Application Report

Streamline RF Synthesizer VCO Calibration and Optimize PLL Lock Time

Noel Fung

ABSTRACT

When VCO-integrated RF synthesizers were first introduced, the typical implementation was a single VCO-core synthesizer. As a consequence, there were numerous devices with the same PLL but different VCO frequency cores to support different applications. Today's modern RF synthesizers are integrated with several VCO cores and output dividers such that a single device can support very low to very high output frequencies. To ensure all VCO cores work in harmony, proper VCO calibration is required. The calibration is critical to VCO frequency selection and meaningful to the frequency switching time. This application note addresses why VCO calibration is necessary, its theory of operation, and the methodology to minimize the calibration time for the LMX2594 wideband RF synthesizer. This application note applies to the LMX2594, LMX2595 and LMX2615-SP devices.

Table of Contents

List of Figures

List of Tables

Trademarks

All trademarks are the property of their respective owners.

1 Wideband VCO Implementation

An inductor in parallel with a capacitor forms a resonant tank circuit, as shown in Figure 1-1.

If the user replaces the capacitor with a varactor diode and applies a different voltage to the varactor diode, the user can control the resonant frequency.

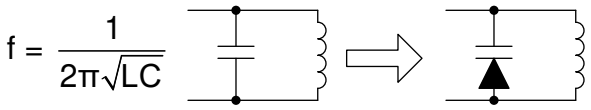

Figure 1-1. LC Resonant Tank Circuit

1.1 Discrete VCO

Figure 1-2 shows a typical Colpitts oscillator that uses a varactor diode to realize frequency tuning. A tuning voltage, Vtune, is used to change the capacitance of the varacter diode, and therefore, change the frequency of the oscillator.

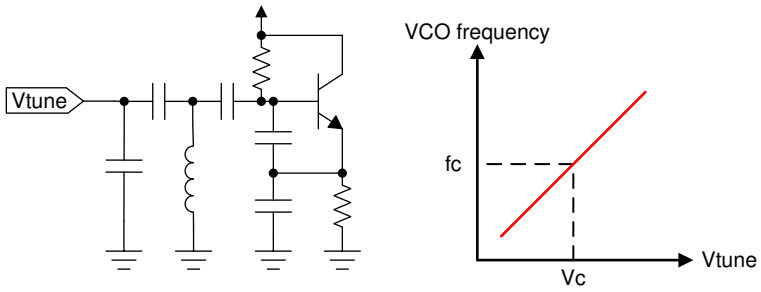

Figure 1-2. Voltage Control Oscillator

Assume when Vtune = Vc, the frequency of the VCO is fc.

If the VCO gain (aka Kvco) is 10 MHz/V, then the corresponding output frequency for Vc ± 1 V is fc ± 10 MHz.

The frequency tuning range of a discrete VCO cannot be very wide. In general, the tuning range is about 10 to 15% of the VCO center frequency.

1.2 Silicon-Based, Switched-Capacitor VCO

If the VCO is built in a chip, it is possible to switch different capacitors in and out of the application to adjust the free running frequency of the VCO.

In other words, with Vtune being fixed, the center frequency of the VCO is adjustable. For example, if C1 is switched in, the corresponding output frequency is f1. As a result, the frequency tuning range of the VCO has been extended from fc ± Vtune to f1 ± Vtune.

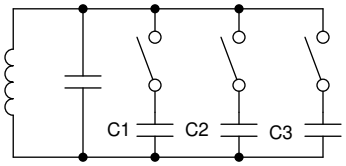

Figure 1-3. Switched-Capacitor Tank Circuit

1.3 VCO Capcode, VCO Core and VCO Amplitude

The term capcode in this document represents the value of the switched capacitors. In LMX2594, the capcode (register R19, VCO_CAPCTRL) is between 0 and 255. The higher the capcode value, the lower is the VCO frequency. To support 7.5-GHz to 15-GHz VCO tuning range, a single switched-capacitor VCO is still not possible. The LMX2594 uses 7 VCO cores (register R20, VCO_SEL) to cover this range. Each VCO has similar capcode structure but different inductor value.

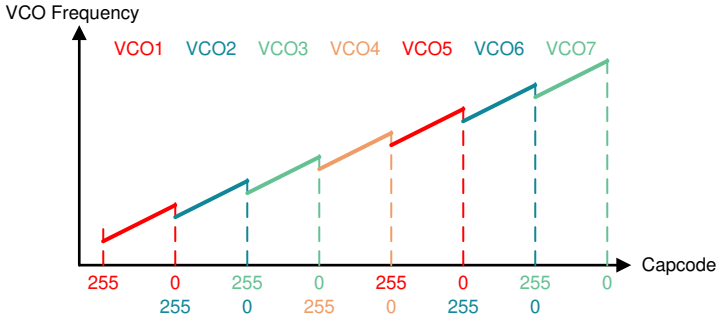

Figure 1-4. VCO Capcode

Note

Each adjacent VCO core has some overlapping regions. This is necessary to ensure seamless VCO core transition and to account for process-voltage-temperature (PVT) variations.

With 256 capcodes and 7 VCO cores, there are altogether 1792 combinations. For a particular frequency, it may be difficult to manually select the right VCO core and capcode, if not impossible. A VCO calibration algorithm is embedded in the chip so that the selection of capcode and VCO core is transparent to the users. In addition to frequency selection, the calibration algorithm also optimizes the phase noise by adjusting the bias of the selected VCO (register R16, VCO_DACISET).

2 VCO Calibration in LMX2594

Although the VCO calibration has been made as user-friendly as possible, take care to ensure that the calibration is robust and returns the optimum performance.

In the coming sections of this application note, the TICS Pro configuration as shown in Figure 2-1 is used as the default configuration for demonstration.

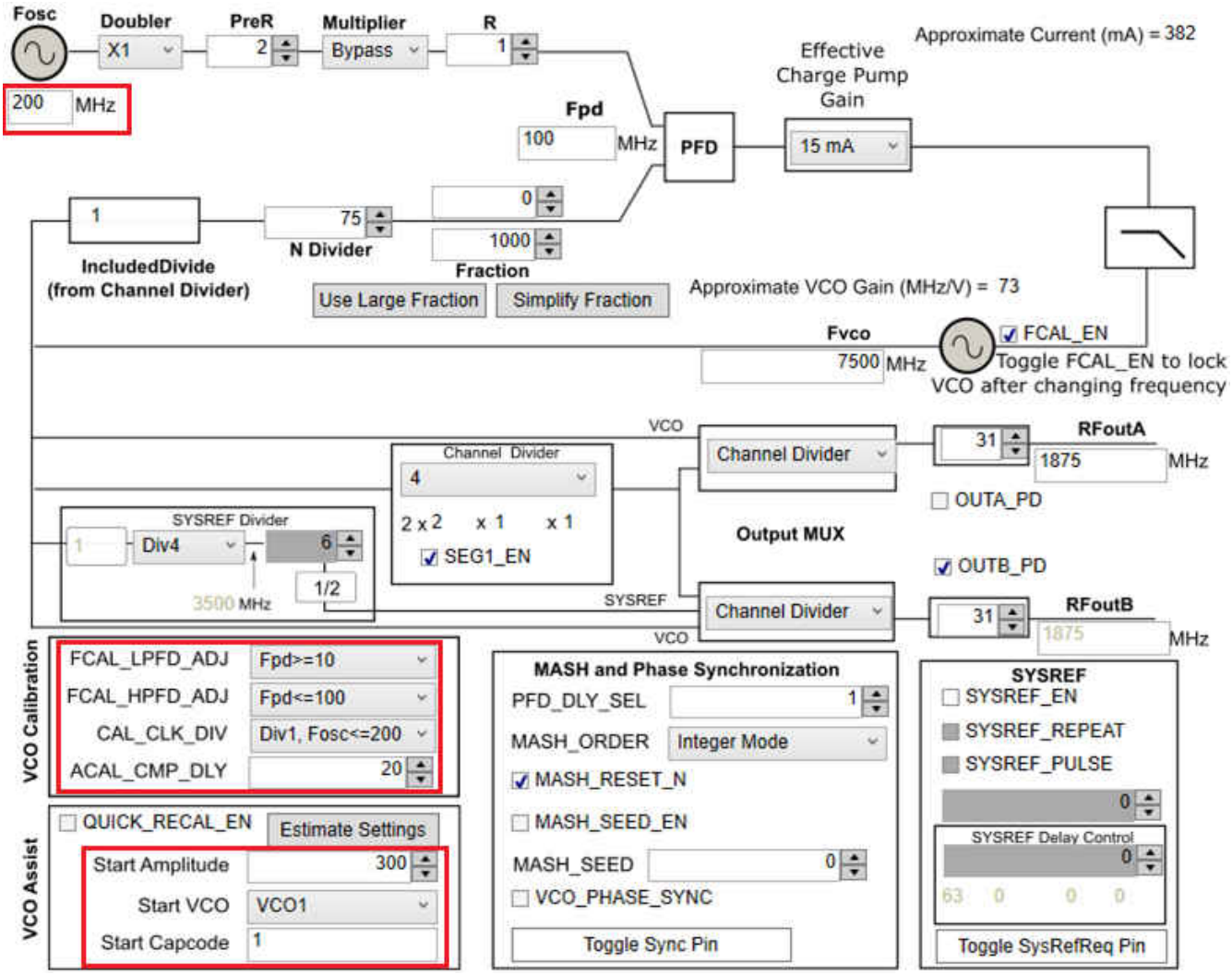

Figure 2-1. Default TICS Pro Configuration

Note the following with regards to the default configuration:

- To provide better illustration, Channel Divider was set to 4 so that the frequency scale is smaller and therefore easier to read.
- To simplify operation, only integer channels are used for demonstration.
- By default, the LMX2594 will scan the VCO core in ascending order.

2.1 Calibration Speed

The speed of calibration will affect the accuracy of a VCO calibration most. Of course, the speed will also impact on the calibration time directly. The highlighted items in Figure 2-1 should be set properly to balance between the calibration time and accuracy. How these parameters affect the calibration behavior is shown in the following sections.

2.1.1 State Machine Clock

The VCO calibration of the LMX2594 runs on state machine clock frequency, f_{SM}. The clock source of the state machine clock comes from the reference clock. Maximum f_{SM} is 200 MHz. Register R1, CAL_CLK_DIV, is used to set this frequency.

$$f_{SM} = f_{OSC} / 2^{CAL\_CLK\_DIV} \quad (1)$$

In the following plots, the VCO jumped from 7500 MHz to 15000 MHz with a different state machine clock frequency. Programming sequence is:

1. Program register R36, PLL_N, from 75 to 150 (decimal).
2. Program register R0 once to trigger the VCO calibration.

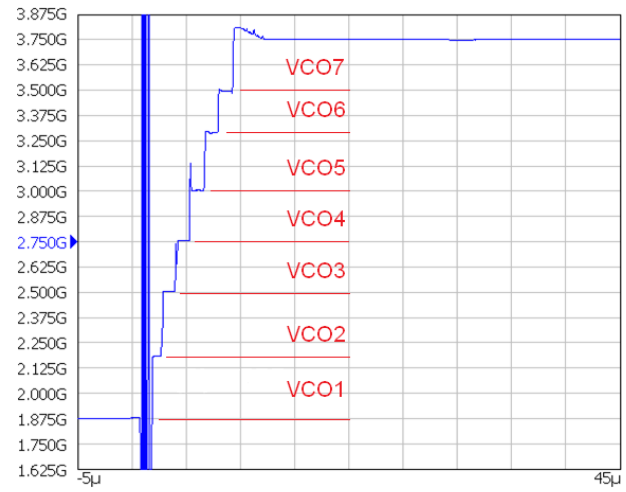

Figure 2-2. f_{SM} = 200 MHz

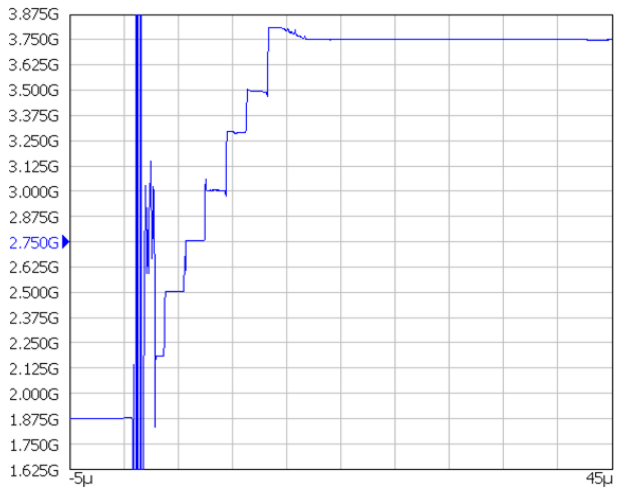

Figure 2-3. f_{SM} = 100 MHz

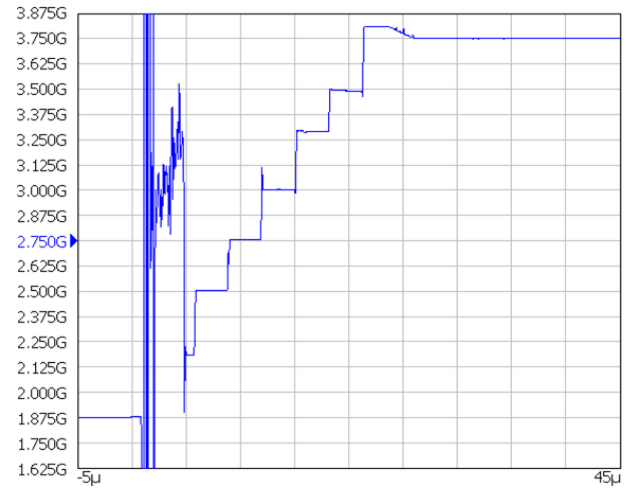

Figure 2-4. f_{SM} = 50 MHz

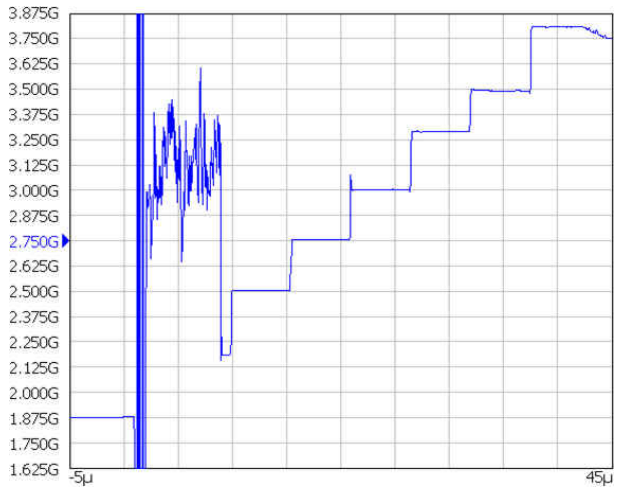

Figure 2-5. f_{SM} = 25 MHz

As shown above, the higher the f_{SM}, the shorter the calibration time. If the frequency switching time is important in certain applications, a higher reference clock frequency source is needed.

2.1.2 FCAL_xxxx_ADJ

In Register R0, FCAL_LPFD_ADJ and FCAL_HPFD_ADJ are used to adjust the calibration speed when the phase detector frequency, f_{PD}, is too small or too high, respectively.

In most of the use cases, f_{PD} will not go below 10 MHz, so FCAL_LPFD_ADJ is basically not used at all.

On the contrary, if f_{PD} is too high, FCAL_HPFD_ADJ should be set properly to slow down the calibration speed.

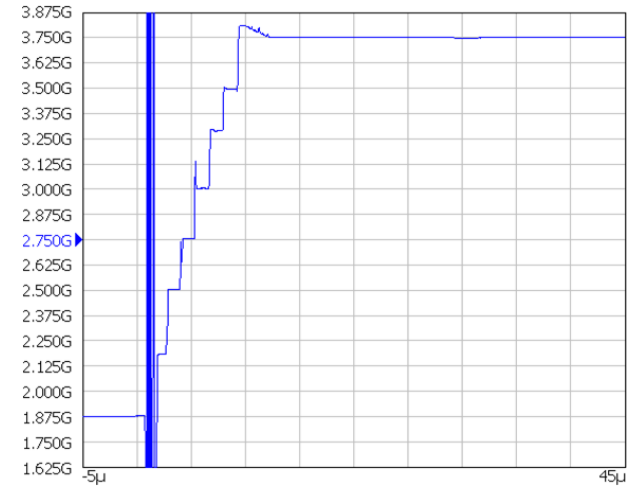

Figure 2-6. FCAL_HPFD_ADJ = 0x0 (f_{PD} ≤ 100 MHz)

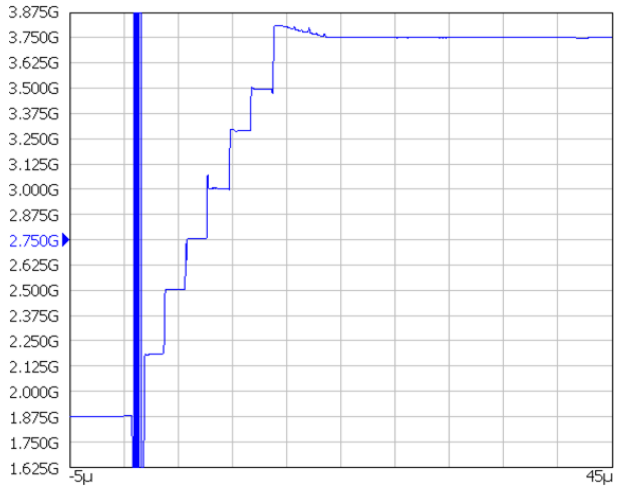

Figure 2-7. FCAL_HPFD_ADJ = 0x1 (100 < f_{PD} ≤ 150 MHz)

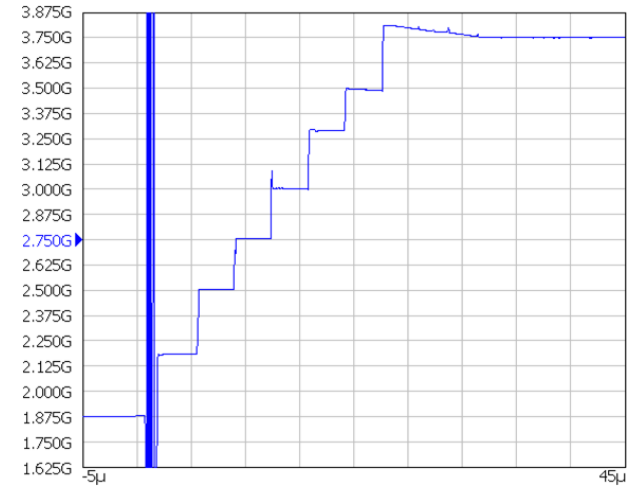

Figure 2-8. FCAL_HPFD_ADJ = 0x2 (150 < f_{PD} ≤ 200 MHz)

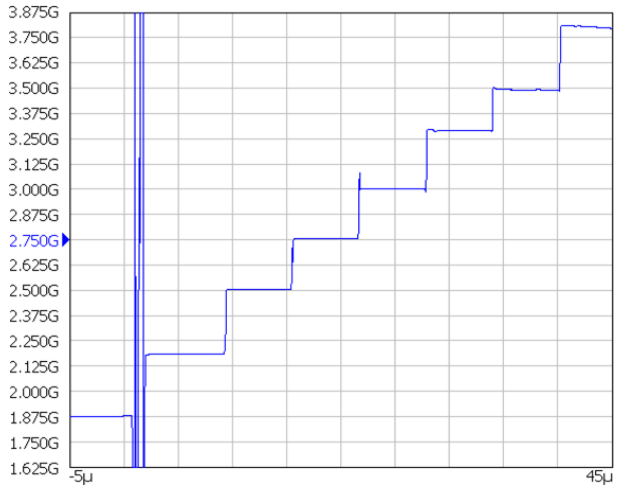

Figure 2-9. FCAL_HPFD_ADJ = 0x3 (f_{PD} > 200 MHz)

The state machine clock frequency remains unchanged here, but the calibration speed is reduced with a higher value of FCAL_HPFD_ADJ.

2.1.3 ACAL_CMP_DLY

In Register R4, ACAL_CMP_DLY is a delay added for VCO amplitude calibration. During amplitude calibration, the bias voltage of the VCO will be adjusted step-by-step until the best phase noise is obtained. This bias voltage comes from an internal LDO. It takes time for the LDO voltage to settle down, therefore a delay is necessary for every step voltage change. Lowering this value can speed up VCO calibration, but lowering it too much may degrade VCO phase noise. The minimum allowable value for this field is 10 to allow the VCO to calibrate to the correct frequency for all scenarios. To yield the best and most repeatable VCO phase noise, use Equation 2 to find the ACAL_CMP_DLY value.

$$ACAL\_CMP\_DLY > f_{SM} / 10\ MHz \quad (2)$$

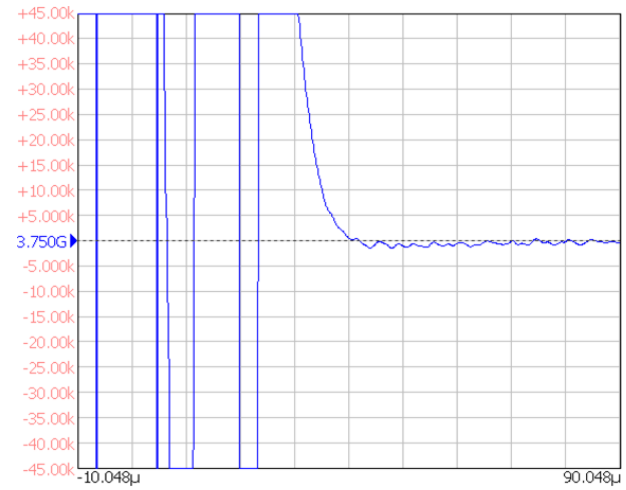

Figure 2-10. ACAL_CMP_DLY = 10, Jump Up

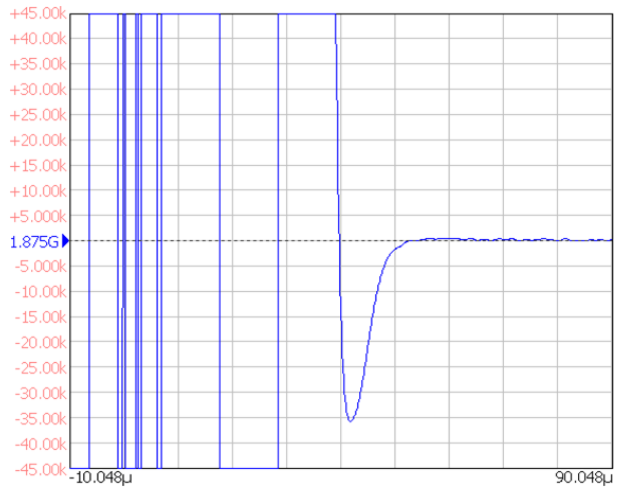

Figure 2-11. ACAL_CMP_DLY = 10, Jump Down

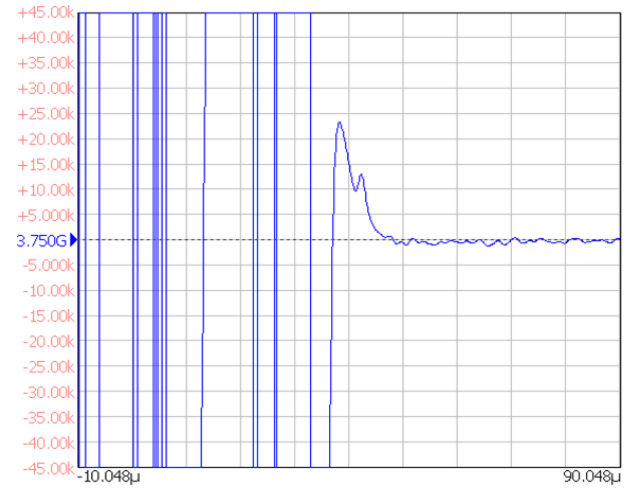

Figure 2-12. ACAL_CMP_DLY = 20, Jump Up

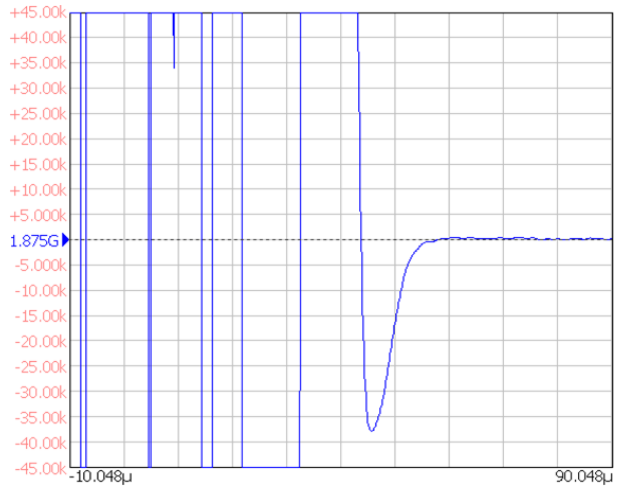

Figure 2-13. ACAL_CMP_DLY = 20, Jump Down

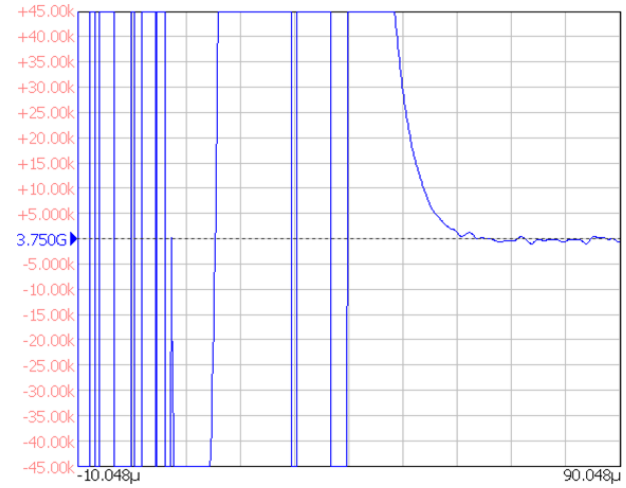

Figure 2-14. ACAL_CMP_DLY = 40, Jump Up

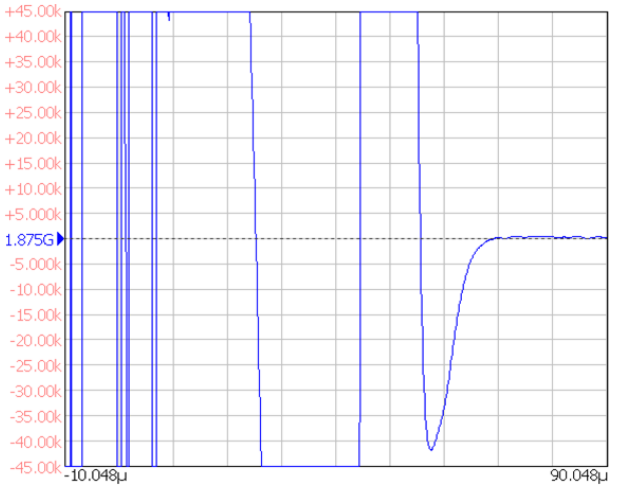

Figure 2-15. ACAL_CMP_DLY = 40, Jump Down

On the left column of the above plots, the VCO jumped from 7500 MHz to 15000 MHz. On the right column, the VCO frequency is flipped.

The state machine clock used in these examples is 200 MHz. Equation 2 suggests ACAL_CMP_DLY = 20. From the plots, the total lock time is increased by approximately 10 µs, compared with the minimum ACAL_CMP_CAL requirement of 10.

2.2 Initial Calibration Parameters Setting

VCO calibration will always start at a pre-defined VCO core, VCO capcode and amplitude setting, no matter what is the present setting. These settings will affect the calibration time.

2.2.1 VCO_SEL

In Register R20, VCO_SEL defines the VCO core that is being used at the beginning of a VCO calibration. The following examples compare the calibration time with different VCO_SEL settings, with the VCO jumping upward and downward.

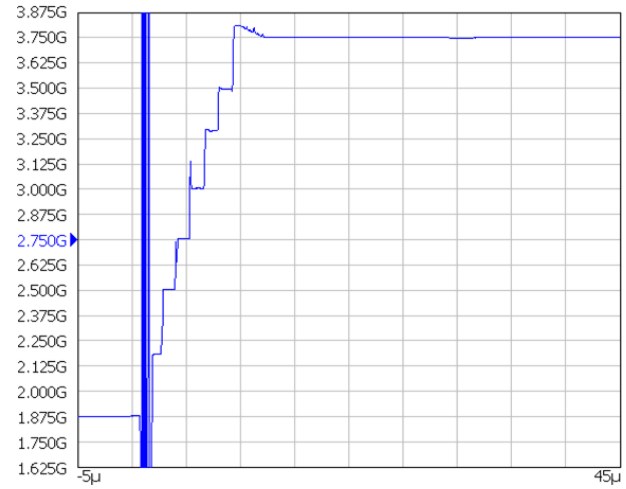

Figure 2-16. VCO_SEL = 1, Jump Up

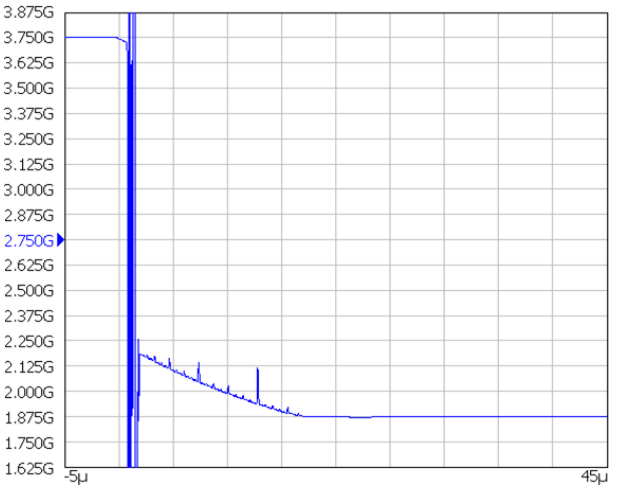

Figure 2-17. VCO_SEL = 1, Jump Down

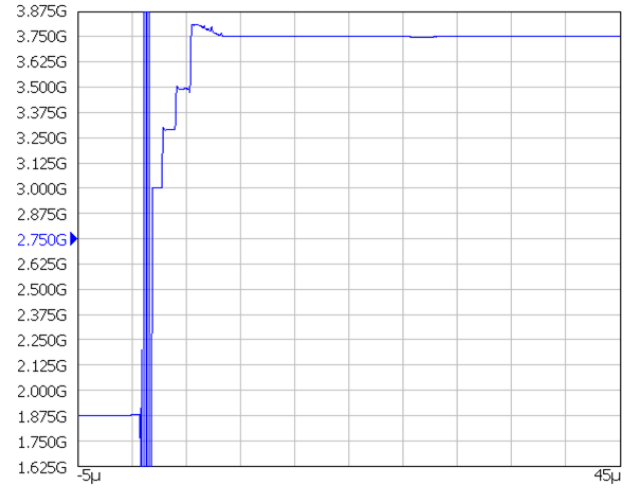

Figure 2-18. VCO_SEL = 4, Jump Up

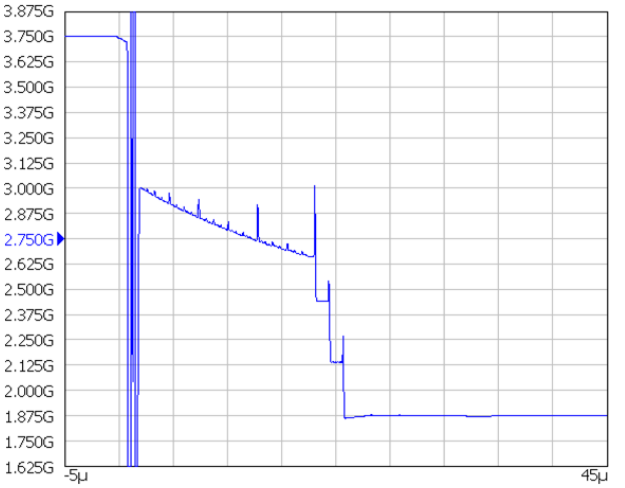

Figure 2-19. VCO_SEL = 4, Jump Down

VCO_SEL affects both jump up and jump down calibration time. In the above examples, jump up time is shorter if we use VCO4 as the starting VCO. However, the jump down time is longer than when VCO1 is being used as the starting VCO. There may be benefits to set it to either VCO1 or VCO7 in some use cases, a well-balanced approach should make it start at the middle of the total VCO coverage range.

2.2.2 VCO_CAPCTRL_STRT

Register R78, VCO_CAPCTRL_STRT sets the initial capcode value being used for VCO calibration. Capcode values are between 0 and 255. 0 represents the highest VCO frequency setting while 255 drives the VCO to the lowest frequency boundary.

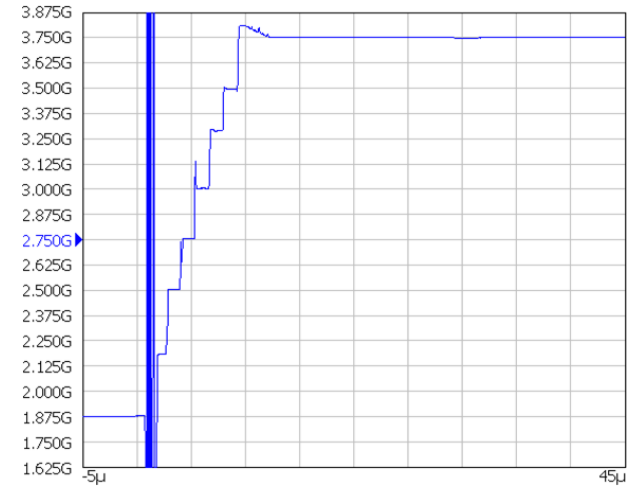

Figure 2-20. VCO_CAPCTRL_STRT = 1, Jump Up

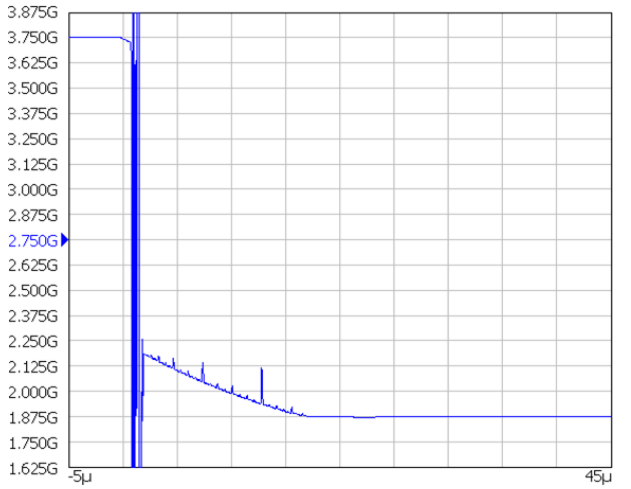

Figure 2-21. VCO_CAPCTRL_STRT = 1, Jump Down

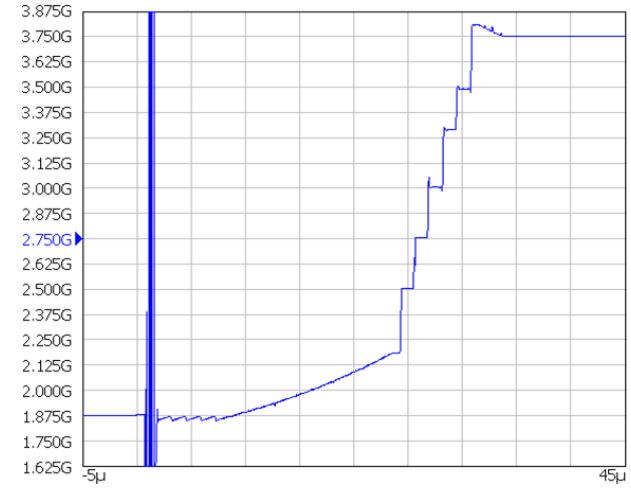

Figure 2-22. VCO_CAPCTRL_STRT = 255, Jump Up

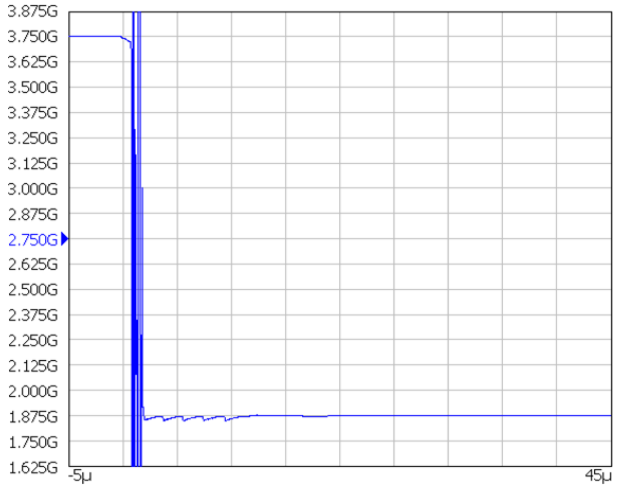

Figure 2-23. VCO_CAPCTRL_STRT = 255, Jump Down

Due to the behavior of the VCO calibration algorithm, the calibration time will be longer in general if the calibration starts with capcode = 255.

2.2.3 VCO_DACISET_STRT

In Register R17, VCO_DACISET_START sets the VCO amplitude value for use in the beginning of VCO calibration. This register has relatively less impact to the calibration time.

2.2.4 Recommended Initial Calibration Parameter Settings

To compromise jump up and jump down behaviors, the following initial calibration parameters are recommended:

- VCO_DACISET_STRT = 300
- VCO_SEL = 4
- VCO_CAPCTRL_STRT = 1

With the suggested initial calibration parameters, here are some examples of how the VCO can jump up and down. The frequency of the VCO core in the following plots are:

- VCO1 = 7500 MHz.
- VCO3 = 10 GHz.
- VCO5 = 12.4 GHz.
- VCO7 = 15 GHz.

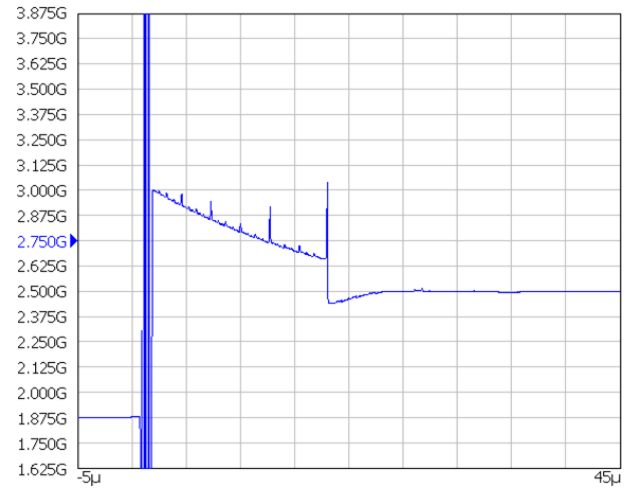

Figure 2-24. VCO1 Jumps to VCO3

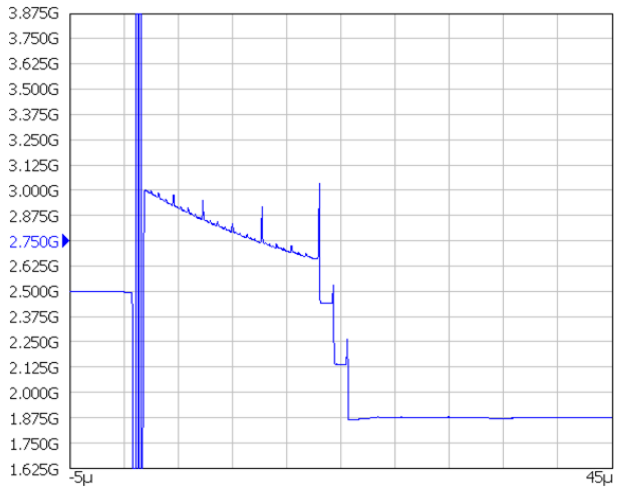

Figure 2-25. VCO3 Jumps to VCO1

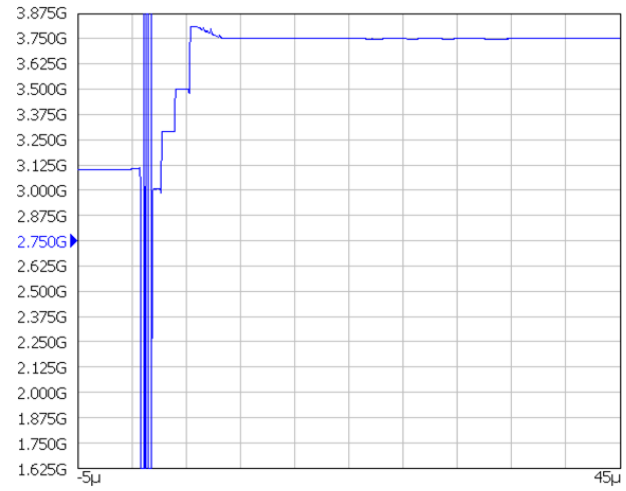

Figure 2-26. VCO5 Jumps to VCO7

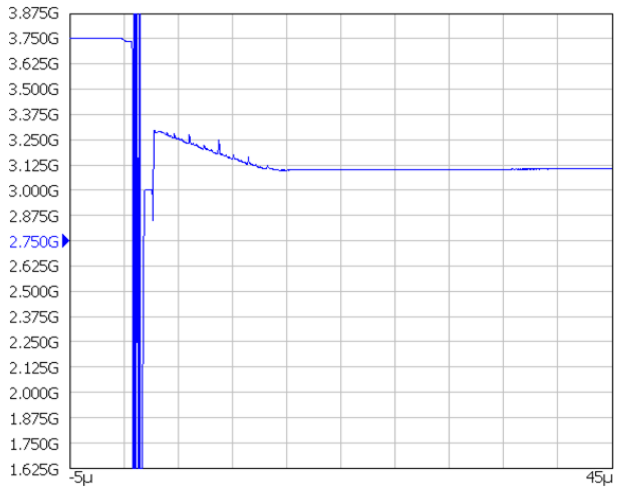

Figure 2-27. VCO7 Jumps to VCO5

3 Reducing Calibration Time

Using the recommended initial calibration parameters, and with properly calibration speed setting, the VCO algorithm will pick the correct VCO core at the right capcode and amplitude setting. However, a sufficient amount of time must be reserved for the VCO calibration routine to complete. To have a control to the calibration time or wants to reduce the calibration time substantially, use partial assist or full assist mode in VCO calibration. With a shorter VCO calibration time, the total VCO frequency switching time will be smaller and therefore it is suitable for use in frequency hopping or sweeping system.

3.1 SPI Programming Speed

In partial assist or full assist mode, in addition to the frequency related registers, such as N-divider and fractional numerator that must be updated, several additional registers need programming. Altogether, there are at least five registers that must be programmed. A register is 24 bits (DATA) + 1 bit (CSB) long, therefore, it will take 25 µs to write a register for a 1-MHz SPI clock rate. Furthermore, the functions or values associated with a register are effective right after the register is written. If the N-divider is updated, for example, the VCO frequency drifts immediately and the PLL loop unlocks while the user is writing the R0 register. All of these changes take place before the VCO starts calibration.

The PLL loop is still closing, and the charge pump voltage goes to the rails as a result. This may affect the PLL lock time when the loop is closed again after the VCO calibration is complete, because the capacitors in the loop filter may take more time to charge up or discharge.

The following plots show the benefits of using a higher SPI clock rate. The VCO switched from VCO5 (12.4 GHz) to VCO7 (14.8 GHz) with the channel divider CHDIV = 8. The N-divider was updated first, followed by register R0, to trigger VCO calibration. As seen in Figure 3-2, even with a SPI clock rate of 2.5 MHz, a register write takes 10 µs. The VCO, meanwhile, already drifted by approximately 80 MHz.

TI recommends to use a higher SPI clock rate whenever possible.

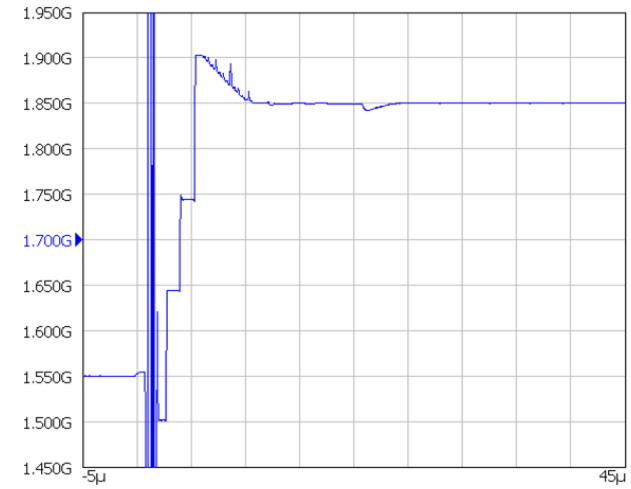

Figure 3-1. SPI CLK = 25 MHz

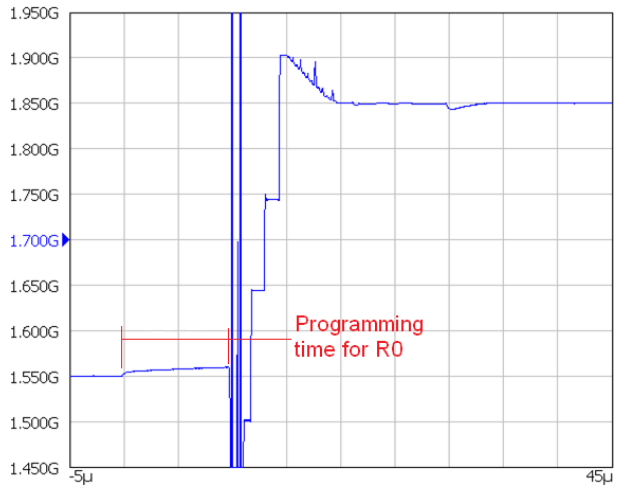

Figure 3-2. SPI CLK = 2.5 MHz

3.2 Partial Assist

In this mode, instead of using the initial calibration parameters as suggested in Section 2.2.4, the user provides the initial starting point for the VCO core (VCO_SEL), capcode (VCO_CAPCTRL_STRT), and amplitude (VCO_DACISET_STRT) based on Table 3-1 on every frequency change before writing to R0 to trigger VCO calibration. To do the partial assist, follow this procedure:

1. Determine the VCO core.
   - Find a VCO core that includes the desired VCO frequency. If at the boundary between two cores, choose one based on phase noise or performance.
2. Use Equation 3 to calculate the VCO capcode.

$$VCO\_CAPCTRL\_STRT = round\left[C_{CoreMin} - (C_{CoreMin} - C_{CoreMax}) \times (f_{VCO} - f_{CoreMin}) / (f_{CoreMax} - f_{CoreMin})\right] \quad (3)$$

3. Use Equation 4 to calculate the VCO amplitude setting.

$$VCO\_DACISET\_STRT = round\left[A_{CoreMin} - (A_{CoreMin} - A_{CoreMax}) \times (f_{VCO} - f_{CoreMin}) / (f_{CoreMax} - f_{CoreMin})\right] \quad (4)$$

Table 3-1. LMX2594 VCO Core Ranges

VCO CORE	$f_{CoreMin}$	$f_{CoreMax}$	$C_{CoreMin}$	$C_{CoreMax}$	$A_{CoreMin}$	$A_{CoreMax}$
VCO1	7500	8600	164	12	299	240
VCO2	8600	9800	165	16	356	247
VCO3	9800	10800	158	19	324	224
VCO4	10800	12000	140	0	383	244
VCO5	12000	12900	183	36	205	146
VCO6	12900	13900	155	6	242	163
VCO7	13900	15000	175	19	323	244

Here is an example.

To jump from VCO1 (7500 MHz) to VCO3 (10 GHz), program the following registers:

- R36 = 0x240064 (PLL_N = 100)
- R20 = 0x14D848 (VCO_SEL = 3)
- R78 = 0x4E0105 (VCO_CAPCTRL_STRT = 130)
- R17 = 0x110130 (VCO_DACISET_STRT = 304)
- R0 = 0x00241C (to trigger calibration)

To jump from VCO3 (10 GHz) to VCO1 (7500 MHz), program the following registers:

- R36 = 0x24004B (PLL_N = 75)
- R20 = 0x14C848 (VCO_SEL = 1)
- R78 = 0x4E0149 (VCO_CAPCTRL_STRT = 164)
- R17 = 0x11012B (VCO_DACISET_STRT = 299)
- R0 = 0x00241C (to trigger calibration)

The test results are shown in Figure 3-7 and Figure 3-8.

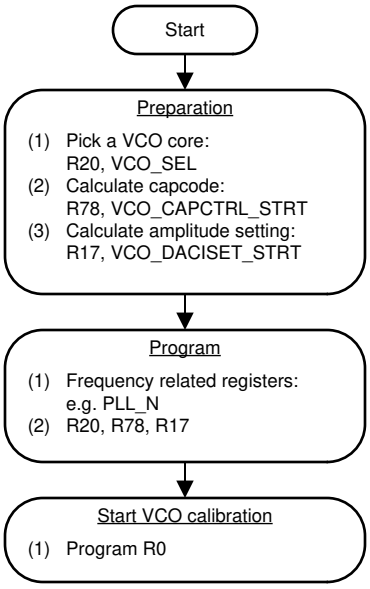

Figure 3-3. Partial Assist Workflow

3.3 Full Assist

In this mode, VCO calibration is completely bypassed. The user forces the VCO core (VCO_SEL), amplitude settings (register R16, VCO_DACISET), and capcode (register R19, VCO_CAPCTRL), and manually sets the value. To get these values for a particular VCO frequency, the user must run an automatic calibration in advance. The user sets an automatic calibration for this frequency, then reads back these values in register R110, rb_VCO_SEL; R111, rb_VCO_CAPCTRL and R112, rb_VCO_DACISET. To use these values, the user must enable register R20, VCO_SEL_FORCE as well as the R8, VCO_CAPCTRL_FORCE and VCO_DACISET_FORCE bits.

Here is an example, assuming VCO_SEL_FORCE, VCO_CAPCTRL_FORCE and VCO_DACISET_FORCE are all equal 1.

To jump from VCO1 (7500 MHz) to VCO3 (10 GHz), program the following registers:

- R36 = 0x240064 (N = 100)
- R20 = 0x14DC48 (VCO_SEL = 3)
- R19 = 0x132787 (VCO_CAPCTRL = 135)
- R16 = 0x10012E (VCO_DACISET = 302)

To jump from VCO3 (10 GHz) to VCO1 (7500 MHz), program the following registers:

- R36 = 0x24004B (N = 75)
- R20 = 0x14CC48 (VCO_SEL = 1)
- R19 = 0x1327AB (VCO_CAPCTRL = 171)
- R16 = 0x10012C (VCO_DACISET = 300)

The test results are shown in Figure 3-9 and Figure 3-10.

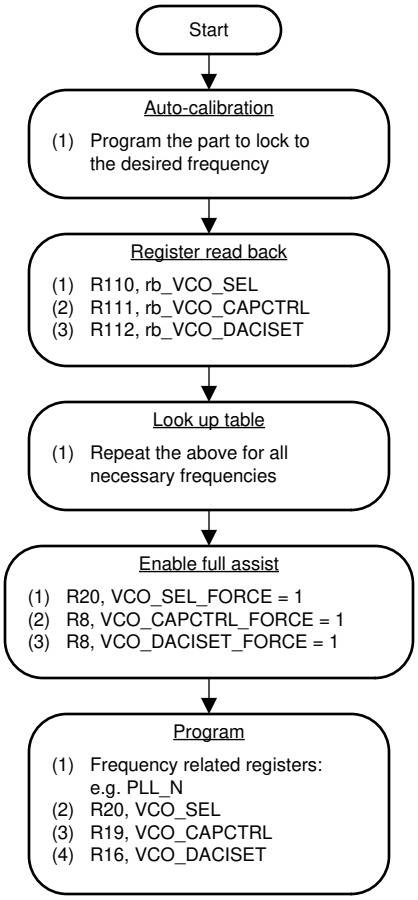

Figure 3-4. Full Assist Workflow

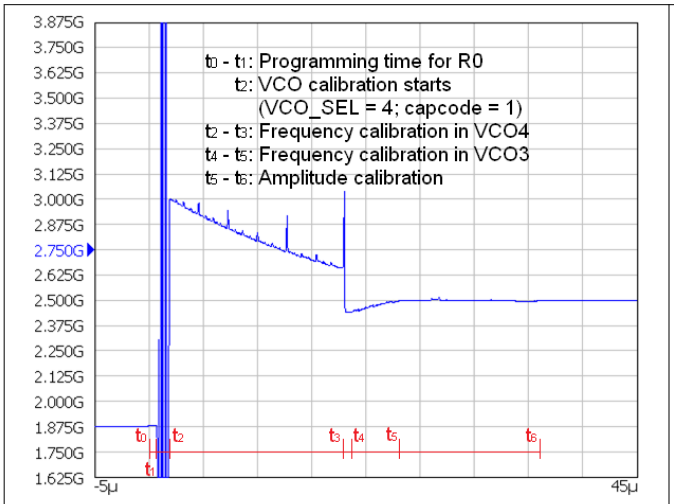

Figure 3-5. Auto-Calibration, Jump Up

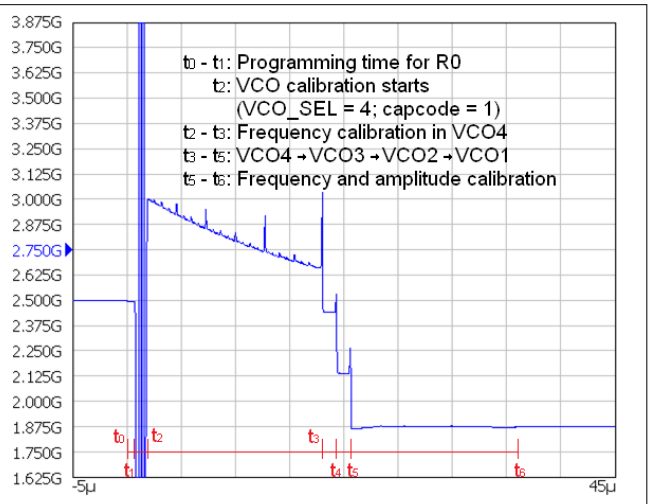

Figure 3-6. Auto-Calibration, Jump Down

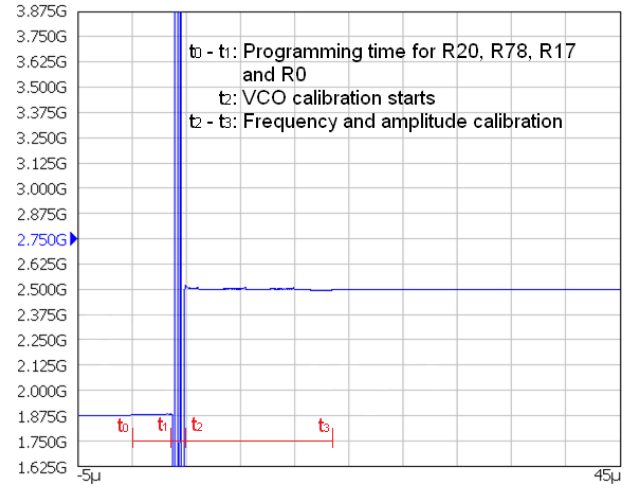

Figure 3-7. Partial Assist, Jump Up

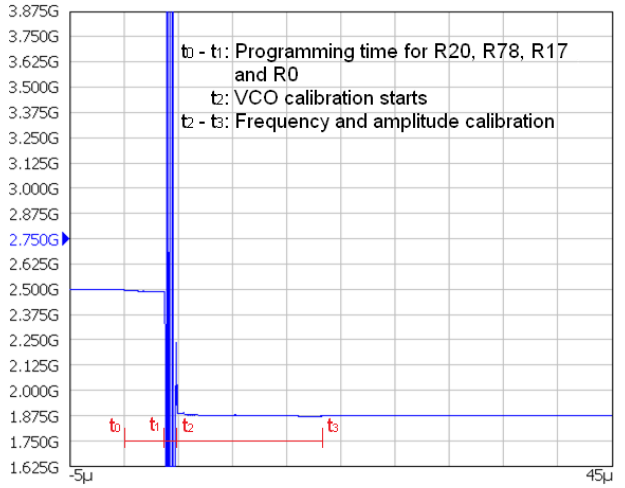

Figure 3-8. Partial Assist, Jump Down

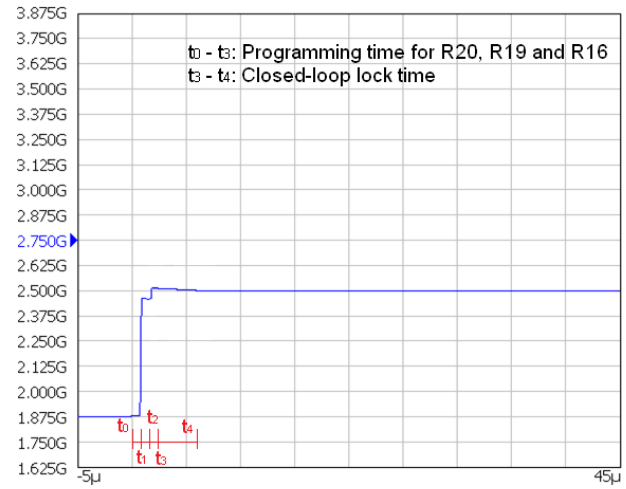

Figure 3-9. Full Assist, Jump Up

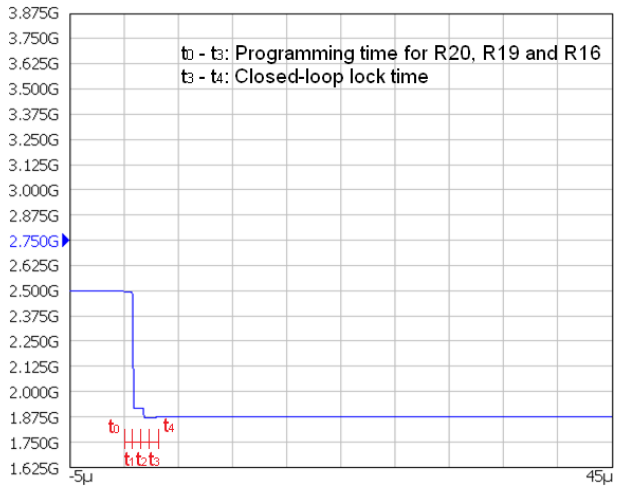

Figure 3-10. Full Assist, Jump Down

4 Summary

With auto-calibration, the calibration time varies but without hassle. To reduce the calibration time, the user can use the listed methods in this document to provide specific calibration parameters. SPI programming speed is also important and should be set fast enough to reduce the unlock duration when the VCO is drifting before the calibration starts.

5 References

Related products information and tools are available in the following links:

- LMX2594
- LMX2595
- LMX2615-SP
- TICS Pro software
- PLLatinum Simulator Tool (PLL Sim)

6 Revision History

NOTE: Page numbers for previous revisions may differ from page numbers in the current version.

Changes from Revision * (January 2020) to Revision A (July 2021)	Page
• Updated the numbering format for tables, figures and cross-references throughout the document	3

IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATA SHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES “AS IS” AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS AND IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, regulatory or other requirements.

These resources are subject to change without notice. TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource. Other reproduction and display of these resources is prohibited. No license is granted to any other TI intellectual property right or to any third party intellectual property right. TI disclaims responsibility for, and you will fully indemnify TI and its representatives against, any claims, damages, costs, losses, and liabilities arising out of your use of these resources.

TI’s products are provided subject to TI’s Terms of Sale or other applicable terms available either on ti.com or provided in conjunction with such TI products. TI’s provision of these resources does not expand or otherwise alter TI’s applicable warranties or warranty disclaimers for TI products.

TI objects to and rejects any additional or different terms you may have proposed.

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265