reference spurs, fractional spurs, integer boundary spurs, and spurs from external sources, such as the power supply. All PLLs have at least one type of spur, and although they can never be eliminated, we can sometimes improve overall performance by trading spurs of one type or frequency for another.

To avoid reference spurs, check the rising edge of reference signal. An edge that is too fast or too large in amplitude will cause strong harmonics in the frequency domain. Also, carefully check the PCB layout to avoid crosstalk between input and output.

To minimize fractional spurs, dither can be added to push the fractional spurs into the noise floor, but this will increase the noise floor slightly.

Integer boundary spurs are rare, and occur only if the output frequency is too close to an integer multiple of the reference frequency such that the loop filter fails to eliminate it. An easy way to solve this problem is to readjust the reference frequency plan. For example, if a boundary spur occurs at 1100 MHz with an 1100.1-MHz output, a 20-MHz reference input, and a 100-kHz loop filter, changing the reference frequency to 30 MHz will eliminate the spur.

### Conclusion

The process of debugging a PLL requires a deep understanding of the PLL, and many problems can be avoided by paying close attention during the design phase. If issues occur during the debugging phase, please follow the suggestions given in this article to analyze the issues one by one and tackle them gradually. For more information, please refer to the wealth of relevant information that can be found on [www.analog.com/pll](http://www.analog.com/pll).

### References

Curtin, Mike, and Paul O'Brien. "Phase-Locked Loops for High-Frequency Receivers and Transmitters—Part 1." *Analog Dialogue*, Volume 33, Number 1, 1999.

Curtin, Mike, and Paul O'Brien. "Phase-Locked Loops for High-Frequency Receivers and Transmitters—Part 2." *Analog Dialogue*, Volume 33, Number 1, 1999.

Curtin, Mike, and Paul O'Brien. "Phase-Locked Loops for High-Frequency Receivers and Transmitters—Part 3." *Analog Dialogue*, Volume 33, Number 1, 1999.

CN0147 Circuit Note. *Powering a Fractional-N Voltage Controlled Oscillator (VCO) with Low Noise LDO Regulators for Reduced Phase Noise*.

Fox, Adrian. "PLL Synthesizers (Ask the Applications Engineer—30)." *Analog Dialogue*, Volume 36, Number 3, 2002.

MT-086 Tutorial. *Fundamentals of Phase-Locked Loops (PLLs)*.

### Author

**Ray Sun** [[ray.sun@analog.com](mailto:ray.sun@analog.com)] graduated with a BEng from Wuhan University of Technology in 2002, joined Analog Devices in 2009, and received an MBA from Huazhong University of Science & Technology in 2010. A field applications engineer based in Wuhan, China, Ray currently supports customers in Central China. In his spare time, Ray is a dog trainer, a zoophile, and a charter member of Toastmasters in Wuhan.

