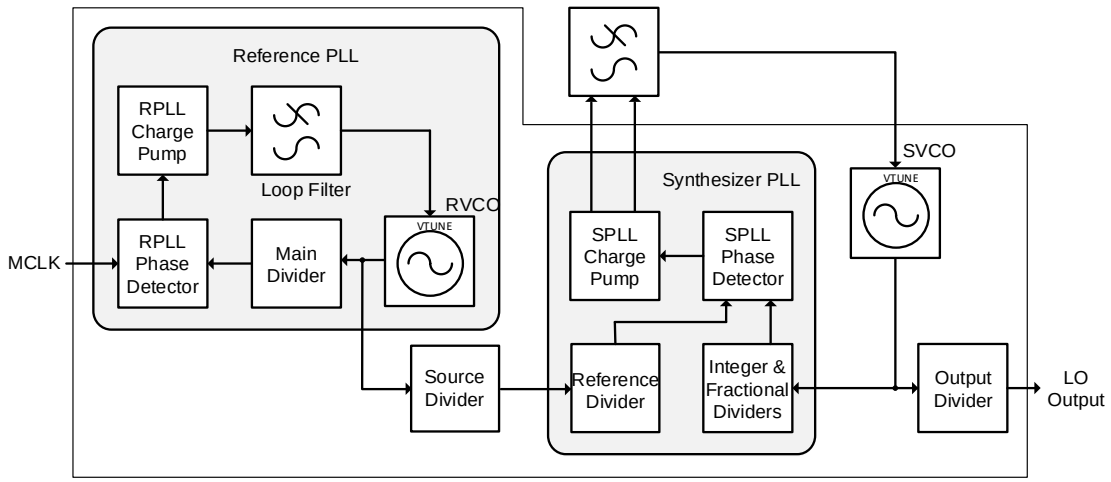
Additional Resources	PE0003/EV9400 Script – Please contact CML Technical Support

1 Introduction

This application note details a PE0003 script that demonstrates the programming steps required to achieve fast tuning and lock of the CMX940 REF_PLL and SPLL synthesizers (whilst coupled to the internal RVCO and SVCO oscillators) without the use of the internal calibration process. This reduces the setup time to retune and achieve lock which is important in frequency hopping applications.



2 History

Version	Changes	Date
1.0	First published	07-02-2022
2.0	Editorial clarifications	25-10-22

3 REF_PLL/SPLL Calibration

3.1 To achieve reliable locking of the Reference PLL (REF_PLL) and the Synthesizer PLL (SPLL) whilst utilising their associated on-chip VCOs, an internal automatic calibration process may be invoked which adjusts various PLL and oscillator parameters to achieve satisfactory performance. These parameters will vary with SPLL programmed frequency and over temperature. The SPLL and internal SVCO are calibrated together; and the calibration process may take several milliseconds to complete.

If the CMX940 is employed in a critical time-to-lock application, such as frequency hopping or channel scanning, the calibration time may exceed the requirements of the application.

With prior knowledge of the required parameter values for each desired frequency in the hop sequence the CMX940 allows the PLL and oscillators to be directly programmed with these values, therefore removing the need for the calibration process and thus reducing the tuning time and achieving lock for each frequency.

This application note assumes that the calibrated parameter values were determined previously by pre-calibrating the CMX940 in the individual end device in which it is to be used and then recorded.

3.2 Following a successful REF_PLL and SPLL lock phase using the automatic calibration function, the relevant parameter values for the wanted target frequency may be read back from the following C-BUS read registers. They are:

REF_RVCO_FREQ_CAL_RD ($DB) – 16 bit
REF_RVCO_BIAS_CAL_RD ($D9) – 8 bit
REF_RVCO_AMP_CAL_RD ($D6) – 8 bit

SVCO_FREQ_CAL_RD ($93) – 16 bit
SVCO_AMP_CAL_RD ($B7) – 16 bit

These values may then be re-written to the CMX940 when a change to this specific target frequency is required. It is worth noting that the calibration values will vary between devices. The associated write registers are:

REF_RVCO_FREQ_CAL ($0B) – 16 bit
REF_RVCO_BIAS_CAL ($09) – 8 bit
REF_RVCO_AMP_CAL ($2C) – 8 bit

SVCO_FREQ_CAL ($53) – 16 bit
SVCO_AMP_CAL ($67) – 16 bit

3.3 During operation the frequency calibration values may vary across time and temperature. The Status Registers REF_DEVICE_STATUS_RD ($83) and DEVICE_STATUS_RD ($C4) may be read to determine if the REF_RVCO_FREQ_CAL or SVCO_FREQ_CAL codes should be modified to compensate for any frequency drift.

For example, if the RVCO output and therefore the REF_PLL frequency slows, the Vtune voltage will drift. In this example the event will be indicated by the setting of the RVCO_CAL_HI b1 in the REF_DEVICE_STATUS_RD register to 1.

It should be noted that, as both RVCO and SVCO have negative tuning slopes, a decrease in frequency will trigger the CAL_HI bit whilst an increase in frequency will trigger the CAL_LO bit.

To compensate for this drift the value REF_RVCO_FREQ_CAL code may be incremented, evaluation suggests typically by +/- 2 bits. This example is illustrated in Figure 1.

	REF_DEVICE_STATUS_RD Register ($83)	
	RVCO_CAL_HI (b1)	RVCO_CAL_LO (b0)
0	RVCO Vtune is below the high threshold	RVCO Vtune is above the low threshold
1	RVCO Vtune is above the high threshold	RVCO Vtune is below the low threshold

	DEVICE_STATUS_RD Register ($C4)	
	SVCO_CAL_HI (b3)	SVCO_CAL_LO (b2)
0	SVCO Vtune is below the high threshold	SVCO Vtune is above the low threshold
1	SVCO Vtune is above the high threshold	SVCO Vtune is below the low threshold

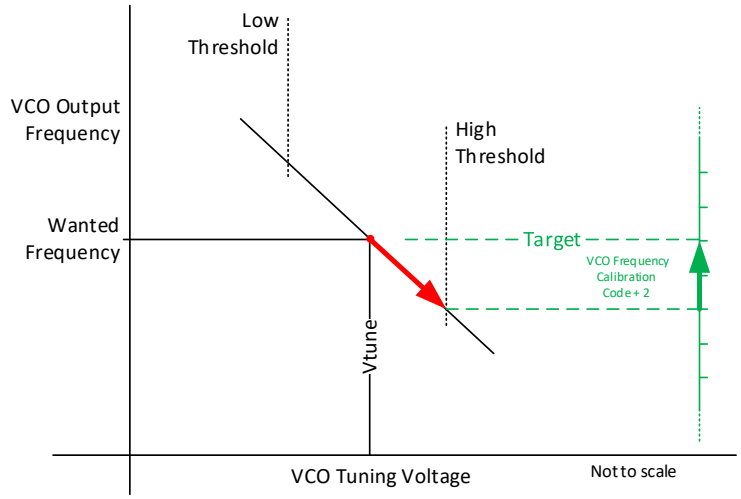


Figure 1 VCO is too slow

The method proposed below has been considered using the EV9400 Evaluation Kit. It is necessary to read and follow the description of the CMX940 and its Reference PLL and Synthesizer PLL in the CMX940 Datasheet to use this application note. Some knowledge of PLL design and optimisation would be an advantage.

4 REF_PLL/SPLL Fast Tuning Process

For the purposes of this application note an SVCO Low Frequency and a High Frequency together with a suitable SPLL output divider ratio was selected to be representative of a typical frequency hopping application. A REF_PLL frequency was also chosen with a suitable division ratio, to provide a low noise reference to the SPLL.

Both REF_PLL and SPLL are auto calibrated initially but are not during the hopping routine as only the SVCO_FREQ and SVCO_AMP parameters require updating between frequencies.
A script has been written based on a notional flow diagram that demonstrates hopping between the Low SVCO and High SVCO frequency extremes. Selected frequencies are as follows:

SPLL Ref Frequency (MHz) [REF_PLL / 10]	SVCO Frequency (MHz)	LOOUT Frequency (MHz) [SVCO / 8]
96	2720	340
96	4080	510

Table 1 Selected REF_PLL and SPLL frequencies for CMX940 fast tune hopping test

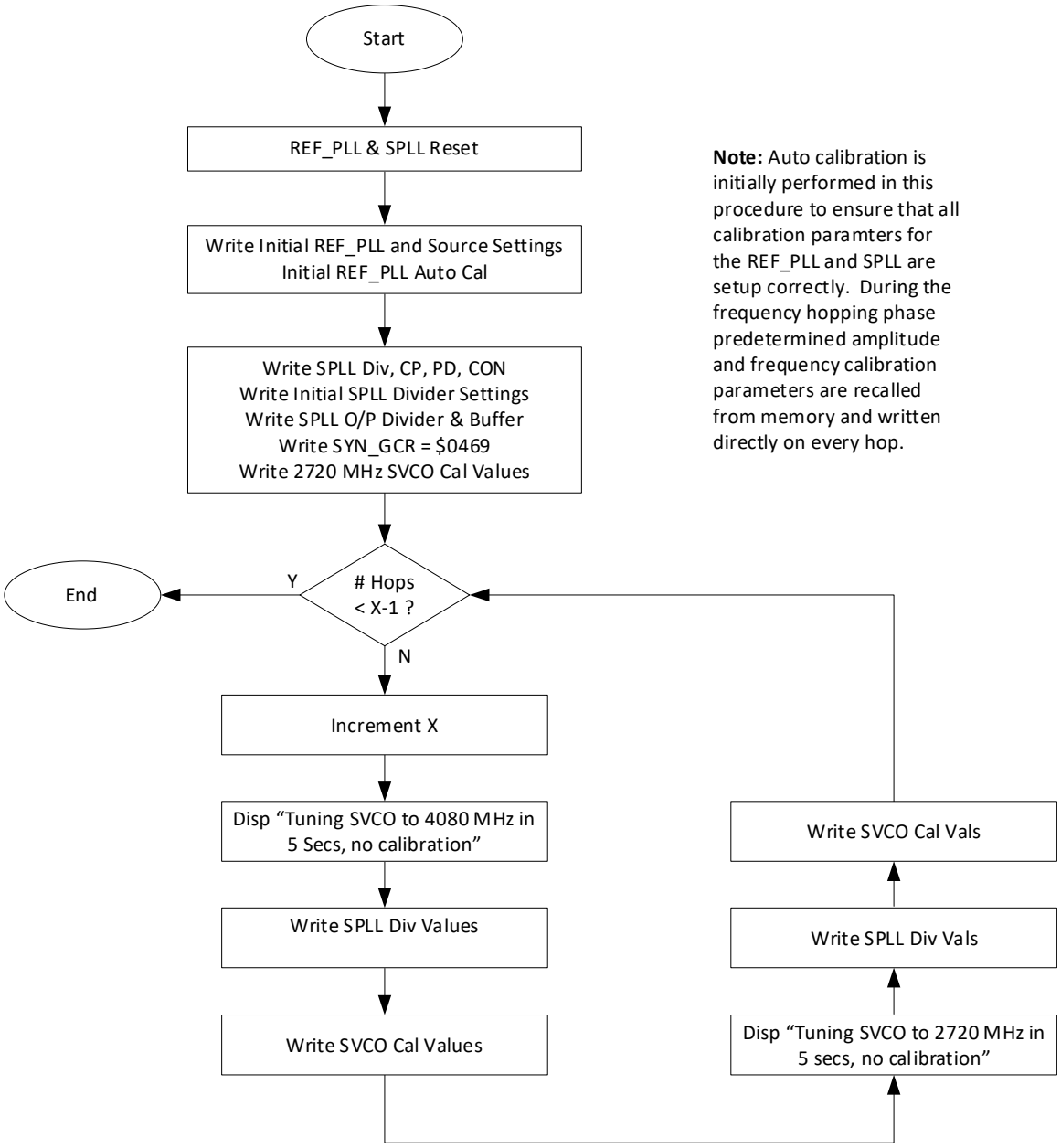


Figure 2 Fast tuning frequency example hopping flow diagram

5 PE0003 Script

The following PE0003 extract demonstrates fast SVCO switching between 2720 and 4080 MHz. This is performed 20 times using a while/end loop.

```
;*********************************************************************************************
; Script start
;*********************************************************************************************

start

device #1

disp "C-BUS 1"

; General Reset
copy 0, *REF_GEN_RST                                    ; REF_PLL General Reset
copy 0, *SYN_GEN_RST                                    ; SPLL General Reset

; Enable REF LO Buffer and REF LO Divider for REF_PLL checking via EV9400 REF_LOOUT (J8)
copy $00, *REF_LO_DIV                                   ; ODIV = 0
copy $0B, *REF_LO_BUF                                   ; PWR_CTRL = +3.2 dBm

; Master Clock Frequency Setup                          ; 38 MHz to nearest integer
copy $26, *REF_MCLK_FREQ

;..................................... REF_PLL Initial Setup

; Reference PLL Divider setup
copy $19, *REF_PLL_MDIV                                 ; Set Integer Divider to 25 for 960 MHz

; Source Divider setup
copy $0A, *SYN_SDIV                                     ; Set Source Divider to 10 for 96 MHz

; GCR for initial REF_PLL/RVCO calibration
copy $3F, *REF_GCR                                      ; GCR enable REF_PLL/RVCO (& MCLK output)
; delay 100                                             ; Optional delay

disp "  "
disp "REF_PLL Divider values loaded & REF_GCR Write = 0x3F"

; SYN_MCLK frequency setup                              ; Set to the output frequency of the Source Divider in MHz
copy $E0, *SYN_MCLK_FREQ                                ; 96 in Hex with leading bit set to 1


;..................................... SPLL Initial Setup

; SPLL initial setup @ 16 bit Resolution
copy $12, *SPLL_CON                                     ; 16 bit FracN
copy $0527, *SPLL_CP                                    ; CP bias to 1600uA, current to 2000uA
copy $8307, *SPLL_BLEED                                 ; Set bleed current 100uA
copy $00, *SPLL_PD                                      ; Set Minimum phase detector pulse width

; Divider values for 2720 MHz
copy $001C, *SPLL_IDIV                                  ; Integer = 28 dec
copy $5555, *SPLL_FDIV0                                 ; Fraction 16 bit, for 2720 Mhz = $5555

;Output Divider & Buffer Setup
copy $05, *OUTPUT_DIV                                   ; Output divider = 8
copy $0C, *LOOUT1_BUF                                   ; Tx output buffer set to max power

; GCR for SPLL with an initial calibration              ; Cal bit cleared when complete
copy $0469, *SYN_GCR                                    ; Enable all SYN functions minus VCO and LO1 OUTPUT.
                                                        ; SPLL_CAL bit 3 Cleared following automatic calibration cycle

disp "  "
```

```
disp "2720 MHz SVCO, SPLL divider, CP, buffer loaded & SYN_GCR Write = 0x0469"
disp "  "
disp "LOOUT1 (J5) = 340 MHz"

;.....Initial setup complete


; Hop Counter
while (loop<20)                                  ; Number (N-1) of hops between 4080 and 2720 MHz

                        add 1, loop


                        ; SPLL frequency = 4080 MHz setup for fast tune

                        disp "  "
                        disp "Tuning SVCO to 4080 MHz in 5 Secs, no calibration”
                        disp "  "
                        disp "Output Buffer ON during Transition"
                        delay 5000

                        ; SVCO divider values for 4080 MHz
                        copy $002B, *SPLL_IDIV          ; Integer = 43 dec
                        copy $8000, *SPLL_FDIV0         ; Fraction 16 bit, for 4080 = $8000

                        ; Write SPLL calibration values for fast tuning to 4080 MHz SVCO frequency
                        copy $81B2, *SVCO_FREQ_CAL      ; Set SVCO_FREQ_CAL for fast tuning
                        copy $55, *SVCO_AMP_CAL         ; Set SVCO_AMP_CAL for fast tuning

                        disp "  "
                        disp "4080 MHz SVCO, SPLL divider, CP, buffer loaded & SYN_GCR remains set to 0x0461"
                        disp "  "
                        disp "LOOUT1 (J5) = 510 MHz"

                        ; End of  fast tuning to 4080 MHz


                        ; SPLL frequency = 2720 MHz setup for fast tune

                        disp "  "
                        disp " Tuning SVCO to 2720 MHz in 5 secs, no calibration"
                        disp "  "
                        disp "Output Buffer ON during Transition "
                        delay 5000

                        ; SVCO divider values for 2720 MHz
                        copy $001C, *SPLL_IDIV          ; Integer = 28 dec
                        copy $5555, *SPLL_FDIV0         ; Fraction 16 bit, for 2720 = $5555

                        ; Write SPLL calibration values for fast tuning to 2720 MHz SVCO frequency
                        copy $009E, *SVCO_FREQ_CAL      ; Set SVCO_FREQ_CAL for fast tuning
                        copy $5D, *SVCO_AMP_CAL         ; Set SVCO_AMP_CAL for fast tuning


                        disp “  “
                        Disp “2720 MHz SVCO,  SPLL divider, CP, buffer loaded & SYN_GCR remains set to 0x0461"

                        disp "  "
                        disp "LOOUT1 (J5) = 340 MHz"

                        ; End of fast tuning to 2720 MHz

endwhile

disp "  "
disp "Script Run Complete"

stop
```

6 Results

Using the above script, it was possible to measure REF_PLL/SPLL lock time across a wide frequency range. A frequency vs time sweep was initiated at the end of the last C-BUS write (CSN Low to High) and stopped when LOOUT1 was stable to within 1kHz of the wanted frequency. It can be seen that the above method achieves lock times <50 µs.

The test frequencies were chosen to ensure that both the high band and low band SVCO were being used, i.e. the greatest possible frequency jump.

Note: The C-BUS transaction time to set up the PLL with the required divider settings and VCO Calibration values would be approximately 10 µs assuming a 10MHz C-BUS SCLK while the host processor is able to write concurrently.

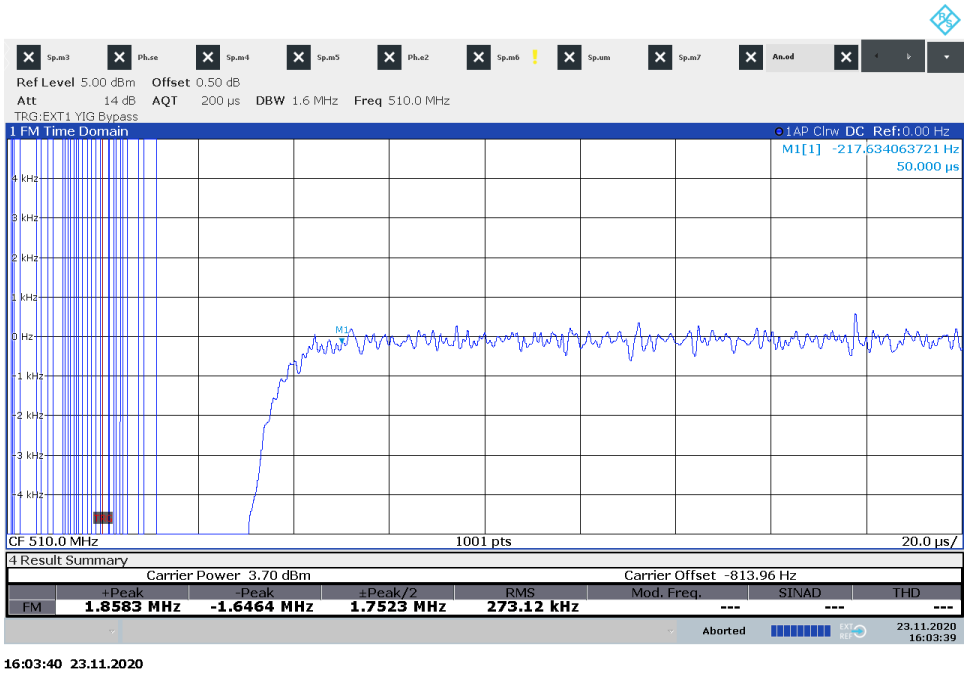


Figure 3 Time to lock for 340 to 510 MHz jump, < 50 µs

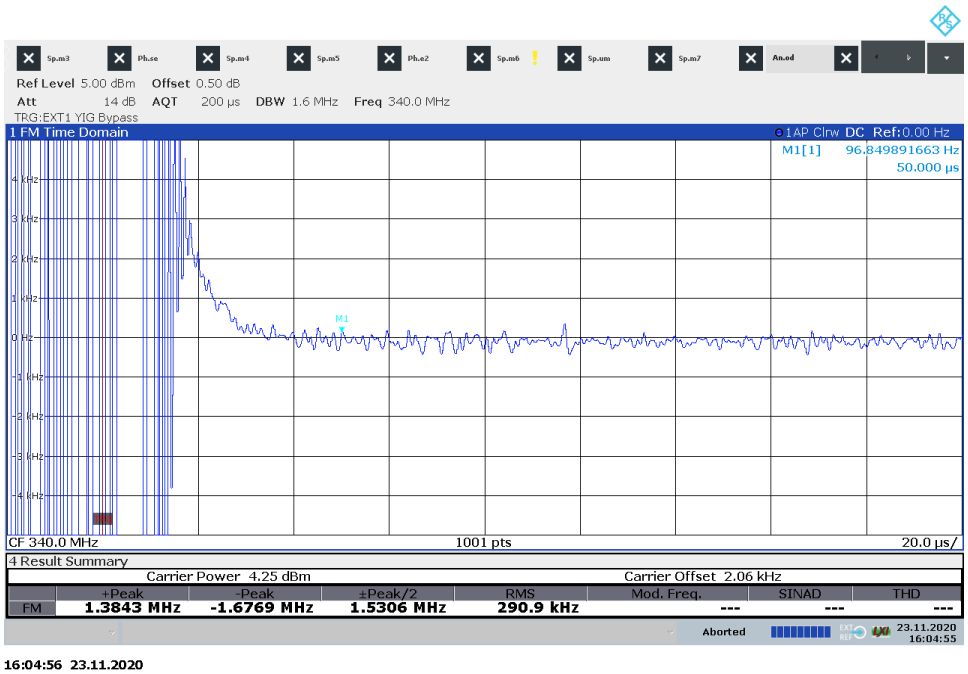


Figure 4 Lock time for 510 to 340 MHz jump, < 50 µs

By using the fast tuning method and only updating the SPLL to achieve a 10MHz hop from 456.250 to 466.250 MHz, a tuning time <30 µs was achievable. This is shown in figure 5 below.

Figure 5 10 MHz step tuning time <30 µs

The above measurements compare extremely well with typical results achieved using automatic calibration for both REF_PLL and SPLL. In earlier testing it was found that that REF_PLL tuning time to achieve lock at at 921.6MHz with /2 at the output divider took approximately 2.2 ms. The SPLL total tuning time took approximately 7.2 ms to achieve 500 MHz whilst both REF_PLL and SPLL took approximately 9.2 ms to achieve 500 MHz.

7 Conclusion

Whilst the automatic REF_PLL and SPLL calibration function within the CMX940 is very convenient and accurate it may not be possible to use it in applications that require very fast tuning times. This application note has demonstrated a method to reduce this time considerably by using calibration values determined by previous characterisation and applied directly to the Reference PLL and Synthesizer PLL in the application.

CML Microcircuits

United Kingdom p: +44 (0) 1621 875500 e: sales@cmlmicro.com techsupport@cmlmicro.com

Singapore p: +65 62888129 e: sg.sales@cmlmicro.com sg.techsupport@cmlmicro.com

United States p: +1 336 744 5050 800 638 5577 e: us.sales@cmlmicro.com us.techsupport@cmlmicro.com

www.cmlmicro.com