Application Report

SNAA289–January 2016

Using a Programmable Input Multiplier to Minimize Integer Boundary Spurs

Dean Banerjee

ABSTRACT

The concept of using a programmable input multiplier, also nicknamed Spur-b-Gone, allows the spurs of a PLL to be drastically improved, especially for the worst case VCO frequencies. This can be done without changing the input frequency, as this is purely implemented inside the PLL. This feature is included on some TI PLLs, such as the LMX2571.

Contents

1 Brief PLL Overview

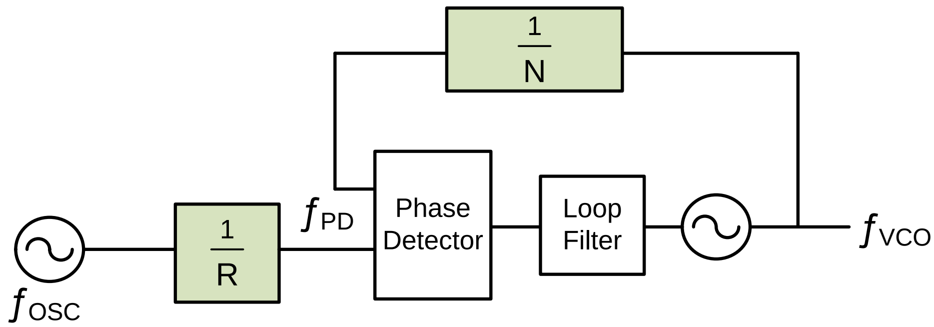

Figure 1. Basic PLL

The PLL (Phased Locked Loop) starts with a fixed frequency (f_{OSC}) and may divide it down by an integer R divider value (R) to get the phase detector frequency (f_{PD}).

$$f_{PD} = \frac{f_{OSC}}{R} \tag{1}$$

The VCO frequency (f_{VCO}) is divided down by the N divider value (N) to also get the phase detector frequency. The N divider may be a fraction. The phase detector compares the N and R divider outputs and puts out correction pulses to the loop filter, which in turn converts this to a voltage. This voltage is converted to a frequency by the Voltage Controlled Oscillator (VCO). The input and output frequencies are related as shown in Equation 2.

$$f_{VCO} = N \times f_{PD} = f_{VCO} \times \frac{N}{R} \tag{2}$$

2 Understanding Integer Boundary Spurs

When the N divider value is not an integer, it is possible to have the integer boundary spur. This spur is typically the most troublesome one and occurs at an offset (f_{spur}) equal to the difference of the VCO Frequency and the closest multiple of the phase detector frequency.

$$f_{Spur} = f_{VCO} \,\%\, f_{PD} \tag{3}$$

For instance, the system in Table 1 would have worst case channels of 2000.1, 2019.9, 2020.1, 2039.9, and 2040.1 MHz. For all these channels, these would be the worst case for the closest integer boundary the spur offset frequency of 100 kHz.

Table 1. Integer Boundary Spur Example

PARAMETER	VALUE
f_{OSC}	20 MHz
f_{PD}	20 MHz
f_{VCO}	2000 – 2050 MHz
Channel Spacing	0.1 MHz

3 Spur-b-Gone Concept

The concept of Spur-b-Gone is to put a multiplier after the input frequency so that the phase detector frequency can be dynamically changed to avoid bad integer boundary spurs.

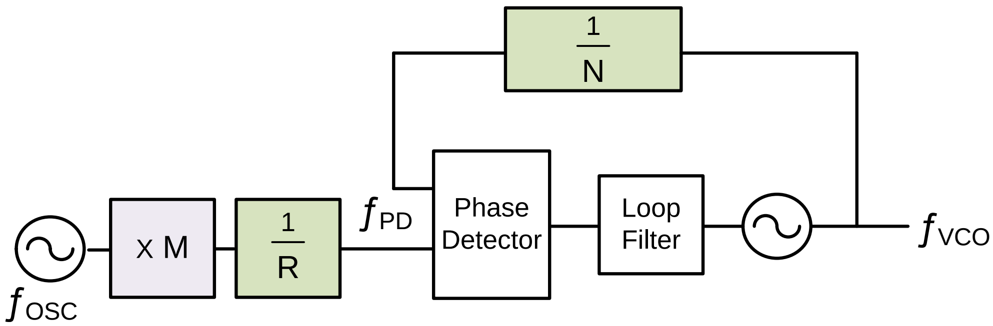

Figure 2. Architecture for Avoiding Spurs

For the example in Table 1, one could change the multiplier value (M) to three and the R divider to two in order to get a 30-MHz phase detector frequency and avoid the integer boundary spurs. It is true that changing the phase detector frequency in this way does change the loop dynamics, but the charge pump can be used to compensate. In this case, when the phase detector is increased from 20 to 30 MHz, which multiplying by 1.5, then the charge pump gain would ideally be divided down by 1.5.

4 Usage of the Programmable Input Multiplier

Consider the example in Table 2, comparing a regular application (no Spur-b-Gone) to an application using a Spur-b-Gone. We see that there is a significant improvement in the VCO integer boundary spur. Also, because this integer boundary spur can mix and be divided down to form other spurs, we also see a ripple effect where other spurs that are derived from this main integer boundary spur at 100 kHz are also reduced.

Table 2. Spur-b-Gone Example

PARAMETER	REGULAR	SPUR-B-GONE
f_{OSC}	20 MHz	20 MHz
R	1	3
M	1	4
f_{PD}	20 MHz	26.66.. MHz
f_{VCO}	5400.1 MHz	5400.1 MHz
Output Frequency	540.01 MHz	540.01 MHz
VCO Integer Boundary Spur at 100 kHz Offset	–63 dbc	–70 dBc
½ VCO Integer Boundary Spur at 50 kHz Offset	–59 dBc	–72 dBc
1/10 Integer Boundary Spur at 10 kHz Offset	–53 dBc	< –72 dBc

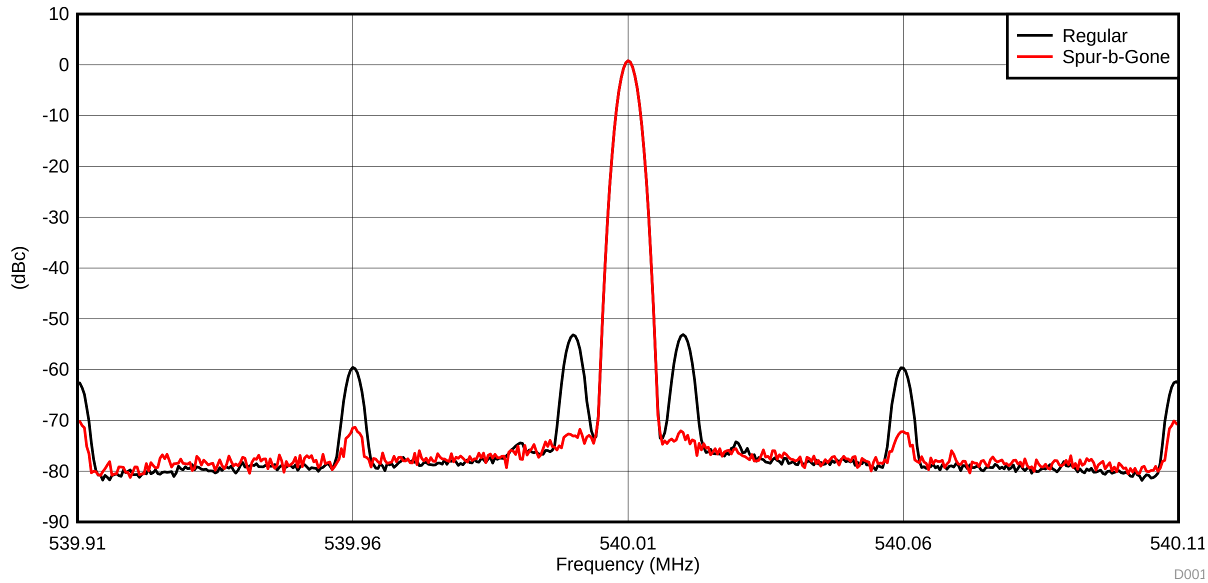

Figure 3. Spur-b-Gone Example

As we see in Figure 3, not only has the VCO integer boundary spur been reduced, but many other spurs are also impacted as this goes through the divider. Note that the charge pump current was adjusted to keep the same loop bandwidth. The loop filter was the default one on the LMX2571 evaluation with maximum charge pump gain for the regular case, but this is not critical, as the key point is the relative change in the spur levels that is achieved by Spur-b-Gone.

5 Understanding Spur-b-Gone in More Depth and How the *Spur-b-Gone* Button Works on CodeLoader

The general concept in using Spur-b-Gone is to maximize the distance from the integer boundaries. However, it also turns out that if one is close to or at the midpoint of two integer channels, then again the spurs will be higher, although not as bad as being right at the integer boundary. Also, as the multiplier value becomes too large, then it also can add noise.

The *Spur-b-Gone* button on CodeLoader works by iterating through all valid PLL_R_PRE, PLL_R and MULT values and calculates an index for each one. Then it chooses the highest (best) index. The index works by putting 40x more weight on the distance to the closest integer channel compared to the midpoint and doing a parallel combination of these. Then it divides by the multiplier value to discourage higher multiplier values unless it provides some benefit.

```
IBS = Fvco % Fpd (Distance to closest integer channel)
IBS2 = Fvco % (Fpd/2) (Distance to closest midpoint between 2 integer channels
If  IBS = 0   then
Index = Infinite (Ideal to be exactly on an integer channel)
Else If  IBS2=0 then
Index = 0  (Stay away from ½ of integer boundary)
Else
If IBS<IBS then
Index = IBS / Mult
Else
 Index = IBS * (40 * IBS2) / (IBS + 40 * IBS2)  / Mult
End If
End If
```