

SignalCore™

PRESERVING SIGNAL INTEGRITY



Hardware Manual

SC5407A & SC5408A

6 GHz RF Upconverter

www.signalcore.com

Table of Contents

1	General Information	4
1.1	Warranty	4
1.2	Copyright & Trademarks.....	4
1.3	International Materials Declarations.....	5
1.4	CE European Union EMC & Safety Compliance Declaration.....	5
1.5	Warnings Regarding Use of SignalCore Products	6
2	Physical Description	7
2.1	Unpacking.....	7
2.2	Setting Up the Device.....	7
2.3	Front Interface Indicators and Connectors.....	8
2.3.1	Signal Connections	9
2.3.2	Device LED Indicators	10
2.3.3	Communication and Supply Connection	10
2.3.4	Mini-USB Connection	11
2.3.5	Reset Button (Pin Hole)	12
3	Functional Description	13
3.1	Overview	13
3.2	The Signal Chain	14
3.2.1	The RF Input	14
3.2.2	The IF3 Input Path and IF3 Mixer	15
3.2.3	The IF2 Path and Second Mixer	15
3.2.4	External IF2 Input	16
3.2.5	The First Mixer and IF1 Path.....	16
3.2.6	The RF Path.....	16
3.2.7	The RF Output Path Switch.....	17
3.2.8	Signal Chain Configurations.....	17
3.3	The LO Module	18
3.3.1	The Reference Clocks	18
3.3.2	The First Local Oscillator (LO1)	19
3.3.3	The Second Local Oscillator (LO2)	19
3.3.4	The Third Local Oscillator (LO3).....	19
3.4	The Interface Module.....	20

4	Hardware Registers.....	21
4.1	Configuration Registers	21
4.1.1	Register 0x01 INITIALIZE	22
4.1.2	Register 0x02 SYSTEM_ACTIVE.....	23
4.1.3	Register 0x03 SYNTH_MODE (2 Bytes).....	23
4.1.4	Register 0x10 RF_FREQUENCY	23
4.1.5	Register 0x11 IF_FREQUENCY.....	24
4.1.6	Register 0x14 RF_FILTER.....	24
4.1.7	Register 0x15 ATTENUATOR.....	25
4.1.8	Register 0x16 SIGNAL_PATH.....	25
4.1.9	Register 0x17 CONFIG_AUTO_GAIN.....	26
4.1.10	Register 0x18 STORE_DEFAULT_STATE	27
4.1.11	Register 0x19 DEVICE_STANDBY	28
4.1.12	Register 0x1A REFERENCE_CLOCK.....	28
4.1.13	Register 0x1B REFERENCE_DAC.....	29
4.1.14	Register 0x1C LO1_PATH	29
4.1.15	Register 0x1D SYNTH_SELF_CAL.....	30
4.1.16	Register 0x1E USER_EEPROM_WRITE	30
4.1.17	Register 0x1F FREQ_PLAN_PARAM	30
4.2	Query Registers	31
4.2.1	Register 0x30 GET_DEVICE_PARAM.....	32
4.2.2	Register 0x31 GET_TEMPERATURE	32
4.2.3	Register 0x32 GET_DEVICE_STATUS.....	33
4.2.4	Register 0x33 GET_DEVICE_INFO	34
4.2.5	Register 0x35 CAL_EEPROM_READ	35
4.2.6	Register 0x36 USER_EEPROM_READ.....	35
4.2.7	Register 0x37 SERIAL_OUT_BUFFER.....	36
4.2.8	Device Parameters Data and Format.....	36
4.2.9	Device Information Parameters and Format	37
5	Communication Interfaces.....	1
5.1	Communication Data Format	1
5.2	USB Interface.....	1
5.2.1	Control Transfer	1

5.2.2	Bulk Transfer	1
5.3	SPI Interface	2
5.3.1	Writing the SPI Bus	3
5.3.2	Reading the SPI Bus	3
5.4	RS232 Interface	4
5.4.1	Writing to the Device Via RS232.....	4
5.4.2	Reading from the Device Via RS232	5
5.5	PXI Express	5
5.5.1	Setting Up the PCI to Serial Bridge	5
5.5.2	Writing to the Device.....	6
5.5.3	Reading from the Device	6
6	Calibration	7
6.1	Calibration EEPROM Map	7
6.2	Absolute Conversion Gain	8
6.3	Absolute Gain of the RF Conversion Path.....	8
6.4	Gain Through the Bypass RF Conversion Path.....	9
6.4.1	Applying Calibration	9
	Revision Table.....	11

1 General Information

1.1 Warranty

This product is warranted against defects in materials and workmanship for a period of three years from the date of shipment. SignalCore will, at its option, repair or replace equipment that proves to be defective during the warranty period. This warranty includes parts and labor.

Before any equipment will be accepted for warranty repair or replacement, a Return Material Authorization (RMA) number must be obtained from a SignalCore customer service representative and clearly marked on the outside of the return package. SignalCore will pay all shipping costs relating to warranty repair or replacement.

SignalCore strives to make the information in this document as accurate as possible. The document has been carefully reviewed for technical and typographic accuracy. If technical or typographical errors exist, SignalCore reserves the right to make changes to subsequent editions of this document without prior notice to possessors of this edition. Please contact SignalCore if errors are suspected. In no event shall SignalCore be liable for any damages arising out of or related to this document or the information contained in it.

EXCEPT AS SPECIFIED HEREIN, SIGNALCORE, INCORPORATED MAKES NO WARRANTIES, EXPRESS OR IMPLIED, AND SPECIFICALLY DISCLAIMS ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. CUSTOMER'S RIGHT TO RECOVER DAMAGES CAUSED BY FAULT OR NEGLIGENCE ON THE PART OF SIGNALCORE, INCORPORATED SHALL BE LIMITED TO THE AMOUNT THERETOFORE PAID BY THE CUSTOMER. SIGNALCORE, INCORPORATED WILL NOT BE LIABLE FOR DAMAGES RESULTING FROM LOSS OF DATA, PROFITS, USE OF PRODUCTS, OR INCIDENTAL OR CONSEQUENTIAL DAMAGES, EVEN IF ADVISED OF THE POSSIBILITY THEREOF. This limitation of the liability of SignalCore, Incorporated will apply regardless of the form of action, whether in contract or tort, including negligence. Any action against SignalCore, Incorporated must be brought within one year after the cause of action accrues. SignalCore, Incorporated shall not be liable for any delay in performance due to causes beyond its reasonable control. The warranty provided herein does not cover damages, defects, malfunctions, or service failures caused by owner's failure to follow SignalCore, Incorporated's installation, operation, or maintenance instructions; owner's modification of the product; owner's abuse, misuse, or negligent acts; and power failure or surges, fire, flood, accident, actions of third parties, or other events outside reasonable control.

1.2 Copyright & Trademarks

Under the copyright laws, this publication may not be reproduced or transmitted in any form, electronic or mechanical, including photocopying, recording, storing in an information retrieval system, or translating, in whole or in part, without the prior written consent of SignalCore, Incorporated.

SignalCore, Incorporated respects the intellectual property rights of others, and we ask those who use our products to do the same. Copyright and other intellectual property laws protect our

products. Use of SignalCore products is restricted to applications that do not infringe on the intellectual property rights of others.

“SignalCore”, “signalcore.com”, and the phrase “preserving signal integrity” are registered trademarks of SignalCore, Incorporated. Other product and company names mentioned herein are trademarks or trade names of their respective companies.

1.3 International Materials Declarations

SignalCore, Incorporated uses a fully RoHS compliant manufacturing process for our products. Therefore, SignalCore hereby declares that its products do not contain restricted materials as defined by European Union directive 2002/95/EC (EU RoHS) in any amounts higher than limits stated in the directive. This statement assumes reliable information and data provided by our component suppliers and may not have been independently verified through other means. For products sold into China, we also comply with the “Administrative Measure on the Control of Pollution Caused by Electronic Information Products” (China RoHS). In the current stage of this legislation, the content of six hazardous materials must be explicitly declared. Each of those materials, and the categorical amount present in our products, are shown below:

鉛 Lead (Pb)	汞 Mercury (Hg)	镉 Cadmium (Cd)	六价铬 Hexavalent Chromium (Cr(VI))	多溴联苯 Polybrominated biphenyls (PBB)	多溴二苯醚 Polybrominated diphenyl ethers (PBDE)
✓	✓	✓	✓	✓	✓

A ✓ indicates that the hazardous substance contained in all of the homogeneous materials for this product is below the limit requirement in SJ/T11363-2006. An X indicates that the particular hazardous substance contained in at least one of the homogeneous materials used for this product is above the limit requirement in SJ/T11363-2006.

1.4 CE European Union EMC & Safety Compliance Declaration

The European Conformity (CE) marking is affixed to products with input of 50 - 1,000 Vac or 75 - 1,500 Vdc and/or for products which may cause or be affected by electromagnetic disturbance. The CE marking symbolizes conformity of the product with the applicable requirements. CE compliance is a manufacturer’s self-declaration allowing products to circulate freely within the European Union (EU). SignalCore products meet the essential requirements of Directives 2004/108/EC (EMC) and 2006/95/EC (product safety) and comply with the relevant standards. Standards for Measurement, Control and Laboratory Equipment include EN 61326 and EN 55011 for EMC, and EN 61010-1 for product safety.

1.5 Warnings Regarding Use of SignalCore Products

- (1) PRODUCTS FOR SALE BY SIGNALCORE, INCORPORATED ARE NOT DESIGNED WITH COMPONENTS NOR TESTED FOR A LEVEL OF RELIABILITY SUITABLE FOR USE IN OR IN CONNECTION WITH SURGICAL IMPLANTS OR AS CRITICAL COMPONENTS IN ANY LIFE SUPPORT SYSTEMS WHOSE FAILURE TO PERFORM CAN REASONABLY BE EXPECTED TO CAUSE SIGNIFICANT INJURY TO A HUMAN.
- (2) IN ANY APPLICATION, INCLUDING THE ABOVE, RELIABILITY OF OPERATION OF THE SOFTWARE PRODUCTS CAN BE IMPAIRED BY ADVERSE FACTORS, INCLUDING BUT NOT LIMITED TO FLUCTUATIONS IN ELECTRICAL POWER SUPPLY, COMPUTER HARDWARE MALFUNCTIONS, COMPUTER OPERATING SYSTEM SOFTWARE FITNESS, FITNESS OF COMPILERS AND DEVELOPMENT SOFTWARE USED TO DEVELOP AN APPLICATION, INSTALLATION ERRORS, SOFTWARE AND HARDWARE COMPATIBILITY PROBLEMS, MALFUNCTIONS OR FAILURES OF ELECTRONIC MONITORING OR CONTROL DEVICES, TRANSIENT FAILURES OF ELECTRONIC SYSTEMS (HARDWARE AND/OR SOFTWARE), UNANTICIPATED USES OR MISUSES, OR ERRORS ON THE PART OF THE USER OR APPLICATIONS DESIGNER (ADVERSE FACTORS SUCH AS THESE ARE HEREAFTER COLLECTIVELY TERMED "SYSTEM FAILURES"). ANY APPLICATION WHERE A SYSTEM FAILURE WOULD CREATE A RISK OF HARM TO PROPERTY OR PERSONS (INCLUDING THE RISK OF BODILY INJURY AND DEATH) SHOULD NOT BE SOLELY RELIANT UPON ANY ONE COMPONENT DUE TO THE RISK OF SYSTEM FAILURE. TO AVOID DAMAGE, INJURY, OR DEATH, THE USER OR APPLICATION DESIGNER MUST TAKE REASONABLY PRUDENT STEPS TO PROTECT AGAINST SYSTEM FAILURES, INCLUDING BUT NOT LIMITED TO BACK-UP OR SHUT DOWN MECHANISMS. BECAUSE EACH END-USER SYSTEM IS CUSTOMIZED AND DIFFERS FROM SIGNALCORE'S TESTING PLATFORMS, AND BECAUSE A USER OR APPLICATION DESIGNER MAY USE SIGNALCORE PRODUCTS IN COMBINATION WITH OTHER PRODUCTS IN A MANNER NOT EVALUATED OR CONTEMPLATED BY SIGNALCORE, THE USER OR APPLICATION DESIGNER IS ULTIMATELY RESPONSIBLE FOR VERIFYING AND VALIDATING THE SUITABILITY OF SIGNALCORE PRODUCTS WHENEVER SIGNALCORE PRODUCTS ARE INCORPORATED IN A SYSTEM OR APPLICATION, INCLUDING, WITHOUT LIMITATION, THE APPROPRIATE DESIGN, PROCESS AND SAFETY LEVEL OF SUCH SYSTEM OR APPLICATION.

2 Physical Description

2.1 Unpacking

All SignalCore products ship in antistatic packaging (bags) to prevent damage from electrostatic discharge (ESD). Under certain conditions, an ESD event can instantly and permanently damage several of the components found in SignalCore products. Therefore, to avoid damage when handling any SignalCore hardware, you must take the following precautions:



1. Ground yourself using a grounding strap or by touching a grounded metal object.
2. Touch the antistatic bag to a grounded metal object before removing the hardware from its packaging.
3. **NEVER touch exposed signal pins.** Due to the inherent performance degradation caused by ESD protection circuits in the RF path, the device has minimal ESD protection against direct injection of ESD into the RF signal pins.
4. When not in use, store all SignalCore products in their original antistatic bags.

Remove the product from its packaging and inspect it for loose components or any signs of damage. Notify SignalCore immediately if the product appears damaged in any way.

2.2 Setting Up the Device

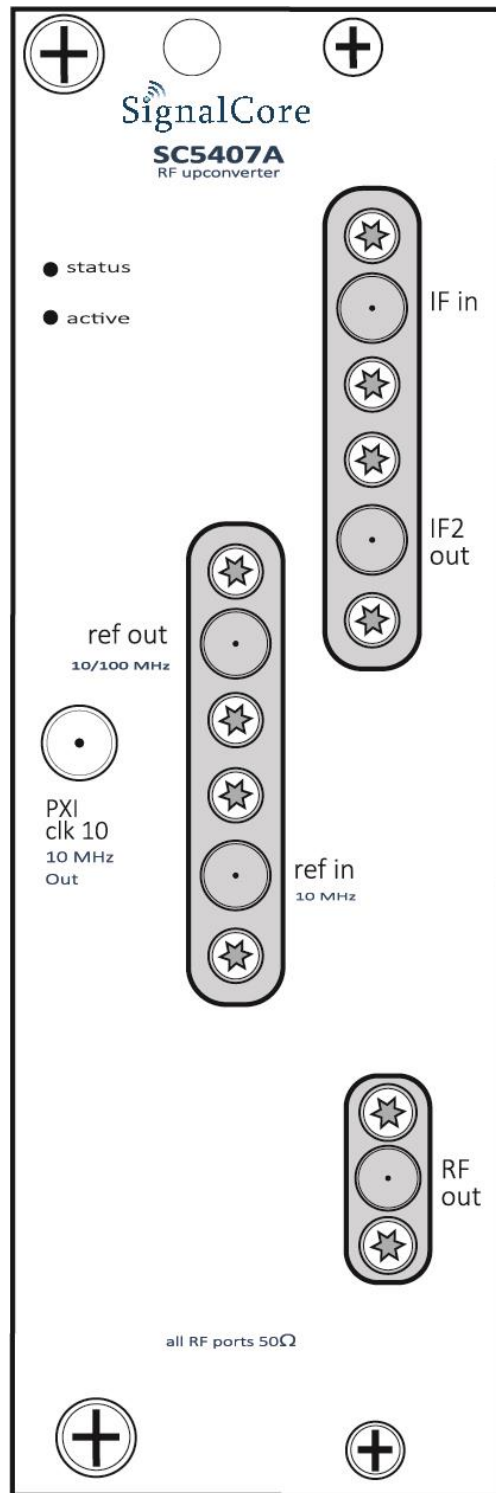
Integration of the SC5407A and SC5408A modules requires attention to maintain effective cooling. Inadequate cooling can cause the temperature inside the RF housing to rise above the maximum for this product, leading to improper performance, reduction of product lifespan, or complete product failure. SignalCore suggests providing moderate airflow across the RF housing. If active cooling is not an option, use thermal interface materials to bond the RF housing to a larger heatsinking surface (i.e. a system enclosure). As each device's integrated system configuration is unique, detailed cooling options cannot be provided.



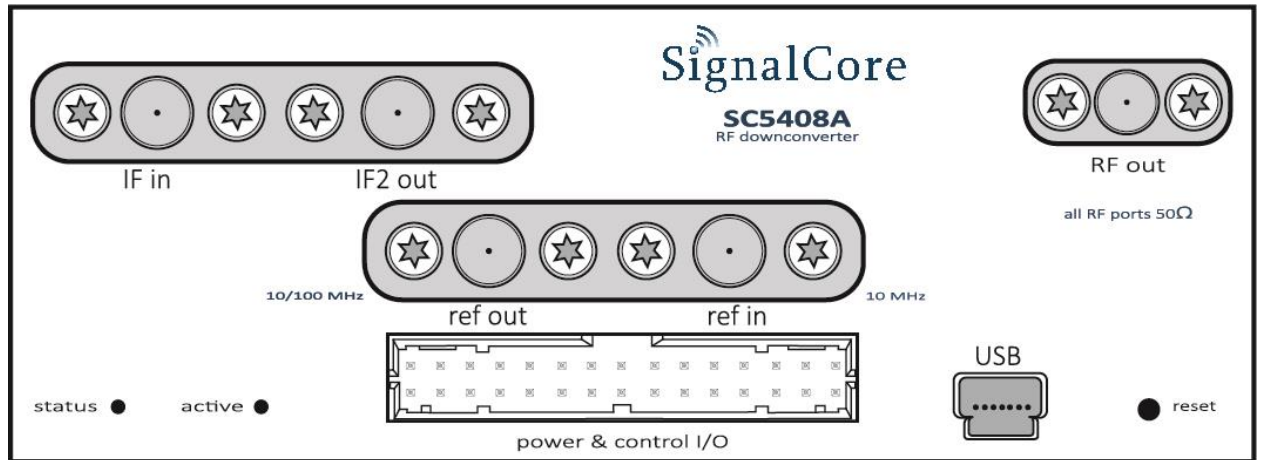
A cooling plan is sufficient when the SC5407A and SC5408A on-board temperature sensors indicate a rise of no more than 20°C above ambient temperature under normal operating conditions.

2.3 Front Interface Indicators and Connectors

The SC5407A is a PXIe-based RF upconverter with all user I/O located on the front face of the module.



The SC5408A is a serial controlled core module with all user connections on the front face of the module.



2.3.1 Signal Connections

All signal connections (ports) on the device are female SMA-type. Exercise caution when fastening cables to the signal connections. Over-tightening any connection can cause permanent damage to the device.



The condition of your system signal connections can significantly affect measurement accuracy and repeatability. Connections that are improperly mated, dirty, damaged, or worn can degrade measurement performance. Clean out any loose, dry debris from connectors with clean, low-pressure air (available in spray cans from office supply stores).



If deeper cleaning is necessary, use lint-free swabs and isopropyl alcohol to gently clean inside the connector barrel and the external threads. Do not mate connectors until the alcohol has completely evaporated. Excess liquid alcohol trapped inside the connector may degrade measurement performance until fully evaporated (this may take several days).



Tighten all SMA connections with 3 in-lb min to 5 in-lb max (56 N-cm max)

- | | |
|----------------|---|
| RF Out | This is the RF output port to the device with nominal impedance of 50 Ω . Its maximum output power is +18dBm. |
| IF2 Out | This is the output to the second IF stage of the converter. Its nominal output frequency is 1250 MHz. |
| IF In | This is the IF input of the device with nominal impedance of 50 Ω . Its maximum input power is +13 dBm. |
| Ref In | This is the 10 MHz reference input to the device, enabling the device to phase lock its internal clocks to an external reference source. This nominally 50 Ω port is AC coupled. Maximum rated input power is +10 dBm. |

Ref Out This is the output reference port of the device, allowing for the export of its internal reference clocks. The reference frequency is selectable between 10 MHz or 100 MHz. Typical power level is 3 dBm.

2.3.2 Device LED Indicators

These are LED indicator lights for the device, and their functions are listed in *Table 1* and *Table 2*.

Table 1. Status LED Indicator

LED Color	Description
Green	The device is functioning properly in the state that it is programmed for.
Amber	Indicates that all functions are on standby mode.
Red	Indicates that one or more local oscillators are not functioning correctly.
Off	No supply or supply error

Table 2. Active Indicator

LED Color	Description
Green	An external interface port has accessed the device.
Red	Input supply voltage exceeded
Off	No current interface access

2.3.3 Communication and Supply Connection

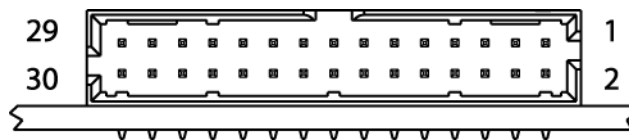


Figure 1. Power and Digital IO Connector

Power and communication to the SC5408A is provided through Molex **Milli-Grid[™]** 2.00mm pitch, 30 position, male header connector. Its part number is 87833-3020. A suggested receptacle female connector is the crimp terminal type 51110-3051 or ribbon type 87568-3093 from Molex. The pin definitions are listed in *Table 3*.



Pinouts are different for different SignalCore products with the same connector type. Please ensure that mating connectors and cables are wired correctly before connection.

Table 3. Interface connector pin out description

PIN #	Description
1,3,5,7	12V Supply Rail

PIN #	Description
2,4,6,8,10,12,16,20,24,28	GND
9,11,13,14,15	Reserved, Do Not Connect
17,18	Reserved, pull high to 3.3V or DNC
19	$\overline{\text{Reset}}$, System reset, logic 0 to reset device
21	Device Active, accessed
22	PLL status
23	RS232 Baud Rate / SPI Mode Select
25	$\overline{\text{CS}}$ Device select for SPI
26	SPI Clock
27	TX/MOSI. TX (DTE) for RS232 or MOSI for SPI
29	RX/MOSE. RX (DTE) for RS232 or MISO for SPI
30	SRDY. This pin is serial ready for SPI

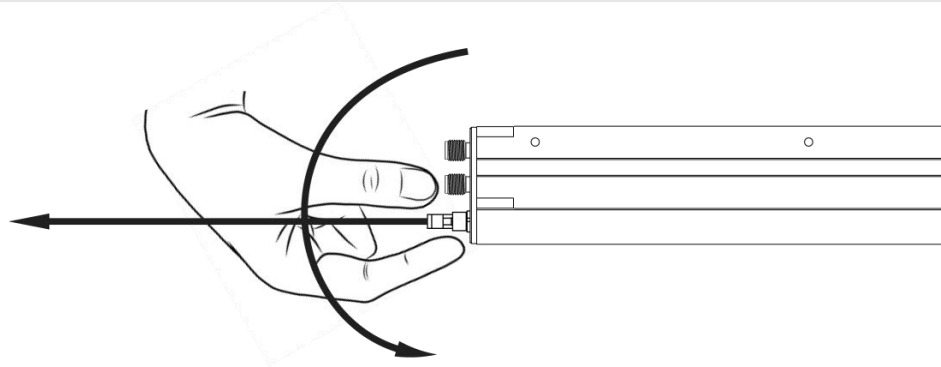
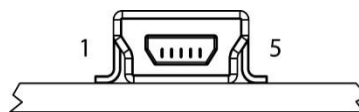


Figure 2. Proper Removal of Latching-Style Ribbon Cable Connectors

2.3.4 Mini-USB Connection



This is a mini-USB Type B connector for USB communication with the device using the standard USB 2.0 protocol (full speed) found on most host computers. The pinout of this connector, viewed from the front, is shown in the following table.

PIN #	USB Function	Description
1	VBUS	Vcc (+5 Volts)
2	D-	Serial Data (neg)
3	D+	Serial Data (pos)
4	ID	Not Used
5	GND	Device Ground (also tied to connector shell)

2.3.5 Reset Button (Pin Hole)

Behind this pin hole is the reset button, which is only available on the SC5408A. Using a pin to lightly depress this momentary-action push button switch will cause a hard reset to the device, putting it back to its default settings. All user settings will be lost. System reset capability can also be accessed through the communication header connector.

3 Functional Description

3.1 Overview

The SC5408A uses USB as its primary interface with an optional SPI or RS232 interface. The SC5407A is a PXIe version of the product.

The upconverter assembly consists of three module parts:

- The Signal Conversion Module contains the mixers, filters, signal amplifiers, and attenuators. This module is referred to as the “signal chain”.
- The Synthesized Local Oscillator Module that contains the 3 synthesized LO signals and the reference signal circuitry.
- The Power Conditioning and Digital Control Board contains the supply switchers that generate the needs rails for the RF modules and an onboard MCU that provides both the computation engine and interface between the user and the RF modules.

The figure below shows how the modules relate to each other.

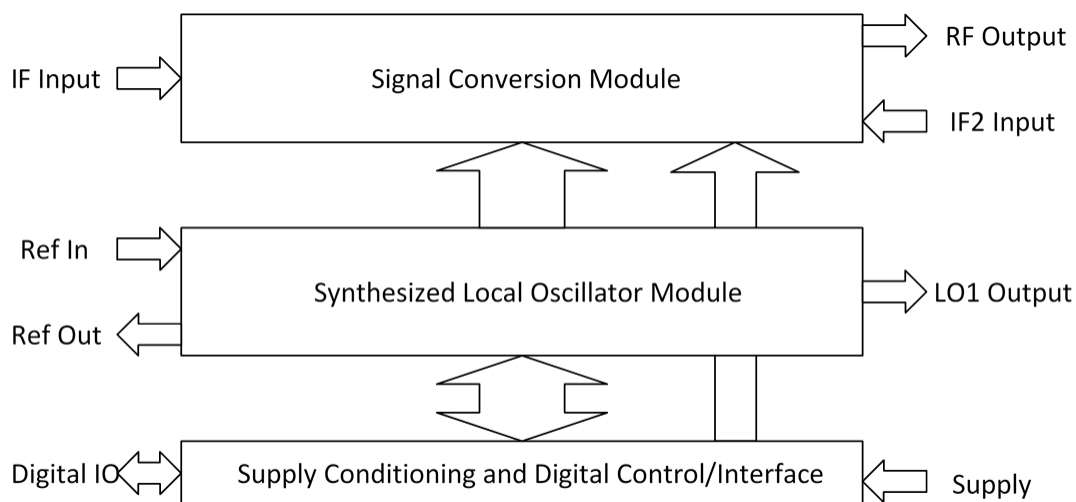


Figure 3. Simplified Block Representation of the Upconverter Assembly Module

The upconverter has a maximum of three conversion stages. Looking into the RF port of the device, the output is the RF stage whose signal is converted from an immediate frequency (IF) stage called IF1. IF1 is converted from the previous IF2 stage, which is the result of conversion from the input IF3 stage. The converted signals in each stage are derived by mixing their previous stage with their local oscillator (LO) signal. This mixing process results in many products and the desired product at each IF stage is picked out using band-pass filters. Signals that appear inside the passband of the filter will be passed through, while those that are outside the passband are suppressed. Generally, the farther an unwanted signal is away from the filters the better it is suppressed. Unwanted byproduct signals that are close to the passband or inside the passband are called spurious signals (or ‘spurs’ for short).

Because of the wide bandwidth of the IF filters there are output RF frequency regions as well as IF input regions where the number and level of spurs are higher than typical, making those regions unsuitable for high dynamic range applications.

The output tunable range of the upconverter is from ≈ 100 kHz to 6 GHz, while the output tunable IF3 range is from ≈ 10 MHz to 500 MHz. However, due to broadband amplifier limitations, the output usable range is 20 MHz to 6 GHz. The input IF also has a usable lower boundary, limited by the bandwidth of the IF filters. The lower range is typically 90 MHz for the 80 MHz filter. The output RF is tuned using the first local oscillator (LO1), and input IF3 is tuned with LO3. IF3 is only tunable when the IF bandwidth is set to 80 MHz or 160 MHz; it is set to a fixed frequency of 1.25 GHz when the converter IF bandwidth is set to 320 MHz.

The frequency accuracy and stability of the device is derived from an onboard temperature-controlled crystal oscillator (TCXO) with accuracy better than 500 ppb. For better accuracy and stability, the upconverter can lock to an external reference of higher precision.

3.2 The Signal Chain

The conversion module contains the mixers, filters, amplifiers, and attenuators used to convert, purify, and maintain the amplitude of the signal, all the way from the input of the device to its output. This cascade of signal conversion and conditioning stages is often referred to as the signal chain, which is shown on the system block drawing in *Figure 4*.

3.2.1 The RF Input

All RF connectors are high quality female stainless steel SMA types rated to 18 GHz of operation. The RF port is AC coupled with a $0.1 \mu\text{F}$ broadband capacitor as indicated in *Figure 4*. The output frequency range at this port is less than 100.0 kHz (only through the bypass conversion path, 20 MHz is the lower usable limit through the conversion path) to greater than 6.0 GHz. The upper limit to which the device tunes to above 6 GHz varies slightly from unit to unit depending on the limits of the oscillator circuitry. The usability of the upper out-of-bounds region depends on the roll-off response of the RF elliptic filter and the IF1 mixer.

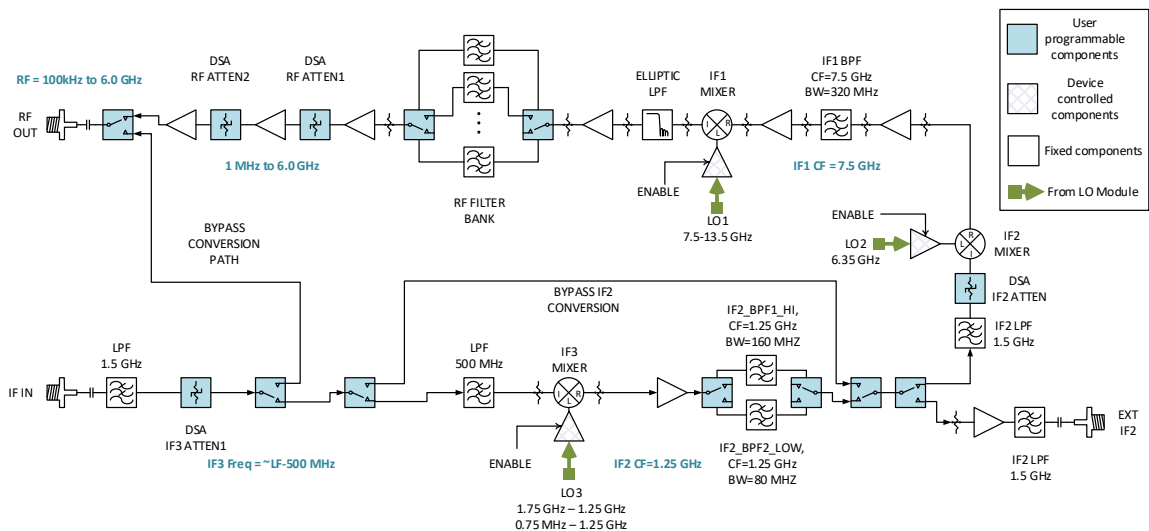


Figure 4. Block diagram of the signal conversion module of the upconverter assembly

3.2.2 The IF3 Input Path and IF3 Mixer

The third IF (IF3) is tunable from ± 5 MHz to ± 500 MHz, as LO3 is tuned from 750 MHz to 1750 MHz in 5 MHz steps. The frequency value of LO3 is based on the following relationship:

$$LO3 = IF2 \mp IF3$$

The negative value of IF3 indicates that the RF spectrum is inverted with respect to the IF input. When spectral inversion is selected, LO3 tunes below IF2, otherwise it tunes above. LO3 is disabled when conversion to IF2 is bypassed. In this case, the input port is directly connected to IF2, resulting in a fixed input IF of 1.25 GHz. Notice that, when the input IF approaches lower frequencies, the frequency of LO3 approaches IF2. The IF3 mixer not being ideal will allow LO3 leakages into IF2, and if these leakages are within the passband of the IF2 filters, they will appear as large spurious signals at the RF output. For this reason, the IF3 frequency should be set to no less than 90 MHz when the 80 MHz bandpass filter is selected.

The attenuator IF3_ATTEN1 is used to adjust the input IF signal. It sets the amplitude of the direct RF path, the IF2 conversion path, and the bypassed IF2 conversion path. For a best balance of linearity and signal-to-noise ratio, the recommended power at this mixer is about -5 dBm.

The IF3 input signal has 2 paths to take after IF3_ATTEN1:

1. Bypass the second conversion stage and it will either be directly mixed up to IF1 via IF2 mixer, or directly piped out to the IF2 output port.
2. Pass on to IF3 mixer and be mixed up to IF2.

3.2.3 The IF2 Path and Second Mixer

The converted IF2 signal at the IF3 mixer passes through the selectable bandpass filters BPF1 and BPF2, whose frequency bandwidths are 160 MHz and 80 MHz respectively. Select the appropriate filter for the desired bandwidth. When the signal bandwidth is not larger than 80 MHz, selecting the 80 MHz filter is always a better option for 2 reasons:

1. It provides better suppression of LO leakages, especially LO3 leakages when input IF frequencies are low.
2. It provides better backward suppression of LO1, which could mix with LO3 to produce in-band spurs.

After the filters, there are 2 paths IF2 can take:

1. Convert to IF1 via the second IF2 mixer.
2. Pass directly out to the IF2 output port.

The second IF (IF2) is typically centered at 1.25 GHz and has narrow settable range with 5 MHz steps. This IF2 is converted to IF1 by mixing with LO2 at the IF2 mixer. The relationship between LO2, IF1, and IF2 is:

$$LO2 = IF1 - IF2$$

IF1 and IF2 are settable with the condition that LO2 must be greater than or equal to 6.25 GHz and less than or equal to 6.45 GHz.

Finally, the IF2_ATTEN is used to adjust the signal level to the mixer, providing the correct levels to achieve the desired linearity.

3.2.4 External IF2 Input

The upconverter provides a direct port output of IF3 or IF2. The IF at this port may be further upconverted to a higher frequency beyond the range of the device. An example of a use case is discussed in *Section 3.3.2*.

3.2.5 The First Mixer and IF1 Path

IF1 is the product of LO2 and IF2, which is passed through a 320 MHz bandpass filter prior to being converted to RF. The center frequency is typically at 7.6 GHz, however, it may be adjusted programmatically from 7.5 GHz to 7.7 GHz, stepping at 5 MHz. The default center frequency is set at the factory at the time of calibration to compensate for shifts in the IF1 filter response due to fabrication variations. Another advantage of setting the IF1 frequency higher is the ability to place it closer to the filter slope region, providing better LO2 leakage suppression.

3.2.6 The RF Path

The RF signal is the product of the tunable LO1 signal and the IF1 signal using the IF1 mixer:

$$RF = LO1 - IF1$$

Because the RF is lower in frequency than both the LO1 and IF1, the elliptic lowpass filter that immediately follows the mixer suppresses the later unwanted signals resulting in a clean output signal. An appropriate filter is selected from the RF filter bank to suppress higher order mixed products between LO1, RF, and IF1 that appear out of band in the output spectrum. The RF amplifiers provide the necessary gain to achieve the required maximum output level, while the RF step attenuators provide amplitude adjustment.

3.2.6.1 RF Filter Selection

When the device is programmed to auto-select the RF filter from the filter bank based on the tuned RF frequency, *Table 4* shows the selection basis:

Table 4. RF Filter Selection vs Frequency

Filter #	Frequency Range
0	0 to < 750 MHz
1	750 to < 1400 MHz
2	1400 to < 1950 MHz
3	1950 to < 2500 MHz
4	2500 to < 3000 MHz
5	3000 to < 3600 MHz
6	≥ 3600 MHz
7	Through path (manually selectable)

3.2.7 The RF Output Path Switch

The RF output can come from 2 selected paths:

1. The default is the conversion path where the RF signal is converted from the IF port.
2. The alternative path is directly from the IF port, bypassing all means of frequency conversion. For lower RF frequencies that can be generated directly from a baseband source such as an arbitrary waveform generator, this path provides the convenience to do so. Another reason for choosing this path for lower frequencies is that conversion to a low RF from an IF may result in more unwanted spurs at the RF output.

When the conversion path is bypassed, all mixer LO drive amplifiers are disabled.

3.2.8 Signal Chain Configurations

Recommended signal chain configurations are provided below for various application scenarios. These are strictly recommendations and not indicative of the upconverter function limitations; users will need to adapt the device to their own applications.

3.2.8.1 *Low Frequency Output*

Due to gain roll-off of the RF amplifiers at low RF frequencies, it is recommended to generate the signals at baseband and use the bypass conversion path.

3.2.8.2 *Dynamic Range Setting*

There are 2 sets of digital step attenuators to control the conversion gain of the upconverter. The first set consists of RF_ATTEN1 and RF_ATTEN2 and the second set consists of IF3_ATTEN1 and IF2_ATTEN. All attenuators have 30 dB range and 1 dB step.

To set the upconverter for better SNR, the gain should be shifted to the IF input path of the device, before the IF3 mixer. The user should set the IF mixer input levels to be between 0 and 3 dBm and set the IF2_ATTEN to 0. The RF attenuators are then used to adjust the final RF output level. However, this means the signal level starts off higher as it enters the first mixer and the subsequent components. As a result, the apparent linearity of the device is lower.

To set the upconverter for better linearity, the gain should be shifted to the output RF path (after the IF1 mixer) and reduced in the IF paths. The signal power level at the IF3 mixer should be approximately -10 dBm and the IF2_ATTEN should be set to 8 – 10 dB for improved linearity. Since the input signal is low, the relative SNR will be lower. But, as the first mixer and subsequent components experience lower power levels, the apparent linearity of the device is higher.

When the device gain is balanced well, the device can achieve SNR better than 130 dBc/Hz while maintaining an output OIP3 better than 30 dBm. When the device is optimized for best SNR, typical values better than -147 dBc/Hz can be achieved.

3.3 The LO Module

The local oscillator (LO) module contains three local oscillators, the reference clocks, and the calibration EEPROM as shown in *Figure 5*.

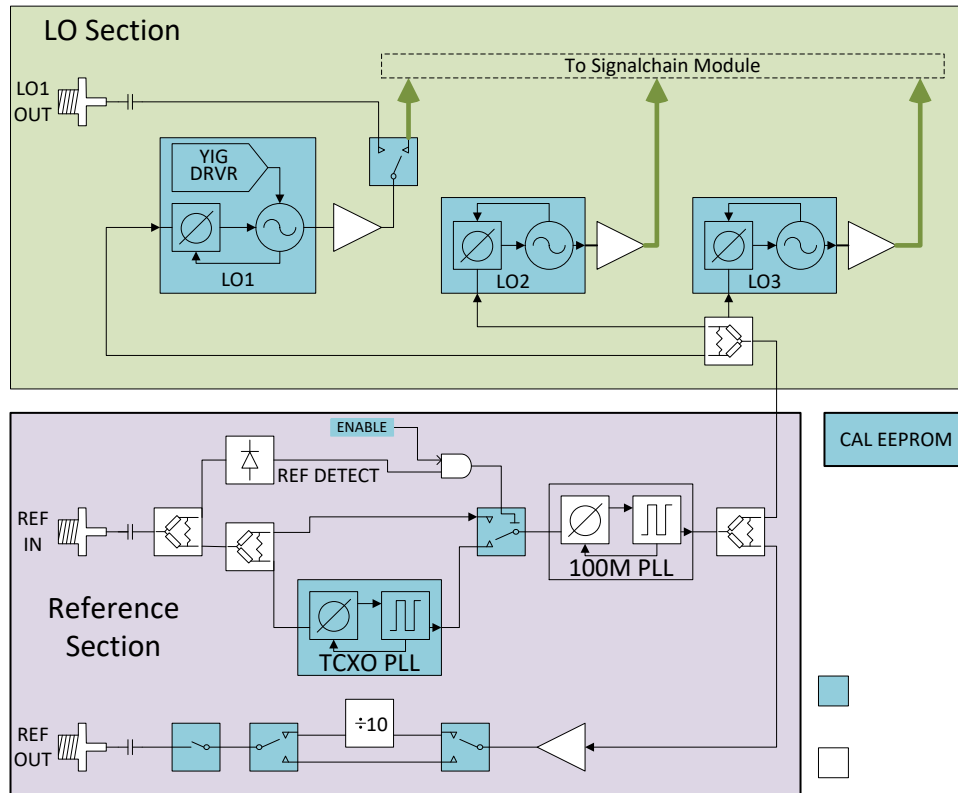


Figure 5. Block diagram of the local oscillator module

3.3.1 The Reference Clocks

The base clock of the upconverter is a 10 MHz Voltage Controlled Temperature Controlled Crystal Oscillator (VCTCXO) with initial accuracy better than 500 ppb once the device has reached a stable temperature. Its initial accuracy is set at the factory via an on-board 14-bit voltage reference DAC. This DAC is accessible for dynamic accuracy calibration. The other reference is a 100 MHz voltage-controlled crystal oscillator (VCXO), which is phase locked to the based reference whenever an external reference source is not used.

When an external reference is selected as the base clock by enabling the device to phase lock to it, the effect only occurs when the presence of this reference is detected. In other words, although the device is programmed to lock externally, it will not attempt to do so until the reference signal is detected at the input reference port. Notice, both the reference clocks will attempt to lock to the external source. Having the VCXO lock directly to the external source has the advantage of utilizing the close-in phase noise of the source; we assume that the external source is superior to

the internal base. Although the internal VCTCXO is not used it is important to have it remain powered on to maintain its temperature stability. Like the VCXO, it is also phase locked to the external source.

The reference output is derived from the buffered VCXO. When its output is enabled, its frequency can be selected for 10 MHz or 100 MHz. Typically output power is +3 dBm as its accuracy and stability is that of either the internal VCTCXO or external reference source.

3.3.2 The First Local Oscillator (LO1)

This first local oscillator (LO1) is a YIG-based synthesizer tunable between at least 7 GHz and more than 14 GHz typically. Its multiple hybrid PLL approach gives it the advantages of having lower phase noise, very low spurs, and a small frequency step of 1 Hz. This signal drives the first mixer of the signal chain. This signal can also be programmed to route to an external connector to drive an external mixer for frequency conversion outside the limits of the device. For example, its frequency could be doubled externally to drive a Ka band mixer to convert the IF2 signal to a higher frequency. A possible block representation of the example is shown in *Figure 6*.

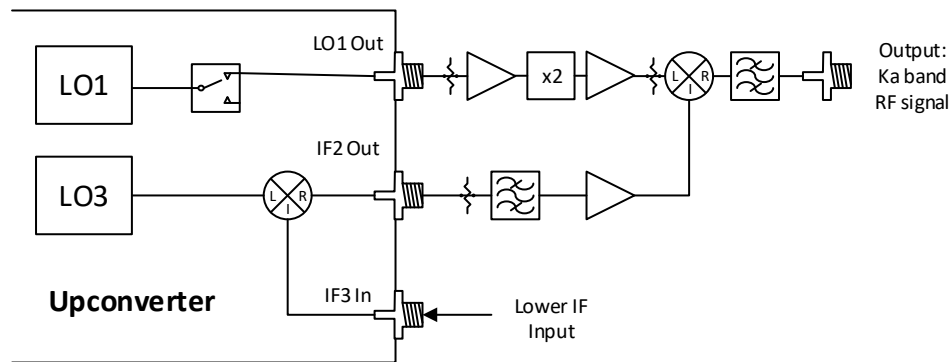


Figure 6. Application example of using external LO1 for up conversion to a Ka band signal.

The frequency of LO1 is set indirectly by programming the IF1 and RF signal frequencies when it is switched to drive the internal mixer for conversion. When it is programmed to drive externally, its frequency can be set directly by writing the appropriate register.

3.3.3 The Second Local Oscillator (LO2)

The second LO (LO2) tunes between 6.25 GHz and 6.45 GHz in 5 MHz frequency steps. It drives the second mixer of the signal chain. Its frequency is indirectly set using IF1 and IF2 values using the following relationship:

$$LO2 = IF1 - IF2$$

The ability to tune LO2 allows for the optimization of IF combinations to move higher order intermodulation products that may form inside the passband to outside the passband.

3.3.4 The Third Local Oscillator (LO3)

The third LO (LO3) frequency is set indirectly with IF2 and IF3 frequency values, and the polarity of the IF3 spectrum is set with respect to the RF. IF3 can be set ideally from DC to 500 MHz. When the RF Spectrum is set to be non-inverted, LO3 tunes from 1.25 GHz up to 1.75 GHz, otherwise it

tunes from 1.25 GHz down to 750 MHz. Caution needs to be taken when tuning to the limits because of larger LO leakages and intermodulation products that could form in-band. For example, when the LO3 is set to close to 1.25 GHz, the IF2 bandpass filters are not sufficiently aggressive to provide the suppression that may be needed to keep the leakage to an acceptable level at the RF output port.

3.4 The Interface Module

Figure 7 shows the power and control interface block for both the SC5407A and SC5408A. All supply rails are produced on board; their voltages are regulated and actively filtered to keep noise to a minimum. Thus, these upconverters are tolerant to “dirty” external power supplies.

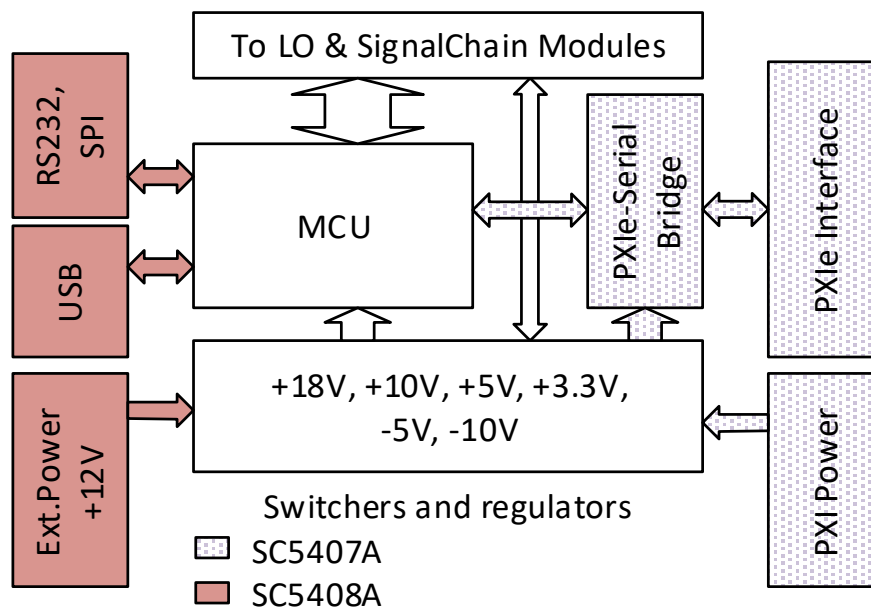


Figure 7. Block diagram of the power and interface module.

4 Hardware Registers

The set of hardware registers of the upconverter may be divided into a configuration set and a query set; the configuration registers are write-only registers to set up the states of the device, while the query registers request the device to prepare and send back data associated with them. The registers are identical for all communication interfaces: PXIe, USB, RS232, and SPI. Data communication to all interfaces are sent in 1 byte wide, that is, data is transferred byte-by-byte. Register data lengths vary in size, and it is important that the correct number of bytes are sent for the corresponding register. Failure to do so may cause communication to lock-up and the device to become unresponsive.

4.1 Configuration Registers

These are write-only registers to configure the device. The registers vary in length to reduce redundant data and improve the communication speed, especially for SPI and RS232 interfaces. Furthermore, it is vitally important that the length of data written to a register is exact because failure to do so will cause the interfaces to misinterpret the incoming data, leaving the device in a stalled state. The total number of bytes is the sum of the register address (1 Byte) and its corresponding data bytes. For example, to set the RF frequency value, eight bytes must be written; the sum of the 1 register byte and 7 data bytes. See the RF_FREQUENCY register of Table 5. The table provides a summary of the configuration registers, and each register is explained in detail.

Table 5. Configuration Registers

Register Name	Reg Add	Serial Range	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
INITIALIZE	0x01	[7:0]	Set to zeros [7:1]							Mode	
SYSTEM_ACTIVE	0x02	[7:0]	Set to zeros [7:1]							Enable "active" LED	
SYNTH_MODE	0x03	[7:0]	Set to zeros [7:2]					Fast-tune	Loop Gain		
RF_FREQUENCY	0x10	[7:0]	Frequency Word (Hz) [7:0]								
		[15:8]	Frequency Word (Hz) [15:8]								
		[23:16]	Frequency Word (Hz) [23:16]								
		[31:24]	Frequency Word (Hz) [31:24]								
		[39:32]	Frequency Word (Hz) [39:32]								
		[47:40]	Frequency Word (Hz) [47:40]								
IF_FREQUENCY	0x11	[55:48]	Set to zeros							LO	
		[7:0]	Frequency Word (Hz) [7:0]								
		[15:8]	Frequency Word (Hz) [15:8]								
		[23:16]	Frequency Word (Hz) [23:16]								
		[31:24]	Frequency Word (Hz) [31:24]								
		[39:32]	Frequency Word (Hz) [39:32]								
		[47:40]	Frequency Word (Hz) [47:40]								
RF_FILTER	0x14	[7:0]	Set to zeros [7:1]					Filter Value (0 to 7)			
ATTENUATOR	0x15	[7:0]	0	Attenuator value							
		[15:8]	Set to zeros [15:11]					Attenuator Number			
		[23:16]	Set to zeros [23:16]								
SIGNAL_PATH	0x16	[7:0]	If3 fil#2 select	If3 fil#1 select		If2 filter select	0	Bypass IF3 conv	Ext. IF2 select	Bypass convert	

Register Name	Reg Add	Serial Range	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
		[15:8]	Set to zeros [15:10]			Filter Value (0 to 7)			Auto RF Filter	Invert	
		[23:16]	Set to zeros [23:16]								
CONFIG_AUTO_GAIN	0x17	[7:0]	Set to zeros [7:5]			Linear mode	0	Load params	Auto Enable		
		[15:8]	Sign	Absolute RF level (0 - 127) dB [6:0]							
		[23:16]	Set to zeros								
		[31:24]	Sign	Absolute IF level (0 - 127) dB [6:0]							
		[39:32]	Set to zeros								
STORE_DEFAULT_STATE	0x18	[7:0]	Set all to zeros								
DEVICE_STANDBY	0x19	[7:0]	Set to zeros [7:4]				Device sections			Enable	
REFERENCE_CLOCK	0x1A	[7:0]	[7:4] set to zeros			PXI 10 enable	Clk 100 enable	Ref out enable	Lock ext		
REFERENCE_DAC	0x1B	[7:0]	DAC WORD [7:0]								
		[15:8]	0	0	DAC WORD [13:8]						
		[23:16]	Set to zeros								
LO1_PATH	0x1C	[7:0]	0	0	0	0	0	0	0	Ext/Int	
SYNTH_SELF_CAL	0x1D	[7:0]	Set all to zeros								
USER_EEPROM_WRITE	0x1E	[7:0]	BYTE DATA [7:0]								
		[15:8]	ADDRESS [7:0]								
		[23:16]	ADDRESS [15:8]								
FREQ_PLAN_PARAM	0x1F	[7:0]	Frequency word (Hz) [7:0]								
		[15:8]	Frequency word (Hz) [15:8]								
		[23:16]	Frequency word (Hz) [23:16]								
		[31:24]	Frequency word (Hz) [31:24]								
		[39:32]	Frequency word (Hz) [39:32]								
		[47:40]	Frequency word (Hz) [47:40]								
		[55:48]	Set to zeros [55:52]				PARAM [3:0]				

4.1.1 Register 0x01 INITIALIZE

This register initializes the devices to the default state or reprograms all the device components with the current state parameters. Note, this register does not need to be called in order to program the device.

Bytes written		2
Bytes read		1

Bits	Type	Name	Width	Description
[0]	WO	Mode	1	0 = Resets the current state 1 = Resets the default to the default or startup state
[7:1]	WO	Unused	7	Set to zeros
[7:0]	RO	Read back byte	8	Read 1 byte back is required for PXIe and RS232

4.1.2 Register 0x02 SYSTEM_ACTIVE

This register turns on or off the active LED indicator on the front connector interface of the device. This register should be called when the device is opened or closed in software.

Bytes written | 2
Bytes read | 1

Bits	Type	Name	Width	Description
[0]	WO	Mode	1	0 = Turns off the active LED 1 = Turns on the active LED
[7:1]	WO	Unused	7	Set to zeros
[7:0]	RO	Read back byte	8	Read 1 byte back is required for PXIe and RS232

4.1.3 Register 0x03 SYNTH_MODE (2 Bytes)

This register configures the PLL loop gain of the local oscillator synthesizers. It also enables or disables faster tuning of the YIG based oscillator of LO1.

Bytes written | 2
Bytes read | 1

Bits	Type	Name	Width	Description
[1:0]	WO	Loop Gain	2	0 = Low loop gain, improves phase noise > 50 kHz 1 = Normal loop gain 2 = High loop gain, improves phase noise < 50 kHz
[2]	WO	Fast Tune	1	0 = Turns fast tune off, default 1 = Turns fast tune on, close in phase noise may degrade
[7:3]	WO	Unused	5	Set to zeros
[7:0]	RO	Read back byte	8	Read 1 byte back is required for PXIe and RS232

4.1.4 Register 0x10 RF_FREQUENCY

This register tunes the device to the input RF frequency.

Bytes written | 8
Bytes read | 1

Bits	Type	Name	Width	Description
[47:0]	W	Frequency word	48	Frequency word in milli-Hz (mHz); mHz used for future compatibility
[48]	W	LO	1	Set to 1 to apply value directly to the first LO. The RF frequency value is unchanged.
[55:49]	W	Unused	7	Set to zeros
[7:0]	RO	Read back byte	8	Read 1 byte back is required for PXIe and RS232

4.1.5 Register 0x11 IF_FREQUENCY

This register sets the final IF value.

Bytes written		8
Bytes read		1

Bits	Type	Name	Width	Description
[47:0]	W	Frequency word	48	Frequency word in milli-Hz (mHz); mHz used for future compatibility
[55:48]	W	Used	8	Set to zeros
[7:0]	R	Read back byte	8	Read 1 byte back is required for PXIe and RS232

4.1.6 Register 0x14 RF_FILTER

This register selects the RF filter manually. This function only works if [bit 9] of register 0x16 is set to 1.

Bytes written		2
Bytes read		1

Bits	Type	Name	Width	Description
[2:0]	W	rfFilterSelect	3	Select the RF filter (0-7)
[7:3]	W	Unused	5	Set to zeros
[7:0]	R	Read back byte	8	Read 1 byte back is required for PXIe and RS232

4.1.7 Register 0x15 ATTENUATOR

This register sets the value of the device attenuators.

Bytes written		4
Bytes read		1

Bits	Type	Name	Width	Description
[7:0]	W	Attenuator value	8	In 0.25 dB LSB, i.e. LSB = 0.25 dB
[10:8]	W	The target attenuator	3	The attenuator number: 0 = RF Atten #1 1 = RF Atten #2 2 = IF2 Atten 3 = External IF input IF2 Atten 4 = IF3 Atten #1 5 = Unused
[23:11]	W	Unused	13	Set to zeros
[7:0]	R	Read back byte	8	Read 1 byte back is required for PXIe and RS232

4.1.8 Register 0x16 SIGNAL_PATH

This register configures how the signal chain is routed.

Bytes written		4
Bytes read		1

Bits	Type	Name	Width	Description
[0]	W	bypassConverter	1	0 = Normal conversion path 1 = Bypasses the conversion, switches RF input directly to IF3 output
[1]	W	If2ExtSelect	1	0 = Deselects 1 = Selects the IF2 input port
[2]	W	bypassIF3Conv	1	0 = IF2 is converted to IF3 1 = IF2 is switched to IF3 output port, bypassing mixer 3 conversion
[3]	W	If1Filter	1	Not implemented

Bits	Type	Name	Width	Description
[4]	W	If2Filter	1	0 = Selects the 160 MHz IF2 filter 1 = Selects the 80 MHz IF2 filter
[6:5]	W	If3Filter1	2	First filter bank of IF3: 0 = Selects the 500 MHz LPF 1 = Selects the 250 MHz LPF 2 = Selects a through path 3 = Not implemented
[7]	W	If3Filter2	1	Second filter bank of IF3: 0 = 1500 MHz LPF, this should always be selected when IF3 conversion is enabled 1 = A narrow 1.25 GHz bandpass filter
[8]	W	invertSpectrum	1	0 = Not spectral inversion 1 = IF3 spectrum inverted w.r.t RF
[9]	W	autoRfFilter	1	0 = Allows manual selection of RF filter 1 = Enables auto selection of RF filter based on RF frequency
[12:10]	W	rfFilterSelect	3	Selects RF filters 0 to 7.
[23:13]	W	Not used	11	Set to zeros
[7:0]	R	Read back byte	8	Read 1 byte back is required for PXIe and RS232

4.1.9 Register 0x17 CONFIG_AUTO_GAIN

This register will configure the device to set the attenuator values and control the RF amplifier to keep the device gain close to the desired gain with changes in RF frequency, as computed from its input parameters such as RF input level, mixer level, and IF out level.

Bytes written | 6
Bytes read | 1

Bits	Type	Name	Width	Description
[0]	W	autoGainEnable	1	0 = Attenuator states are not manipulated when RF values are changed 1 = Computes and applies new sets of attenuator values to the device as frequency is changed

Bits	Type	Name	Width	Description
[1]	W	loadParams	1	0 = Not to load new input parameter such as RF level, etc. Will not load any information from bits [31:2] 1 = Will load bits [31:2]
[2]	W	autoCntrlRfAmp	1	Set to zeros
[4:3]	W	linearMode	2	0 = Optimized linearity and noise 1 = Better SNR 2 = Better linearity
[7:5]	W	Unused	3	Set to zeros
[14:8]	W	rfLevel	7	Set the absolute level of the expected RF input level in dB.
[15]	W	rfLevel Sign bit	1	0 = Positive 1 = Negative
[22:16]	W	mixerLevel	7	Set to zero
[23]	W	mixerLevel Sign bit	1	Set to zero
[30:24]	W	ifLevel	7	Set the absolute level of the expected output IF3 level in dB
[31]	W	ifLevel Sign bit	1	0 = Positive 1 = Negative
[39:32]	W	Unused	8	Set to zeros
[7:0]	R	Read back byte	8	Read 1 byte back is required for PXIe and RS232

4.1.10 Register 0x18 STORE_DEFAULT_STATE

This register will store all current settings of the device into EEPROM and will become the default startup setting.

Bytes written		2
Bytes read		1

Bits	Type	Name	Width	Description
[7:0]	W	Unused	8	Set to zeros
[7:0]	R	Read back byte	8	Read 1 byte back is required for PXIe and RS232

4.1.11 Register 0x19 DEVICE_STANDBY

This register sets either the entire device or sections of the device into standby mode. Placing a section into standby involves powering down its circuitry. This conserves power and eliminates unwanted LO signals when they are not used.

Bytes written | 2
Bytes read | 1

Bits	Type	Name	Width	Description
[0]	W	State	1	0 = Active 1 = Standby enabled
[3:1]	W	Section	3	0 = Entire device 1 = LO1 2 = LO2 3 = LO3 4 = Signalchain
[7:4]	W	Unused	4	Set to zeros
[7:0]	R	Read back byte	8	Read 1 byte back is required for PXIe and RS232

4.1.12 Register 0x1A REFERENCE_CLOCK

This register configures the reference clock behavior.

Bytes written | 2
Bytes read | 1

Bits	Type	Name	Width	Description
[0]	W	LockEnable	1	0 = Uses internal 10 MHz TCXO 1 = Enables frequency locking to external 10 MHz reference source

Bits	Type	Name	Width	Description
[1]	W	RefOutEnable	1	Reference out port enabled
[2]	W	Clk10Enable	1	0 = 10 MHz output 1 = 100 MHz output
[3]	W	PXI10Enable	1	Only on SC5407A to enable export of the 10 MHz backplane clock
[7:4]	W	Unused	4	Set to zeros
[7:0]	R	Read back byte	8	Read 1 byte back is required for PXIe and RS232

4.1.13 Register 0x1B REFERENCE_DAC

This register makes adjustments to the 10 MHz TCXO accuracy via DAC to its tuning port.

Bytes written		4
Bytes read		1

Bits	Type	Name	Width	Description
[15:0]	W	Tuning DAC word	16	DAC WORD
[23:16]	W	Unused	8	Set to zeros
[7:0]	R	Read back byte	8	Read 1 byte back is required for PXIe and RS232

4.1.14 Register 0x1C LO1_PATH

This register sets the path of LO1; either it drives the internal IF1 mixer or drives out to the LO OUT port connector.

Bytes written		2
Bytes read		1

Bits	Type	Name	Width	Description
[0]	W	Path	1	0 = LO1 drives IF1 Mixer 1 = Drives LO1 to LO OUT port
[7:1]	W	Unused	7	Set to zeros

Bits	Type	Name	Width	Description
[7:0]	R	Read back byte	8	Read 1 byte back is required for PXIe and RS232

4.1.15 Register 0x1D SYNTH_SELF_CAL

This register will start the YIG synthesizer calibration. Note that although the calibration procedure takes about 6-8 seconds to complete, the register returns a byte almost immediately.

Bytes written | 2
Bytes read | 1

Bits	Type	Name	Width	Description
[7:0]	W	Unused	8	Set to zeros
[7:0]	R	Read back byte	8	Read 1 byte back is required for PXIe and RS232

4.1.16 Register 0x1E USER_EEPROM_WRITE

This register writes a byte to the onboard EEPROM.

Bytes written | 2
Bytes read | 1

Bits	Type	Name	Width	Description
[7:0]	W	Data	8	Byte to be written to the address
[23:8]	W	Address	16	EEPROM address
[7:0]	R	Read back byte	8	Read 1 byte back is required for PXIe and RS232

4.1.17 Register 0x1F FREQ_PLAN_PARAM

This register sets up the frequency plan parameters and stores them as the default values.

Bytes written | 8
Bytes read | 1

Bits	Type	Name	Width	Description
[47:0]	W	Data	48	Data for the parameter

Bits	Type	Name	Width	Description
[50:48]	W	Parameter	3	0 = RF frequency 1 = IF1 frequency 2 = IF2 frequency 3 = IF3 frequency
[55:51]	W	Unused	5	Set to zeros
[7:0]	R	Read back byte	8	Read 1 byte back is required for PXIe and RS232

4.2 Query Registers

These are request for data registers, in that a request for certain data is made by writing to the specific register first, followed by reading back the requested data. Some registers may require instruction data to specify the type of data to return, while others do not need any. For example, the register GET_DEVICE_PARAM (0x30) returns the RF Frequency, IF1 Frequency, IF3 Frequency, etc.; the data returned depends on the parameter value of the instruction byte.

Returned data length is always 8 bytes (64 bits), with the first byte being the most significant (MSB). It is **important that all 8 bytes** are read to clear the interface buffers. How the different interface buses handle the return data is explained in the *Device Information Parameters and Format* section.

Table 6. Query Registers

Register Name	Reg Add	Serial Range	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
GET_DEVICE_PARAM	0x30	[7:0]								Parameter	
GET_TEMPERATURE	0x31	[7:0]								Zeros [7:0]	
GET_DEVICE_STATUS	0x32	[7:0]								Zeros [7:0]	
GET_DEVICE_INFO	0x33	[7:0]								Info	
CAL_EEPROM_READ	0x35	[7:0]								EEPROM ADDRESS [7:0]	
		[15:8]								EEPROM ADDRESS [15:8]	
USER_EEPROM_READ	0x36	[7:0]								EEPROM ADDRESS [7:0]	
		[15:8]								EEPROM ADDRESS [15:8]	
SERIAL_OUT_BUFFER	0x37	[7:0]								Zeros [7:0]	
		[15:8]								Zeros [15:8]	
		[23:16]									Zeros [23:16]
		[31:24]									Zeros [31:24]
		[39:32]									Zeros [39:32]
		[47:40]									Zeros [47:40]
		[55:48]									Zeros [55:48]

4.2.1 Register 0x30 GET_DEVICE_PARAM

Write to this register the required device parameter to query from the device.

Bytes written | 2
Bytes read | 8

Bits	Type	Name	Width	Description
[3:0]	W	Parameter	1	0: Returns current RF frequency 1: Current IF1 frequency 2: Current IF2 frequency 3: Current IF3 frequency 4: Current LO1 frequency 5: Current LO2 frequency 6: Current LO3 frequency 7: Current attenuator values 8: Current signalchain configuration
[7:4]	W	Unused	4	Set to zeros
[63:0]	R	Read back bytes	64	Returned Data (See Device Parameter Data section for more info)

4.2.2 Register 0x31 GET_TEMPERATURE

Write to this register to query the device temperature.

Bytes written | 2
Bytes read | 8

Bits	Type	Name	Width	Description
[7:0]	W	Unused	7	Set to zeros
[31:0]	R	Temperature Data	32	These 32 bits of data need to be type casted back to float. i.e. <code>var_float = *(float*)&var_u32</code> where <code>var_u32</code> is some unsigned integer that holds the 32 bits of read data
[63:32]	R	Invalid data	32	Ignore

4.2.3 Register 0x32 GET_DEVICE_STATUS

Write to this register to query the current operating conditions.

Bytes written		2
Bytes read		8

Bits	Type	Name	Width	Description
[7:0]	W	Unused	7	Set to zeros
[0]	R	PLL status: LO1 sum	1	The summing PLL of LO1
[1]	R	PLL status: LO1 crs	1	The coarse tuning PLL of LO1
[2]	R	PLL status: LO1 fine	1	The fine tuning PLL of LO1
[3]	R	PLL status: LO2	1	LO2
[4]	R	PLL Status: LO3	1	LO3
[5]	R	PLL status: VCXO	1	100 MHz OCXO
[6]	R	PLL status: TCXO	1	TCXO, only valid when lock to external reference is enabled
[7]	R	PLL status: ref_detect	1	Detected a reference signal of sufficient amplitude
[8]	R	Device Accessed	1	When the SET SYS ACTIVE register is accessed
[9]	R	LockExtRef	1	Enable device to lock to external reference
[10]	R	RefOutEnable	1	Reference output enable
[11]	R	RefOutFreqSelect	1	10MHz(0) or 100MHz(1) selected as output reference frequency
[12]	R	Lo1PwrUp	1	LO1 synth power up
[13]	R	Lo2PwrUp	1	LO2 synth power up
[14]	R	Lo3PwrUp	1	LO3 synth power up
[16:15]	R	loopGain	2	Loop gain of the Synthesizers

Bits	Type	Name	Width	Description
[17]	R	fastTune	1	LO1 set to fast tune
[18]	R	Lo1OutEnable	1	LO1 route to output port
[23:19]	R	Reserved	2	Invalid
[24]	R	scPwrUp	1	Signalchain power up
[25]	R	bypassConv	1	RF conversion bypassed, direct to IF3
[26]	R	If2ExtSelect	1	IF2 input from external port select
[27]	R	bypassIF3Conv	1	IF3 conversion bypassed, IF2 directed to IF3 output port
[28]	R	Reserved	1	Ignore
[29]	R	IF2FilterSelect	1	IF2 filters: 0 = 160 MHz, 1 = 80 MHz
[31:30]	R	Reserved	2	IF3 filter bank #1 filter select
[32]	R	Reserved	1	IF3 filter bank #2 select
[33]	R	invertSpectrum	1	RF amplifier enable
[34]	R	Reserved	1	Spectrum at IF3 is inverted
[35]	R	autoGainEnable	1	Auto calculation of gain enabled (default 0). Software gain calculation used.
[36]	R	Reserved	1	Auto set the RF preamplifier when autoGainEnable = 1
[37]	R	autoRfFilter	1	Auto set the RF filter as frequency is changed
[63:38]	R	Invalid data	29	Ignore

4.2.4 Register 0x33 GET_DEVICE_INFO

Write to this register to query the device information such as serial number.

Bytes written		2
Bytes read		8

Bits	Type	Name	Width	Description
[3:0]	W	DeviceInfo	4	0: Product serial number and Interface type 1: Hardware revision 2: Firmware revision 3: Device Interface 4: Manufacture date 5: Last calibration date
[7:4]	W	Unused	4	Ignore
[63:0]	R	Data	64	Information data (see Device Info data)

4.2.5 Register 0x35 CAL_EEPROM_READ

Write to this register to query 8 bytes of data from the calibration EEPROM at the starting address.

Bytes written	4
Bytes read	8

Bits	Type	Name	Width	Description
[15:0]	W	Address	16	Starting EEPROM address
[23:16]	W	Unused	8	Zeros
[63:0]	R	Data	64	8 bytes of data, LSB is the byte at the start address

4.2.6 Register 0x36 USER_EEPROM_READ

Write to this register to query 8 bytes of data from the user EEPROM at the starting address.

Bytes written	4
Bytes read	8

Bits	Type	Name	Width	Description
[15:0]	W	Address	16	Starting EEPROM address
[23:16]	W	Unused	8	Zeros
[63:0]	R	Data	64	8 bytes of data, LSB is the byte at the start address

4.2.7 Register 0x37 SERIAL_OUT_BUFFER

Writing to this register only provides the 64 clock edges (Reg + 7 data bytes) to transfer serial data from the device through SPI. Other interfaces do not use this register.

Bytes read | 8

Bits	Type	Name	Width	Description
[55:0]	W	Unused	56	Zeros, just to provide clocking for SPI data from device
[63:0]	R	Data	64	8 bytes of data

4.2.8 Device Parameters Data and Format

The data read back from the GET_DEVICE_PARAM register has a total of 8 bytes, however not all bytes contain valid data. The table below shows the valid data for each of the parameters.

Table 7. Device parameter data

Device Parameter Name	Param num	Data type	Byte 7	Byte 6	Byte 5	Byte 4	Byte 3	Byte 2	Byte 1	Byte 0
RF FREQ	0	U64	zeros	Data						
IF1 FREQ	1	U64	zeros	Data						
IF2 FREQ	2	U64	zeros	Data						
IF3 FREQ	3	U64	zeros	Data						
LO1 FREQ	4	U64	zeros	Data						
LO2 FREQ	5	U64	zeros	Data						
LO3 FREQ	6	U64	zeros	Data						
ATTEN VALUES	7	8xU8	zeros	NA	RF A1	RF A2	IF2 A	EXT IF	IF3 A1	NA
CHAIN CONFIG	8	2xU16	zeros	zeros	zeros	zeros	SC Gain		SC Config	

4.2.8.1 Frequency parameter values

Frequency parameter values are returned in the first 7 bytes with the last byte padded with zeros. The least significant bit (LSB) is 1/1000 Hertz (the data is returned in 1 one thousandth of a Hertz). Divide the data by 1000 to obtain the result in Hertz.

4.2.8.2 Attenuator values

Each attenuator value is returned as one byte, and the LSB is in 0.25 dB. Divide each byte by 4 to obtain the result in dB.

4.2.8.3 Signal Chain Configuration

The signal chain configuration data is in the first 2 bytes, and their data is represented in the following table.

Table 8. Signal chain configuration data representation

Bit	Description
[0]	Bypass Conversion
[1]	IF2 external port select
[2]	Bypass IF3 conversion
[3]	NA
[4]	IF2 filter selection 0 = 160 MHz, 1 = 80 MHz
[6:5]	IF3 Filter 1 selection
[7]	IF3 Filter 2 selection
[8]	IF spectral inversion
[9]	RF preamplifier state

The first order computed gain of the device is returned in bytes 3 and 4. Data bits [14:0] represent the absolute value in 1/100 of a dB, while bit [15] is the sign bit.

4.2.9 Device Information Parameters and Format

Not all the 8 bytes read back contain valid data. The following table shows the valid bytes of data for each of the parameters.

Device Parameter Name	Param num	Data type	Byte 7	Byte 6	Byte 5	Byte 4	Byte 3	Byte 2	Byte 1	Byte 0
INTERFACE	0	U32				Intrfce	Serial Number			
REVISIONS	1	F32	Software revision				Hardware revision			
DATE	2	U32	Cal year	Month	Day	Mfg year	Month	Day		

4.2.9.1 Interface Information

The first 4 bytes contain the device serial number as an unsigned 32-bit integer. Byte [4] contains the interface as represented in the following table.

Table 9. Interface ID

Bit	Description
[0]	PXI/PXLe (1 if available)
[1]	USB

Bit	Description
[2]	SPI
[3]	RS232
[7:4]	Undefined

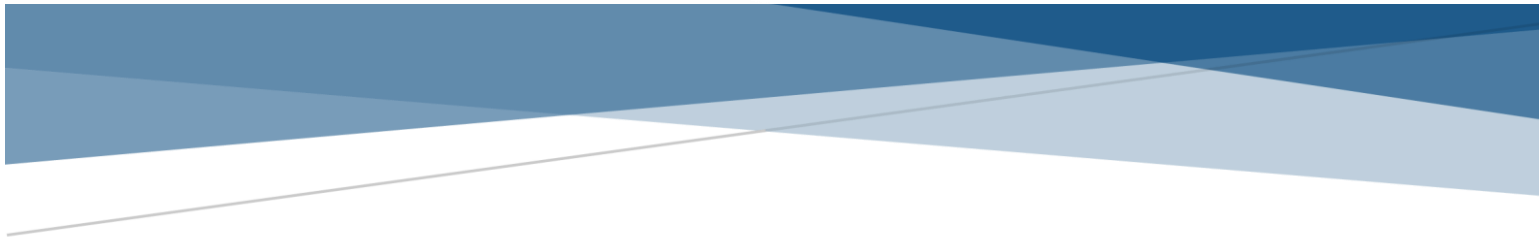
4.2.9.2 *Revision Information*

The first 4 bytes represent the hardware revision and last 4 bytes represent the firmware revision of the device. These 4 bytes encompass a 32-bit floating point number so the data needs to be type casted from an unsigned 32-bit value to float value.

4.2.9.3 *Date Information*

The first 4 bytes represent the manufactured date and the last 4 bytes represent the last calibration date. The date format is outlined in the following table.

Bit	Type	Description
[0]	U8	Day
[1]	U8	Month
[4:3]	U16	Year (i.e. 2016)



Section 2

Communication Interfaces and Calibration

5 Communication Interfaces

The SC5407A has a PXI express interface, while the SC5408A has 2 communication interfaces:

1. USB and SPI
2. USB and RS232

This section will examine the communication aspects of the product, focusing on data transfer to and from the device on each interface. Although the registers are identical for all interfaces, there are subtle differences in the implementation of the interfaces to transfer the data.

5.1 Communication Data Format

All data sent and received by all interfaces is sent as buffers of unsigned bytes. For example, to change RF frequency of the device to 6 GHz we perform the following:

1. Frequency is sent in 1000th of Hertz, so the data that represents the frequency is 6,000,000,000,000 milli-Hertz.
2. This number can be represented by a 64-bit unsigned long, and in Hexadecimal is 0x 0000 0574 FBDE 6000. Only the least 6 bytes are needed to represent all frequencies allowable for this device.
3. A byte data buffer needs to be 8 bytes for register RF_FREQUENCY (address 0x10), so the byte array buffer to be sent would be:

```
[0x10][0x00][0x05][0x74][0xFB][0xDE][0x60][0x00]
```

The register address byte [0x10] is the first member of the buffer to be sent.

5.2 USB Interface

There are 2 transfer types for the USB interface.

- Control transfer
- Bulk transfer

5.2.1 Control Transfer

The USB control transfer parameters are:

ENDPOINT_IN	0x80
ENDPOINT_OUT	0x00
TYPE_VENDOR	0x40
RECIP_INTERFACE	0x01

5.2.2 Bulk Transfer

The USB bulk transfer parameters are:

ENDPOINT_IN	0x83
ENDPOINT_OUT	0x04

The bulk transfer from the host to the device operates on a loopback with a data buffer of 8 bytes. When a device register is addressed, and upon completion of the register task, such as changing frequency, it will send back 8 bytes, which the host must read to clear the transfer buffers. Unlike the other interface methods, where only the required number of bytes needs to be sent for a given register, 8 bytes are needed for every USB bulk transfer. For example, if a configuration register requires only 4 bytes to be sent, these bytes will be the first of the 8 bytes and the last 4 bytes are zeros. The returned 8 bytes do not carry valid data for a configuration register. However, they do carry valid data for query registers.

5.3 SPI Interface

The SPI interface on the device is implemented using an 8-bit (single Byte) buffer for both the input and output, hence, it needs to be read and cleared by the device before consecutive bytes can be transferred to and from it. The process of clearing the SPI buffer and decisively moving it into the appropriate register takes CPU time, so a time delay is required between consecutive bytes written to or read from the device by the host. The chip-select pin (\overline{CS}) must be asserted low before data is clocked in or out of the product. Furthermore, pin \overline{CS} must be asserted low for the entire duration of a register transfer.

Once a full transfer has been received, the device will proceed to process the command and de-assert low the SRDY pin. The status of this pin may be monitored by the host because when it is de-asserted low, the device will ignore any incoming data. The device SPI is ready when the previous command is fully processed and the SRDY pin is re-asserted high. It is important that the host either monitors the SRDY pin or waits for 500 μs between register writes.

There are 2 SPI modes: 0 and 1. The default mode is 1, where data is clocked in and out of the device on the falling edge of the clock signal. In mode 0, data is clocked in and out on the rising edge. To select mode 0, pin 23 of the interface connector must be pulled low to ground as the device is powered on or as the reset line (pin 19) is toggled low-high. If pin 23 is pulled high or left unconnected, mode 1 is selected.

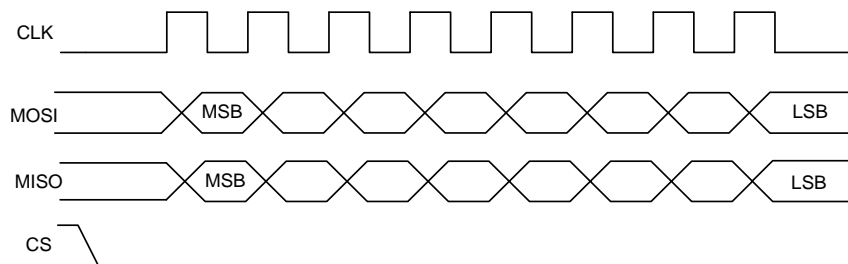


Figure 8. SPI Mode 1 shown.

Register writes are accomplished in a single write operation. Register lengths vary depending on the register. They vary in lengths of 2 to 8 bytes, with the first byte sent being the register address followed by the data associated with that register. The (\overline{CS}) pin must be asserted low for a minimum period of 1 μs (T_s , see Figure 9) before data is clocked in, and must remain low for the

entire register write. The clock rate may be as high as 5.0 MHz ($T_C = 0.2 \mu s$), however, if the external SPI signals do not have sufficient integrity due to trace issues, the rate should be lowered.

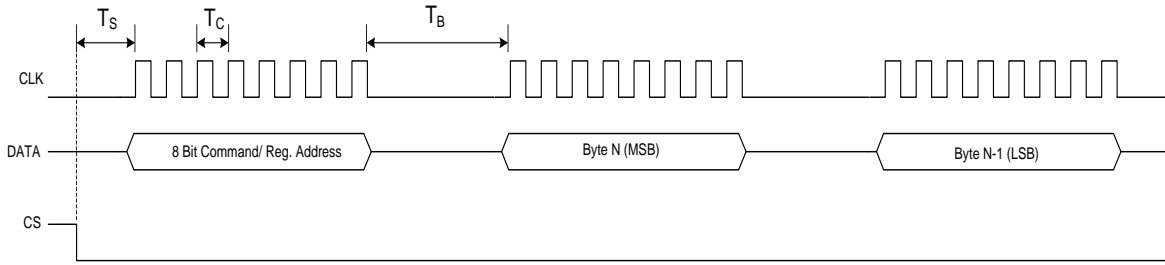


Figure 9. SPI timing.

As mentioned above, the SPI architecture limits the byte rate since after every byte transfer the input and output SPI buffers need to be cleared and loaded respectively by the device SPI engine. Data is transferred between the input buffer and internal register buffers. The time required to perform this task is indicated by T_B , which is the time interval between the end of one byte transfer and the beginning of another. The recommended minimum time delay for T_B is $1 \mu s$. The number of bytes transferred depends on the register. It is important that the correct number of bytes is transferred for the associated register, because once the first byte (MSB) containing the device register address is received, the device will wait for the desired number of associated data bytes. The device will hang if an insufficient number of bytes are written to the register. To clear a hung condition, the device will need an external hard reset. The time required to process a command is also dependent on the command itself. Measured times for command completions are typically between $50 \mu s$ to $300 \mu s$ after reception.

5.3.1 Writing the SPI Bus

The SPI transfer size (in bytes) depends on the register being targeted. The first byte sent is the register address and subsequent bytes contain the data associated with the register. As data from the host is being transferred to the device via the MOSI line, data present on its SPI output buffer is simultaneously transferred back, MSB first, via the MISO line. The data returned is invalid for configuration registers. The following figure shows the contents of a single 3-byte SPI command written to the device. The *Hardware Registers* section provides information on the number of data bytes and their contents for an associated register. There is a minimum of 1 data byte for each register even if the data contents are “zeros”.

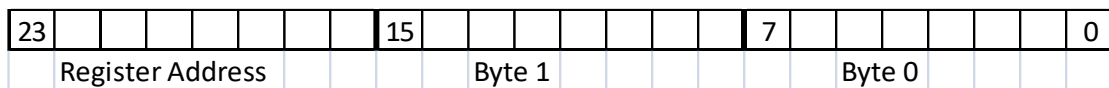


Figure 10. Example of a 3-byte write

5.3.2 Reading the SPI Bus

Data is simultaneously read back during an SPI transfer cycle. Requested data from a prior command is available on the device SPI output buffers, and these are transferred back to the user host via the MISO pin. To obtain valid requested data would require querying the SERIAL_OUT_BUFFER, which requires 8 bytes or 64 clock cycles; 1 byte for the device register

(0x37) and 7 empty bytes (MOSI) to clock out the returned data (MISO). An example of reading the device RF parameters (IF3 frequency) from the device is shown in the following figure.

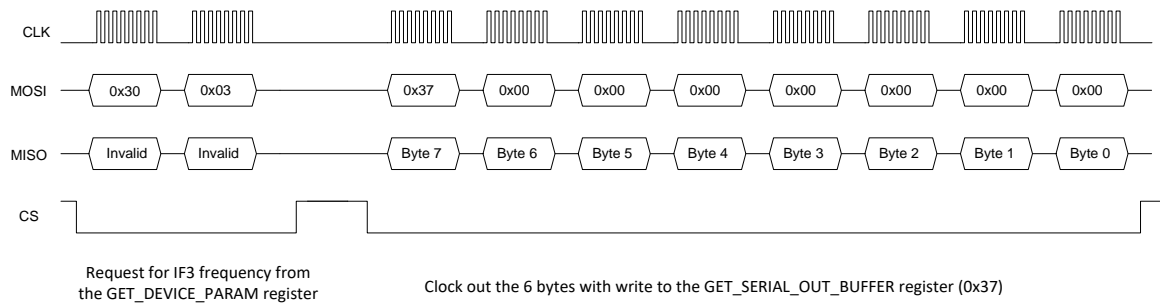


Figure 11. Query example: Write followed by Read to the GET_SERIAL_BUFFER

In the figure above, the first transfer cycle is to make the request for IF3 frequency data through the GET_DEVICE_PARAM register. The subsequent cycle is to clock the data that was requested by sending 64 clocks into the GET_SERIAL_BUFFER register.

5.4 RS232 Interface

The RS232 version of the SC5408A has a standard interface buffered by an RS232 transceiver so that it may interface directly with many host devices, such as a desktop computer. The interface connector for RS232 communication is labeled “Digital I/O” on the front of the panel. Refer to *Table 3. Interface connector pin out description* for position and pin-out information. The RS232 device communication control set is provided in the following table.

Table 10. RS232 Control Setting

Baud rate	Rate of transmission, 115200 if pin 23 is GND, 57600 if pulled high or open
Data bits	The number of bits in the data is fixed at 8.
Parity	Parity is 0 (zero)
Stop bits	1 stop bit
Flow control	0 (zero) or none

Only 3-wire RS232 is required since hardware flow control is not used. These connections are the Tx, Rx, and Gnd. This interface is common on most host computers and microcontrollers, so user access to host ports is readily provided by the computer OS or microcontroller hardware registers.

5.4.1 Writing to the Device Via RS232

It is important that all necessary bytes associated with any one register are fully sent. In other words, if a register requires a total of 6 bytes (address plus data), then all 6 bytes must be sent even though the last byte may be null. The device, upon receiving the first register addressing byte, will wait for all the associated data bytes before acting on the register instruction. Failure to complete the register transmission will cause the device to behave erratically or hang. Information

for writing to the configuration registers is provided in *Table 5. Configuration Registers*. Upon the execution of the register that was sent, the device will return one (1) byte of data with bit 1 high to indicate success. This byte must be read by the host to clear its receive buffer so that reading subsequent registers will not contain corrupted data. Furthermore, reading back this byte will ensure that the device is ready for the next register command.

5.4.2 Reading from the Device Via RS232

To query information from the device, the query registers are addressed, and data is returned. *Table 6. Query Registers* contains the query register information. As with the configuration registers, it is important that all data byte(s) (write) associated with the query registers are sent even if they are null. All queries will return 8 bytes of data (read) with the first received byte being the most significant (MSB). Section 4.2 *Query Registers* provides the format details of the received data.

5.5 PXI Express

The PXIe interface contains a high-speed PCIe-to-Serial bridge chip. This bridge chip communicates with the onboard microcontroller serially. The interface on the bridge chip resides at offset addresses between 0x00 and 0xFF from BAR0; which is memory mapped. A kernel level driver for the operating system is needed to access this memory address. A simple driver using IO controls should be sufficient to read and write byte data to this block of addresses. Although SignalCore provides the driver and API for these products, information is provided here for users who may need to write drivers for a different operating system or a different driver. An example would be writing the API for the Linux operating system.

5.5.1 Setting Up the PCI to Serial Bridge

The serial function of the bridge chip must first be initialized before it can communicate with the onboard microcontroller, and hence communication between the microcontroller and the PXIe bus. The initialization can be done at the kernel level mode or at the user level mode, the decision is left to the user. The following table lists the programming order of the bridge register addresses to initialize and setup the serial port function.

Step	BAR0 Register Address	Data (Byte)
1	0x10CC	0x00
2	0x1004	0x80
3	0x10C2	0x04
4	0x10C1	0x00
5	0x10C3	0x01
6	0x1003	0x80
7	0x1000	0x10
8	0x1001	0x00
9	0x1003	0x03
10	0x1002	0x07

5.5.2 Writing to the Device

Bytes that are written to the device must go through the bridge chip. In this section, we will first look at the write cycle of each byte, and then the write cycle of each device register. Do note the difference between the bridge register addresses and the device register addresses.

5.5.2.1 *Single Byte Write*

The serial transfer buffer register address is located at 0x1000 offset from BAR0 of the bridge chip, however, before writing byte data to this register, its status needs to be checked to confirm that it is ready to accept a new buffer set of bytes. The status register is located at 0x1005; it must be read and bit 7 must be **high** to indicate that the transfer register is ready to receive the next byte buffer. Checking the status register of the serial bridge chip is required before every new command write.

5.5.2.2 *Device Register Write*

The process of writing the device registers is the same as writing an RS232 port, so the description of Section 5.4.1 *Writing to the Device Via RS232* is applicable. Writing the device registers involves sending byte-by-byte data as described previously. Section 4.1 *Configuration Registers* provides information on the number of configuration write bytes needed for each device register. The first byte sent is the device register address, followed by the most significant byte of the register's associated data. When a device register is fully written, that is, all its data has been sent to the device, it will return 1 byte. This returned byte must be read (by the host) to clear the transfer buffer so that later received data are not corrupted. Section 5.5.3.1 *Single Byte Read* describes how a byte read cycle is performed.

5.5.3 Reading from the Device

Device data is passed back to the host via the bridge chip byte-by-byte. Details about a single byte read process and an entire register read process are explained in the following subsections.

5.5.3.1 *Single Byte Read*

The serial transfer buffer register address is located at 0x00 offset from BAR0 of the bridge chip. Before valid data can be read from the transfer register, its ready status must first be confirmed. The status register is located at 0x05; it must be read and bit 0 must be **high** to indicate that valid data is available. Checking the status register for available data is required before **every byte** read.

5.5.3.2 *Device Register Read*

After a write request to the device is made, 8 bytes of data is available to be read back. Use the single byte read process, as mentioned previously, to read all the bytes. See Section 4.2 *Query Registers* for information of the exact number of request write bytes and the number of request read bytes, which is 8. All 8 bytes must be read to fully clear the transfer buffer; the first byte read is the most significant byte.

6 Calibration

6.1 Calibration EEPROM Map

Table 11 represents the EEPROM map of the device calibration values. All values are stored as little-endian 4-byte floating point numbers. Every point is 4-bytes long. Access to the data is possible through the CAL_EEPROM_READ register, which reads 8 bytes starting at the address pointed to by the register input.

Table 11. Calibration EEPROM Map

Offset address	Points	Length (Bytes)	Data Type	Description
0x00	664	664	U_8	Factory Reserved
0x298	1	4	Float_32	Calibration Temperature
0x29B	2	8	Float_32	Temp coeff C1, C2
0x2B8	3x8	96	Float_32	
0x338	22	88	Float_32	IF response cal frequencies 25, 50, 75, ..., 550 in MHz
0x390	22	88	Float_32	IF3_response, relative gain to 225 MHz at each calibration frequency
0x3E8	22	88	Float_32	N/A
0x5CC	30	120	Float_32	Relative If3_atten#1, 1-30 dB, 1 dB step
0x644	30	120	Float_32	Relative If2_atten, 1-30 dB, 1 dB step
0x6CC	1	4	Float_32	Relative Bypass IF3 gain, dB
0x6D0	1	4	Float_32	N/A
0x6D4	30	120	Float_32	Relative Bypass IF3_atten#1, 1-30 dB, 1 dB step
0x7D8	1	4	Float_32	Relative IF3_invert gain
0x7DC	1	4	Float_32	Relative IF2_Low BW gain
0x7E8	17	68	Float_32	Bypass Conv frequencies (start 50 MHz, 50 MHz steps to 850 MHz, 17 points) in MHz
0x82C	17	68	Float_32	Bypass Conversion Absolute gain at each freq
0x870	17x30	2040	Float_32	Bypass Relative IF3#2 atten at each freq. a1_f0, a1_f1,...,a1_fN a2_f0, a2_f1,...,a2_fN a30_f0, a30_f1,...,a30_fN

Offset address	Points	Length (Bytes)	Data Type	Description
0x1868	1	4	Float_32	IF2 ext absolute gain dB
0x186C	30	120	Float_32	Relative IF2ext Attenuator attenuation, 1-30 dB, 1 dB step
0x18F8	62	248	Float_32	RF calf req, 100-6200, every 100, in MHz
0x19F0	62	248	Float_32	RF absolute gain at every freq thru path
0x1AE8	62	248	Float_32	Relative RF gain at every freq filter path
0x1BE0	62x30	7440	Float_32	RF atten#1, 1-30 dB, 1 dB step for each freq a1_f0, a1_f1,...,a1_fN a2_f0, a2_f1,...,a2_fN a30_f0, a30_f1,...,a30_fN
0x38F0	62x30	7440	Float_32	RF atten#2, 1-30 dB, 1 dB step for each freq a1_f0, a1_f1,...,a1_fN a2_f0, a2_f1,...,a2_fN a30_f0, a30_f1,...,a30_fN

6.2 Absolute Conversion Gain

It is difficult to make gain measurements of every configuration setting of the upconverter as there are many combinations of attenuator and frequency settings, not to mention the various operating temperatures. However, the gain of the device can be computed to a reasonable level of accuracy with less data if the assumption is that every component setting is independent from one another. For example, assume the attenuator state of the RF_ATTEN#1 does not affect the state of RF_ATTEN#2, so with this independence there are only 2*30 attenuation measurements that need to be made; there is a mutually exclusive relationship. On the other hand, if they are not independent, for every setting of RF_ATTEN#1, 30 measurements are needed for RF_ATTEN#2, bringing the total number of measurements to 30*30.

6.3 Absolute Gain of the RF Conversion Path

Assuming independency, and to be able to compute the gain of the upconverter at any setting, an absolute reference state of the device must be determined first. Once this reference state is established, compensation can be applied to other relative configured states of the device. The established reference state of the device is as follows:

- All attenuators are set to 0 dB
- IF2 Filter setting is 0, or 160 MHz filter selected
- IF3_Filter1 setting is 0, 500 MHz LPF selected

- IF3_Filter2 setting is 0, 1500 MHz LPF is selected
- IF3 bypass conversion is 0, RF is converted to by mixer 3
- IF frequency is set at 150 MHz
- RF is tuned from 100 MHz to 6200 MHz in 100 MHz steps

The gain of the device in this state is measured at different frequencies in the range of 100 MHz to 6.2 GHz. This set of measurements is the RF absolute gain stored at starting address 0x19F0 of the calibration EEPROM. All other measurements taken as deviations from this reference setting are relative.

6.4 Gain Through the Bypass RF Conversion Path

The absolute gain through the bypass path is stored beginning at address 0x82C. The absolute gain measurement is made every 50 MHz, between 50 MHz and 500 MHz.

6.4.1 Applying Calibration

The gain and attenuator values are relative measurements from the absolute reference values made over various RF frequencies. These relative values are either subtracted (attenuation) or added (gain) to the absolute gain value to determine the gain of the relative configuration. For example, assume the RF_ATTEN#1 is at 20 dB, IF3_ATTEN#1 is at 10 dB, IF2_Filter bank is set to select the 80 MHz filter, IF3 frequency is set to 140 MHz, and the RF frequency is set to 1.550 GHz. A possible systematic approach would be:

1. Compute the relative RF Atten#1 value by interpolation because there is no measurement value at 1.55 GHz. A simple linear interpolation between the measured 1.5 GHz and 1.6 GHz values would provide a good estimate. A local spline interpolation over 5-6 surrounding points would provide better accuracy. Let us call this value *rf_atten1*.
2. The IF3 Atten#1 attenuation value is read directly from memory; however, it must be corrected for frequency offset from 150 MHz. There are offset gain response values measured at 125 MHz and 150 MHz, so a simple linear interpolation between 2 points should be sufficient. Let us call this correct attenuation value *if3_atten1*.
3. The IF2 filter gain is read directly from memory, which we shall call *if2_filter*.
4. The absolute gain at the frequency can be determined using interpolation between 2 absolute gain points, which we shall call *G_abs*.
5. The gain for this configuration is calculated using:

$$G = G_{abs} + if_{filter} - rf_{atten1} - if3_{atten1}$$

If the current device temperature is different from the calibration temperature, the gain correction due to temperature difference is computed using

$$\Delta G_{temp} = C_1(T - T_0) + C_2(T - T_0)^2 .$$

ΔG_{temp} is the gain correction, C_1 and C_2 are the first and second order temperature gain coefficients respectively, and T and T_0 are the current temperature and calibration temperature respectively. Adding this correction to the previously calculated gain will compensate the value for temperature deviation.

$$G = G_{abs} + if_{filter} - rf_{atten1} - if3_{atten1} + \Delta G_{temp}$$

Revision Table

Revision	Revision Date	Description
1.0	4/2/17	Initial Release
1.1	6/3/19	Address Removed
2.0	8/5/20	Updated Formatting
2.1	7/21/21	Removed note about Pin 16 from Table 9
2.2	12/7/22	Corrected torque range

