CMOS 200 MSPS 14-Bit Quadrature Digital Upconverter

AD9857

FEATURES

200 MHz Internal Clock Rate
14-Bit Data Path
Excellent Dynamic Performance
80 dB SFDR @ 65 MHz (±100 kHz) A_{OUT}
4×–20× Programmable Reference Clock Multiplier
Reference Clock Multiplier PLL Lock Detect Indicator
Internal 32-Bit Quadrature DDS
FSK Capability
8-Bit Output Amplitude Control
Single-Pin Power-Down Function
Four Programmable, Pin-Selectable Signal "Profiles"
SIN(x)/x Correction (Inverse SINC Function)
Simplified Control Interface
10 MHz Serial, 2- or 3-Wire SPI-Compatible
3.3 V Single Supply
Single-Ended or Differential Input Reference Clock
80-Lead LQFP Surface-Mount Packaging
Three Modes of Operation
Quadrature Modulator Mode
Single-Tone Mode
Interpolating DAC Mode

APPLICATIONS

HFC Data, Telephony, and Video Modems
Wireless Base Station
Agile, L.O. Frequency Synthesis
Broadband Communications

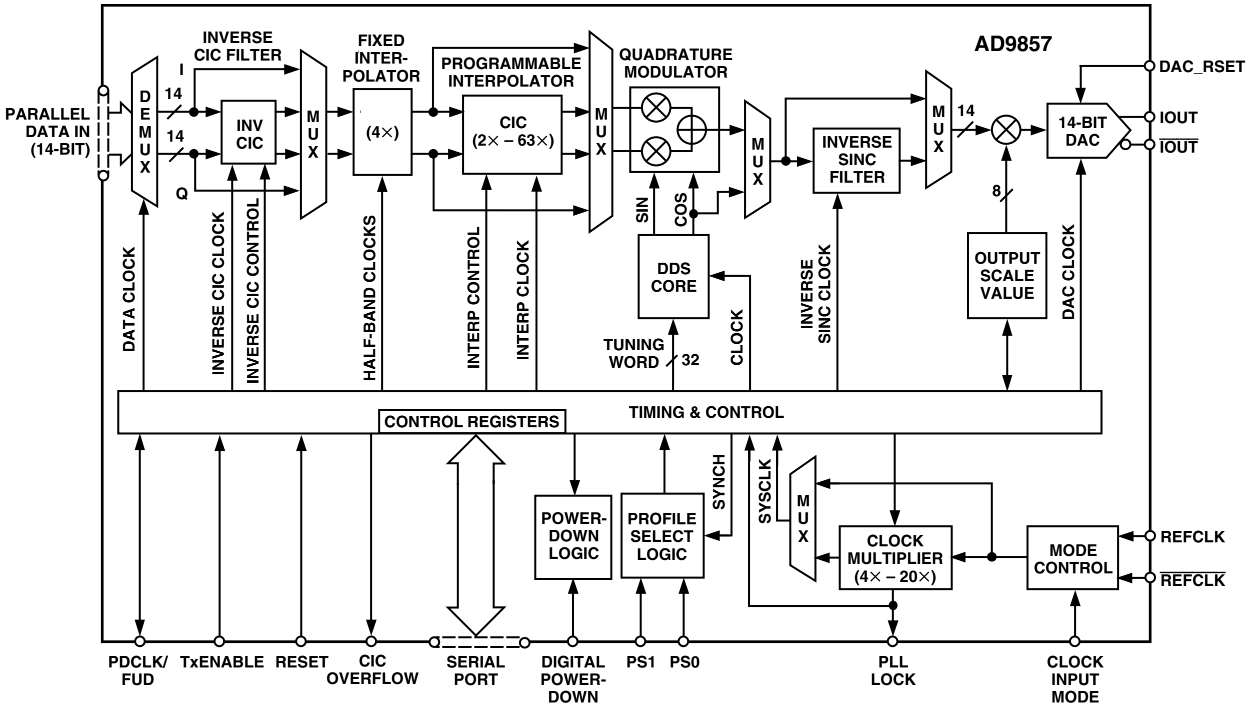
GENERAL DESCRIPTION

The AD9857 integrates a high-speed Direct Digital Synthesizer (DDS), a high-performance, high-speed 14-bit digital-to-analog converter (DAC), clock multiplier circuitry, digital filters, and other DSP functions onto a single chip, to form a complete quadrature digital upconverter device. The AD9857 is intended to function as a universal I/Q modulator and agile upconverter, single-tone DDS, or interpolating DAC for communications applications, where cost, size, power dissipation, and dynamic performance are critical attributes.

The AD9857 offers enhanced performance over the industry-standard AD9856, as well as providing additional features.

The AD9857 is available in a space-saving surface-mount package and is specified to operate over the extended industrial temperature range of –40°C to +85°C.

FUNCTIONAL BLOCK DIAGRAM



REV. B

Information furnished by Analog Devices is believed to be accurate and reliable. However, no responsibility is assumed by Analog Devices for its use, nor for any infringements of patents or other rights of third parties that may result from its use. No license is granted by implication or otherwise under any patent or patent rights of Analog Devices.

One Technology Way, P.O. Box 9106, Norwood, MA 02062-9106, U.S.A.
Tel: 781/329-4700
www.analog.com
Fax: 781/326-8703

TABLE OF CONTENTS

MODES OF OPERATION

The AD9857 has three operating modes:

- Quadrature Modulation Mode (Default)
- Single-Tone Mode
- Interpolating DAC Mode

Mode selection is accomplished by programming a control register via the Serial Port. The Inverse SINC Filter and output scale multiplier are available in all three modes.

Quadrature Modulation Mode

In Quadrature Modulation Mode, both the I and Q data paths are active. A block diagram of the AD9857 operating in the Quadrature Modulation Mode is shown in Figure 1.

In Quadrature Modulation Mode, the PDCLK/FUD pin is an output and functions as the Parallel Data Clock (PDCLK), which serves to synchronize the input of data to the AD9857. In this mode, the input data must be synchronized with the rising edge of PDCLK. The PDCLK operates at *twice* the rate of either the I or Q data path. This is due to the fact that the I and Q data must be presented to the parallel port as two 14-bit words multiplexed in time. One I word and one Q word together comprise one internal *sample*. Each sample is propagated along the internal data pathway in parallel fashion.

The DDS core provides a quadrature (sin and cos) local oscillator signal to the quadrature modulator, where the I and Q data are multiplied by the respective phase of the carrier and summed together, to produce a quadrature-modulated data stream.

All of this occurs in the digital domain, and only then is the digital data stream applied to the 14-bit DAC to become the quadrature-modulated analog output signal.

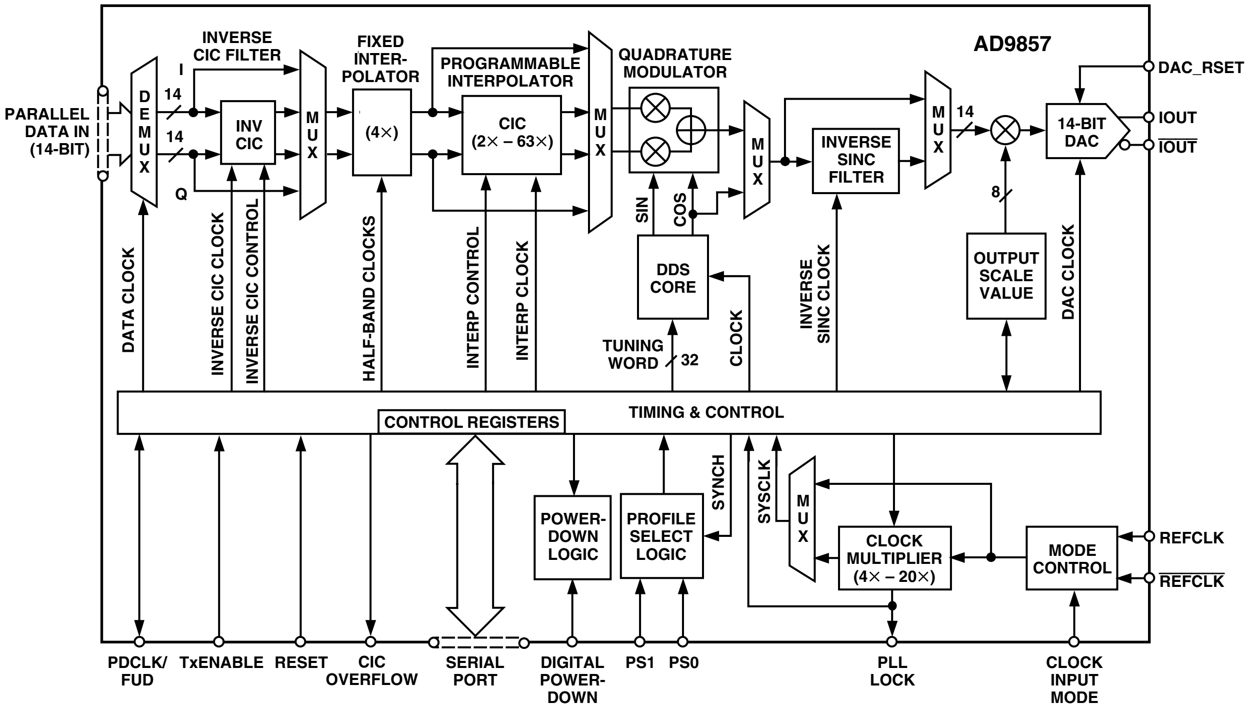


Figure 1. Quadrature Modulation Mode

Single-Tone Mode

A block diagram of the AD9857 operating in the Single-Tone Mode is shown in Figure 2. In the Single-Tone Mode, both the I and Q data paths are disabled from the 14-bit Parallel Data Port up to and including the modulator. The PDCLK/ FUD pin is an input and functions as a Frequency Update (FUD) control signal. This is necessary because the frequency tuning word is programmed via the asynchronous serial port. The FUD signal causes the new frequency tuning word to become active.

In Single-Tone Mode, the cosine portion of the DDS serves as the signal source. The output signal consists of a single frequency as determined by the tuning word stored in the appropriate control register, per each profile.

In the Single-Tone Mode, no 14-bit parallel data is applied to the AD9857. The internal DDS core is used to produce a single frequency signal according to the tuning word. The single-tone signal then moves toward the output, where the Inverse SINC Filter and the output scaling can be applied. Finally, the digital single-tone signal is converted to the analog domain by the 14-bit DAC.

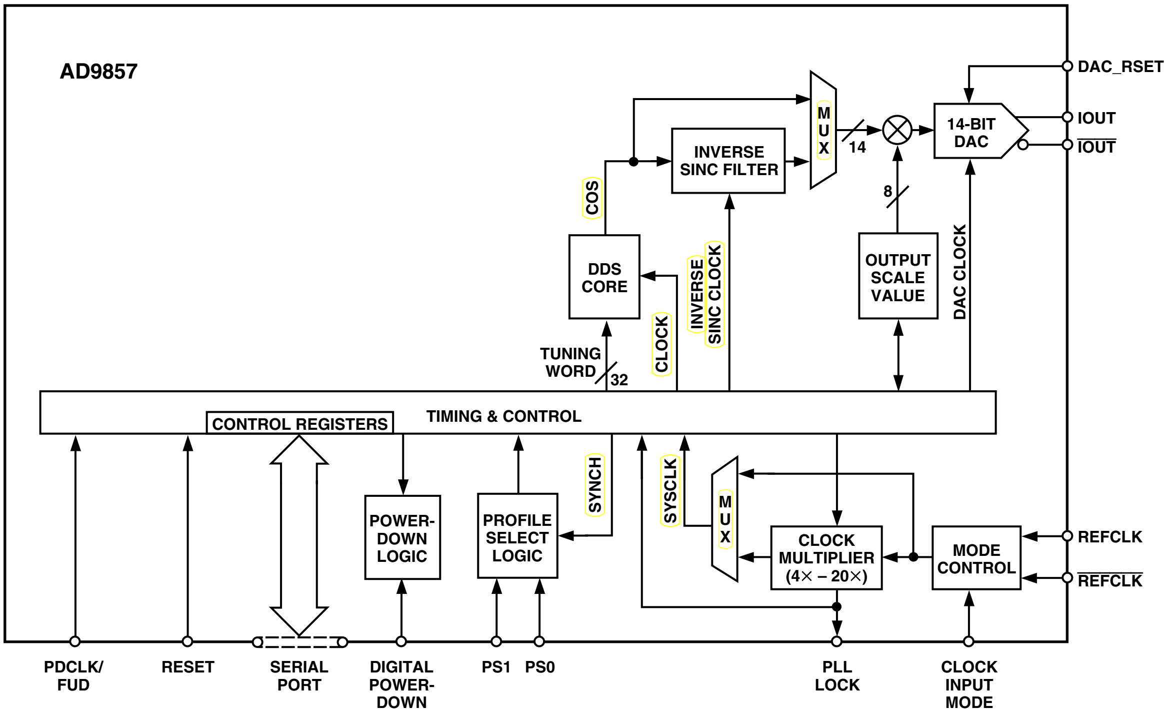


Figure 2. Single-Tone Mode

Inverse CIC Filter

The Inverse CIC (Cascaded Integrator Comb) Filter precompensates the data to offset the slight attenuation gradient imposed by the CIC Filter (see the Programmable (2×– 63×) CIC Interpolating Filter section). The I (or Q) data entering the first half-band filter occupies a maximum bandwidth of one-half f_{DATA} as defined by Nyquist (where f_{DATA} is the sample rate at the input of the first half-band filter). This is shown graphically in Figure 6.

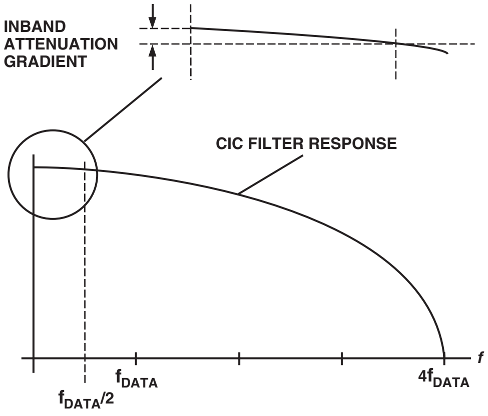


Figure 6. CIC Filter Response

If the CIC Filter is employed, the inband attenuation gradient could pose a problem for those applications requiring an extremely flat pass band. For example, if the spectrum of the data as supplied to the AD9857 I or Q path occupies a significant portion of the one-half f_{DATA} region, the higher frequencies of the data spectrum will receive slightly more attenuation than the lower frequencies (the worst-case overall droop from f = 0 to one-half f_{DATA} is < 0.8 dB). This may not be acceptable in certain applications. The Inverse CIC Filter has a response characteristic that is the inverse of the CIC Filter response over the one-half f_{DATA} region.

The net result is that the product of the two responses yields in an extremely flat pass band, thereby eliminating the inband attenuation gradient introduced by the CIC Filter. The price to be paid is a slight attenuation of the input signal of approximately 0.5 dB for a CIC interpolation rate of 2 and 0.8 dB for interpolation rates of 3 to 63.

The Inverse CIC Filter is implemented as a digital FIR Filter with a response characteristic that is the inverse of the Programmable CIC Interpolator. The product of the two responses yields a nearly flat response over the baseband Nyquist bandwidth. The Inverse CIC Filter provides frequency compensation that yields a response flatness of ±0.05 dB over the baseband Nyquist bandwidth, allowing the AD9857 to provide excellent SNR over its performance range.

The Inverse CIC Filter can be bypassed by setting Control Register 06h<0>. It is automatically bypassed if the CIC interpolation rate is 1×. Whenever this stage is bypassed, power to the stage is shut off, thereby reducing power dissipation.

Fixed Interpolator (4×)

This block is a fixed 4× interpolator. It is implemented as two half-band filters. The output of this stage is the original data upsampled by 4×.

Before presenting a detailed description of the half-band filters, recall that in the case of the Quadrature Modulation Mode the input data stream is representative of complex data; i.e., two input samples are required to produce one I/Q data pair. The I/Q sample rate is one-half the input data rate. The I/Q sample rate (the rate at which I or Q samples are presented to the input of the first half-band filter) will be referred to as f_{IQ}. Since the AD9857 is a quadrature modulator, f_{IQ} represents the baseband of the internal I/Q sample pairs. It should be emphasized here that f_{IQ} is not the same as the baseband of the user's symbol rate data, which must be upsampled before presentation to the AD9857 (as will be explained later). The I/Q sample rate (f_{IQ}) puts a limit on the minimum bandwidth necessary to transmit the f_{IQ} spectrum. This is the familiar Nyquist limit and is equal to one-half f_{IQ}, hereafter referred to as f_{NYQ}.

Together, the two half-band filters provide a factor-of-four increase in the sampling rate ($4 \times f_{IQ}$ or $8 \times f_{NYQ}$). Their combined insertion loss is 0.01 dB, so virtually no loss of signal level occurs through the two half-band filters. Both half-band filters are linear phase filters, so that virtually no phase distortion is introduced within the pass band of the filters. This is an important feature as phase distortion is generally intolerable in a data transmission system.

The half-band filters are designed so that their composite performance yields a usable pass band of 80% of the baseband Nyquist frequency (0.2 on the frequency scale below). Within that pass band, the ripple will not exceed 0.002 dB. The stopband extends from 120% to 400% of the baseband Nyquist frequency (0.3 to 1.0 on the frequency scale below) and offers a minimum of 85 dB attenuation. The composite response of the two half-band filters together are shown in Figures 7 and 8.

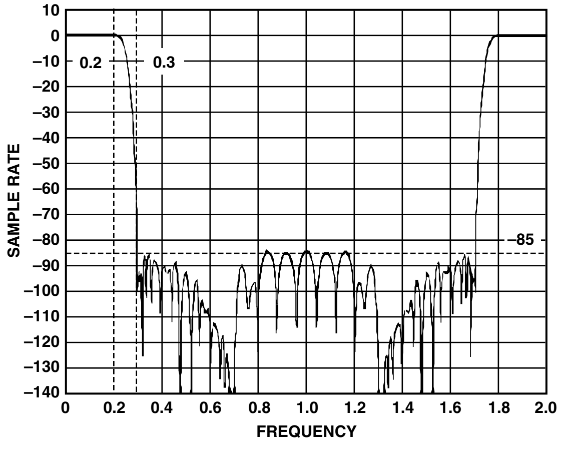


Figure 7. Half-Band 1 and 2 Frequency Response; Frequency Relative to HB1 Output Sample Rate

Quadrature Modulator

The digital quadrature modulator stage is used to frequency shift the *baseband* spectrum of the incoming data stream up to the desired carrier frequency (this process is known as *upconversion*).

At this point the incoming data has been converted from an incoming sampling rate of f_{IN} to an I/Q sampling rate equal to SYSCLK. The purpose of the upsampling process is to make the data sampling rate equal to the sampling rate of the carrier signal.

The carrier frequency is controlled numerically by a Direct Digital Synthesizer (DDS). The DDS uses the internal reference clock (SYSCLK) to generate the desired carrier frequency with a high degree of precision. The carrier is applied to the I and Q multipliers in quadrature fashion (90° phase offset) and summed to yield a data stream that represents the *quadrature modulated carrier.*

The modulation is done digitally which eliminates the phase and gain imbalance and crosstalk issues typically associated with analog modulators. Note that the modulated "signal" is actually a number stream sampled at the rate of SYSCLK, the same rate at which the output D/A converter is clocked.

The quadrature modulator operation is also controlled by spectral invert bits in each of the four profiles. The quadrature modulation takes the form:

$$I \times \cos(\omega) + Q \times \sin(\omega)$$

when the spectral invert bit is set to a Logic 1.

$$I \times \cos(\omega) - Q \times \sin(\omega)$$

when the spectral invert bit is set to a Logic 0.

DDS Core

The Direct Digital Synthesizer (DDS) block generates the sin/cos carrier reference signals that digitally modulate the I/Q data paths. The DDS frequency is tuned via the serial control port with a 32-bit tuning word (per profile). This allows the AD9857's output carrier frequency to be very precisely tuned while still providing output frequency agility.

The equation relating output frequency (f_{OUT}) of the AD9857 digital modulator to the frequency tuning word (FTWORD) and the system clock (SYSCLK) is:

$$f_{OUT} = (FTWORD \times SYSCLK)/2^{32} \qquad (2)$$

where f_{OUT} and $SYSCLK$ frequencies are in Hz and $FTWORD$ is a decimal number from 0 to 2,147,483,647 (2^{31}–1)

Example: Find the FTWORD for f_{OUT} = 41 MHz and SYSCLK = 122.88 MHz

If f_{OUT} = 41 MHz and SYSCLK = 122.88 MHz, then

$$FTWORD = 556AAAAB\ hex \qquad (3)$$

Loading 556AAAABh into control bus registers 08h–0Bh (for Profile 1) programs the AD9857 for f_{OUT} = 41 MHz, given a SYSCLK frequency of 122.88 MHz.

Inverse SINC Filter

The sampled carrier data stream is the input to the digital-to-analog converter (DAC) integrated onto the AD9857. The DAC output spectrum is shaped by the characteristic sin(x)/x (or SINC) envelope, due to the intrinsic zero-order hold effect associated with DAC-generated signals. Since the shape of the SINC envelope is well known, it can be compensated for. This envelope restoration function is provided by the optional Inverse SINC Filter preceding the DAC. This function is implemented as an FIR Filter, which has a transfer function that is the exact inverse of the SINC response. When the Inverse SINC Filter is selected, it modifies the incoming data stream so that the desired carrier envelope, which would otherwise be shaped by the SINC envelope, is restored. However, this correction is only complete for carrier frequencies up to approximately 45% of SYSCLK.

Note also that the Inverse SINC Filter introduces about a 3.5 dB loss at low frequencies as compared to the gain with the Inverse SINC Filter turned off. This is done to flatten the overall gain from dc to 45% of SYSCLK.

The Inverse SINC Filter can be bypassed if it is not needed. If the Inverse SINC Filter is bypassed, its clock is stopped, thus reducing the power dissipation of the part.

Output Scale Multiplier

An 8-bit multiplier (*Output Scale Value* in the block diagram) preceding the DAC provides the user with a means of adjusting the final output level. The multiplier value is programmed via the appropriate control registers, per each profile. The LSB weight is 2^{-7}, which yields a multiplier range of 0 to 1.9921875, or nearly 2×. Since the quadrature modulator has an intrinsic loss of 3 dB ($1/\sqrt{2}$), programming the multiplier for a value of $\sqrt{2}$) will restore the data to the full-scale range of the DAC when the device is operating in the Quadrature Modulation Mode. Since the AD9857 defaults to the Modulation mode, the default value for the multiplier is B5h (which corresponds to $\sqrt{2}$).

Programming the output scale multiplier to unity gain (80h) bypasses the stage, reducing power dissipation.

14-Bit D/A Converter

A 14-bit digital-to-analog converter (DAC) is used to convert the digitally processed waveform into an analog signal. The worst-case spurious signals due to the DAC are the harmonics of the fundamental signal and their aliases (please see Analog Devices, *DDS Tutorial* at www.analog.com/dds to request the tutorial containing a detailed explanation of aliases). The wideband 14-bit DAC in the AD9857 maintains spurious-free dynamic range (SFDR) performance of –60 dBc up to A_{OUT} = 42 MHz and –55 dBc up to A_{OUT} = 65 MHz.

The conversion process will produce aliased components of the fundamental signal at n × SYSCLK ± FCARRIER (n = 1, 2, 3). These are typically filtered with an external RLC filter at the DAC output. It is important for this analog filter to have a sufficiently flat gain and linear phase response across the bandwidth of interest to avoid modulation impairments.

The AD9857 provides true and complemented current outputs on A_{OUT} and $\overline{A_{OUT}}$ respectively. The full-scale output current is set by the RSET resistor at DAC_RSET. The value of RSET for a particular IOUT is determined using the following equation:

$$RSET = 39.93/IOUT \qquad (4)$$

For example, if a full-scale output current of 20 mA is desired, then *RSET* = (39.93/0.02), or approximately 2 kΩ. Every doubling of the RSET value will halve the output current.

The full-scale output current range of the AD9857 is 5 mA–20 mA. Full-scale output currents outside of this range will degrade SFDR performance. SFDR is also slightly affected by output matching; the two outputs should be terminated equally for best SFDR performance.

The output load should be located as close as possible to the AD9857 package to minimize stray capacitance and inductance. The load may be a simple resistor to ground, an op amp current-to-voltage converter, or a transformer-coupled circuit.

Driving an LC Filter without a transformer requires that the filter be doubly terminated for best performance. Therefore, the filter input and output should both be resistively terminated with the appropriate values. The parallel combination of the two terminations will determine the load that the AD9857 will see for signals within the filter pass band. For example, a 50 Ω terminated input/output low-pass filter will look like a 25 Ω load to the AD9857.

The output compliance voltage of the AD9857 is –0.5 V to +1.0 V. Any signal developed at the DAC output should not exceed 1.0 V, otherwise, signal distortion will result. Furthermore, the signal may extend below ground as much as 0.5 V without damage or signal distortion. The use of a transformer with a grounded center tap for common-mode rejection results in signals at the AD9857 DAC output pins that are symmetrical about ground.

As previously mentioned, by differentially combining the two signals, the user can provide some degree of common-mode signal rejection. A differential combiner might consist of a transformer or an op amp. The object is to combine or amplify only the difference between two signals and to reject any common, usually undesirable, characteristic, such as 60 Hz hum or "clock feed-through" that is equally present on both input signals. The AD9857 true and complement outputs can be differentially combined using a broadband 1:1 transformer with a grounded, center-tapped primary to perform differential combining of the two DAC outputs.

Reference Clock Multiplier

It is often difficult to provide a high-quality oscillator with an output in the frequency range of 100 MHz – 200 MHz. The AD9857 allows the use of a lower-frequency oscillator that can be multiplied to a higher frequency by the on-board Reference Clock Multiplier, implemented with a Phase Locked Loop architecture. See the Ease of Use section for a more thorough discussion of the Reference Clock Multiplier feature.

INPUT DATA PROGRAMMING

Control Interface—Serial I/O

The AD9857 serial port is a flexible, synchronous, serial communications port allowing easy interface to many industry-standard microcontrollers and microprocessors. The serial I/O is compatible with most synchronous transfer formats, including both the Motorola 6905/11 SPI and Intel 8051 SSR protocols.

The interface allows read/write access to all registers that configure the AD9857. Single or multiple byte transfers are supported as well as MSB first or LSB first transfer formats. The AD9857's serial interface port can be configured as a single pin I/O (SDIO) or two unidirectional pins for in/out (SDIO/SDO).

General Operation of the Serial Interface

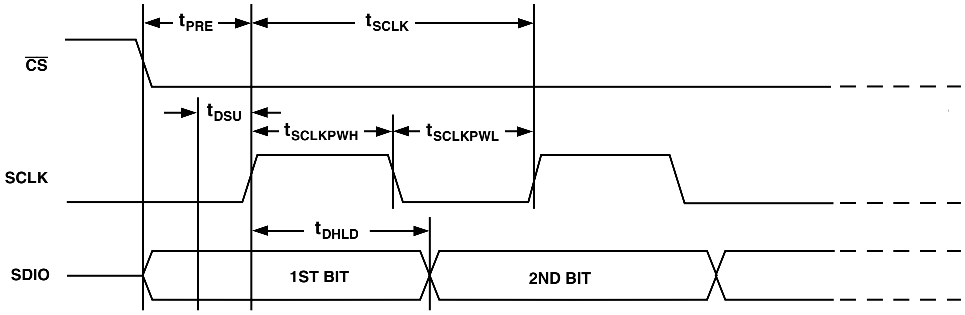
There are two phases to a communication cycle with the AD9857. Phase 1 is the instruction cycle, which is the writing of an instruction byte into the AD9857, coincident with the first eight SCLK rising edges. The instruction byte provides the AD9857 serial port controller with information regarding the data transfer cycle, which is Phase 2 of the communication cycle. The Phase 1 instruction byte defines whether the upcoming data transfer is read or write, the number of bytes in the data transfer (1–4), and the starting register address for the first byte of the data transfer.

The first eight SCLK rising edges of each communication cycle are used to write the instruction byte into the AD9857. The remaining SCLK edges are for Phase 2 of the communication cycle. Phase 2 is the actual data transfer between the AD9857 and the system controller. Phase 2 of the communication cycle is a transfer of 1, 2, 3, or 4 data bytes as determined by the instruction byte. Normally, using one communication cycle in a multibyte transfer is the preferred method. However, single byte communication cycles are useful to reduce CPU overhead when register access requires one byte only. An example of this may be to write the AD9857 SLEEP bit.

At the completion of any communication cycle, the AD9857 serial port controller expects the next eight rising SCLK edges to be the instruction byte of the next communication cycle.

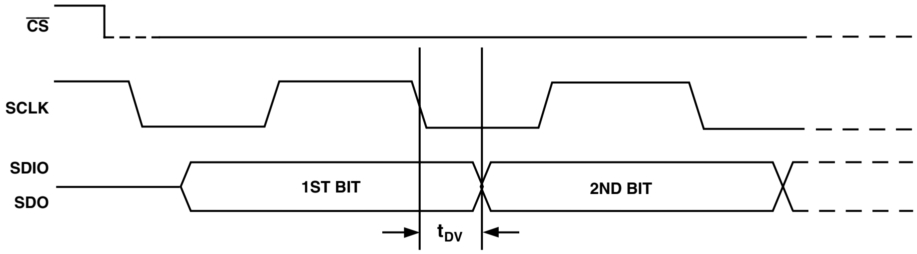
All data input to the AD9857 is registered on the rising edge of SCLK. All data is driven out of the AD9857 on the falling edge of SCLK.

Figures 10 and 11 illustrate the Data Write and Data Read operations on the AD9857 Serial Port.



SYMBOL	DEFINITION	MIN
t_{PRE}	$\overline{CS}$ SETUP TIME	40ns
t_{SCLK}	PERIOD OF SERIAL DATA CLOCK	100ns
t_{DSU}	SERIAL DATA SETUP TIME	30ns
$t_{SCLKPWH}$	SERIAL DATA CLOCK PULSEWIDTH HIGH	40ns
$t_{SCLKPWL}$	SERIAL DATA CLOCK PULSEWIDTH LOW	40ns
t_{DHLD}	SERIAL DATA HOLD TIME	0ns

Figure 10. Timing Diagram for Data Write to AD9857



SYMBOL	DEFINITION	MAX
t_{DV}	DATA VALID TIME	30ns

Figure 11. Timing Diagram for Data Read from AD9857

Figures 12–15 are useful in understanding the general operation of the AD9857 Serial Port.

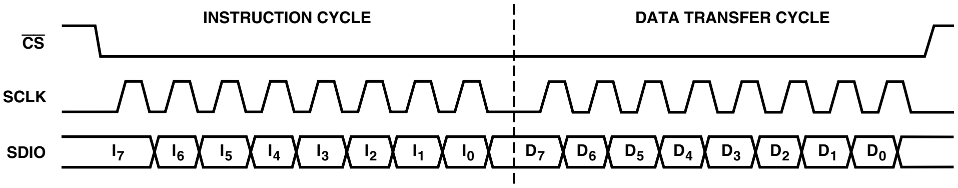

INSTRUCTION CYCLE
DATA TRANSFER CYCLE
$\overline{CS}$
SCLK
SDIO
I_7 I_6 I_5 I_4 I_3 I_2 I_1 I_0
D_7 D_6 D_5 D_4 D_3 D_2 D_1 D_0

Figure 12. Serial Port Writing Timing—Clock Stall Low

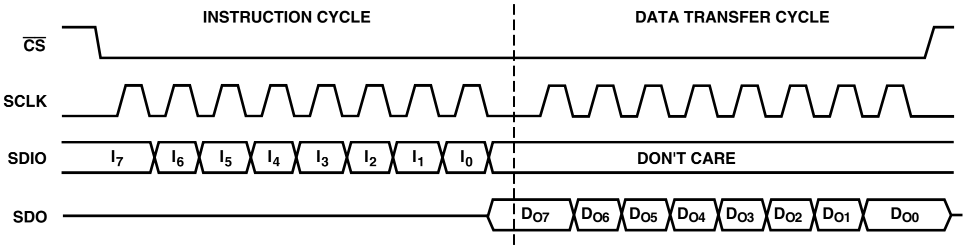

INSTRUCTION CYCLE
DATA TRANSFER CYCLE
$\overline{CS}$
SCLK
SDIO
I_7 I_6 I_5 I_4 I_3 I_2 I_1 I_0
DON'T CARE
SDO
D_{O7} D_{O6} D_{O5} D_{O4} D_{O3} D_{O2} D_{O1} D_{O0}

Figure 13. 3-Wire Serial Port Read Timing—Clock Stall Low

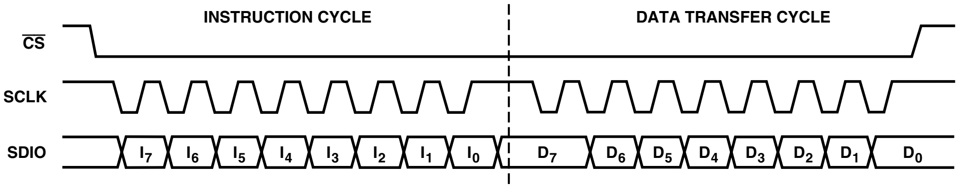

INSTRUCTION CYCLE
DATA TRANSFER CYCLE
$\overline{CS}$
SCLK
SDIO
I_7 I_6 I_5 I_4 I_3 I_2 I_1 I_0
D_7 D_6 D_5 D_4 D_3 D_2 D_1 D_0

Figure 14. Serial Port Write Timing—Clock Stall High

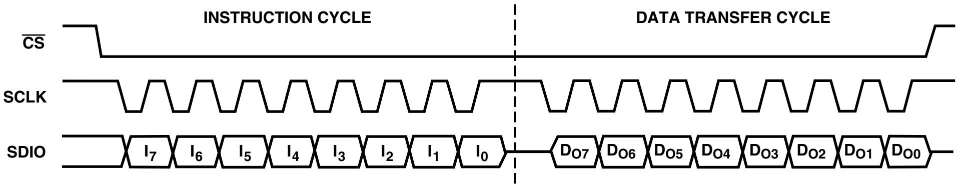

INSTRUCTION CYCLE
DATA TRANSFER CYCLE
$\overline{CS}$
SCLK
SDIO
I_7 I_6 I_5 I_4 I_3 I_2 I_1 I_0
D_{O7} D_{O6} D_{O5} D_{O4} D_{O3} D_{O2} D_{O1} D_{O0}

Figure 15. 2-Wire Serial Port Read Timing—Clock Stall High

Instruction Byte

The instruction byte contains the following information as shown in Table II.

Table II. Instruction Byte Information

MSB	D6	D5	D4	D3	D2	D1	LSB
R/$\overline{W}$	N1	N0	A4	A3	A2	A1	A0

R/$\overline{W}$—Bit 7 of the instruction byte determines whether a read or write data transfer will occur after the instruction byte write. Logic high indicates a read operation. Logic 0 indicates a write operation.

N1, N0—Bits 6 and 5 of the instruction byte determine the number of bytes to be transferred during the data transfer cycle of the communications cycle. The bit decodes are shown in Table III.

Table III. N1, N0 Decode Bits

N1	N0	Description
0	0	Transfer 1 Byte
0	1	Transfer 2 Bytes
1	0	Transfer 3 Bytes
1	1	Transfer 4 Bytes

A4, A3, A2, A1, A0—Bits 4, 3, 2, 1, and 0 of the instruction byte determine which register is accessed during the data transfer portion of the communications cycle. For multibyte transfers, this address is the starting byte address. The remaining register addresses are generated by the AD9857.

SERIAL INTERFACE PORT PIN DESCRIPTIONS

SCLK—Serial Clock. The serial clock pin is used to synchronize data to and from the AD9857 and to run the internal state machines. SCLK maximum frequency is 10 MHz. we use 1MHz

$\overline{CS}$—Chip Select. Active low input that allows more than one device on the same serial communications lines. The SDO and SDIO pins will go to a high-impedance state when this input is high. If driven high during any communications cycle, that cycle is suspended until $\overline{CS}$ is reactivated low. Chip Select can be tied low in systems that maintain control of SCLK.

SDIO—Serial Data I/O. Data is always written into the AD9857 on this pin. However, this pin can be used as a bidirectional data line. The configuration of this pin is controlled by Bit 7 of register address 00h. The default is logic zero, which configures the SDIO pin as bidirectional. we will use unidirectional write

SDO—Serial Data Out. Data is read from this pin for protocols that use separate lines for transmitting and receiving data. When the AD9857 operates in a single bidirectional I/O mode, this pin does not output data and is set to a high-impedance state.

SYNCIO—Synchronizes the I/O port state machines without affecting the addressable registers contents. An active high input on the SYNC I/O pin causes the current communication cycle to abort. After SYNC I/O returns low (Logic 0) another communication cycle may begin, starting with the instruction byte write.

MSB/LSB Transfers

The AD9857 Serial Port can support both most significant bit (MSB) first or least significant bit (LSB) first data formats. This functionality is controlled by the Control Register 00h<6> bit. The default value of Control Register 00h<6> is low (MSB first). When Control Register 00h<6> is set high, the AD9857 serial port is in LSB first format. The instruction byte must be written in the format indicated by Control Register 00h<6>. That is, if the AD9857 is in LSB first mode, the instruction byte must be written from least significant bit to most significant bit.

Multibyte data transfers in MSB format can be completed by writing an instruction byte that includes the register address of the most significant byte. In MSB first mode, the serial port internal byte address generator decrements for each byte required of the multibyte communication cycle. Multibyte data transfers in LSB first format can be completed by writing an instruction byte that includes the register address of the least significant byte. In LSB First mode, the serial port internal byte address generator increments for each byte required of the multibyte communication cycle.

Notes on Serial Port Operation

The AD9857 serial port configuration bits reside in Bits 6 and 7 of register address 0h. It is important to note that the configuration changes immediately upon writing to this register. For multibyte transfers, writing to this register may occur during the middle of a communication cycle. Care must be taken to compensate for this new configuration for the remainder of the current communication cycle.

The AD9857 serial port controller address will roll from 19h to 0h for multibyte I/O operations if the MSB first mode is active. The serial port controller address will roll from 0h to 19h for multibyte I/O operations if the LSB first mode is active.

The system must maintain synchronization with the AD9857 or the internal control logic will not be able to recognize further instructions. For example, if the system sends an instruction byte for a 2-byte write, then pulses the SCLK pin for a 3-byte write (8 additional SCLK rising edges), communication synchronization is lost. In this case, the first 16 SCLK rising edges after the instruction cycle will properly write the first two data bytes into the AD9857, but the next eight rising SCLK edges are interpreted as the next instruction byte, not the final byte of the previous communication cycle.

When synchronization is lost between the system and the AD9857, the SYNC I/O pin provides a means to re-establish synchronization without reinitializing the entire chip. The SYNC I/O pin enables the user to reset the AD9857 state machine to accept the next eight SCLK rising edges to be coincident with the instruction phase of a new communication cycle. By applying and removing a "high" signal to the SYNC I/O pin, the AD9857 is set to once again begin performing the communication cycle in synchronization with the system. Any information that had been written to the AD9857 registers during a valid communication cycle prior to loss of synchronization will remain intact.

CONTROL REGISTER DESCRIPTION

Reference Clock (REFCLK) Multiplier—Register Address 00h, Bits 0, 1, 2, 3, 4 User variable #1

A 5-bit number (M), the value of which determines the multiplication factor for the internal PLL (Bit 4 is the MSB). The system clock (SYSCLK) is M times the frequency of the REFCLK input signal. If M = 01h, the PLL circuit is bypassed and $f_{SYSCLK} = f_{REFCLK}$. If 04h ≤ M ≤ 14h, the PLL multiplies the REFCLK frequency by M (4–20 decimal). Any other value of M is considered an invalid entry.

PLL Lock Control—Register Address 00h, Bit 5

When set to a Logic 0, the device uses the status of the PLL Lock Indicator pin to internally control the operation of the 14-bit parallel data path. When set to a Logic 1, the internal control logic ignores the status of the PLL Lock Indicator pin.

LSB First—Register Address 00h, Bit 6

When set to a Logic 1, the serial interface accepts serial data in LSB first format. When set to a Logic 0, MSB first format is assumed.

SDIO Input Only—Register Address 00h, Bit 7

When set to a Logic 1, the serial data I/O pin (SDIO) is configured as an input only pin. When set to a Logic 0, the SDIO pin has bidirectional operation.

Operating Mode—Register Address 01h, Bits 0, 1

00h: Selects the Quadrature Modulation Mode of operation.
01h: Selects the Single-Tone Mode of operation.
02h: Selects the Interpolating DAC Mode of operation.
03h: Invalid entry.

Auto Power-Down—Register Address 01h, Bit 2

When set to a Logic 1, the device automatically switches into its low-power mode whenever TxENABLE is deasserted for a sufficiently long period of time. When set to a Logic 0, the device only powers down in response to the Digital Power-Down pin.

Full Sleep Mode—Register Address 01h, Bit 3 logic 0

When set to a Logic 1, the device completely shuts down.

Reserved—Register Address 01h, Bit 4

Reserved—Register Address 01h, Bit 5

This bit must always be set to 0.

Inverse SINC Bypass—Register Address 01h, Bit 6

When set to a Logic 1, the Inverse Sinc Filter is BYPASSED.
When set to a Logic 0, the Inverse Sinc Filter is active.

CIC Clear—Register Address 01h, Bit 7

When set to a Logic 1, the CIC Filters are cleared. When set to a Logic 0, the CIC Filters operate normally.

PROFILE #0 user variable #2

Tuning Word—Register Address 02h, Bits 0, 1, 2, 3, 4, 5, 6, 7

The lower byte of the 32-bit frequency tuning word, Bits 0–7.

Tuning Word—Register Address 03h, Bits 0, 1, 2, 3, 4, 5, 6, 7

The second byte of the 32-bit frequency tuning word, Bits 8–15.

Tuning Word—Register Address 04h, Bits 0,1, 2, 3, 4, 5, 6, 7

The third byte of the 32-bit frequency tuning word, Bits 16–23.

Tuning Word—Register Address 05h, Bits 0, 1, 2, 3, 4, 5, 6, 7

The fourth byte of the 32-bit frequency tuning word, Bits 24–31.

Inverse CIC Bypass—Register Address 06h, Bit 0

When set to a Logic 1, the Inverse CIC Filter is BYPASSED.
When set to a Logic 0, the Inverse CIC Filter is active.

Spectral Invert—Register Address 06h, Bit 1

The quadrature modulator takes the form:
I × cos(ω) + Q × sin(ω) when set to a Logic 1.
I × cos(ω) – Q × sin(ω) when set to a Logic 0.

CIC Interpolation Rate—Register Address 06h, Bits 2, 3, 4, 5, 6, 7

00h: Invalid entry.
01h: CIC Filters BYPASSED.
02h–3Fh: CIC interpolation rate (2–63, decimal).

Output Scale Factor—Register Address 07h, Bits 0, 1, 2, 3, 4, 5, 6, 7 user variable #3

An 8-bit number that serves as a multiplier for the data pathway before the data is delivered the DAC. It has an LSB weight of 2^{-7} (0.0078125). This yields a multiplier range of 0 to 1.9921875.

PROFILE #1

Tuning Word—Register Address 08h, Bits 0, 1, 2, 3, 4, 5, 6, 7

The lower byte of the 32-bit frequency tuning word, Bits 0–7.

Tuning Word—Register Address 09h, Bits 0, 1, 2, 3, 4, 5, 6, 7

The second byte of the 32-bit frequency tuning word, Bits 8–15.

Tuning Word—Register Address 0Ah, Bits 0, 1, 2, 3, 4, 5, 6, 7

The third byte of the 32-bit frequency tuning word, Bits 16–23.

Tuning Word—Register Address 0Bh, Bits 0, 1, 2, 3, 4, 5, 6, 7

The fourth byte of the 32-bit frequency tuning word, Bits 24–31.

Inverse CIC Bypass—Register Address 0Ch, Bit 0

When set to a Logic 1, the Inverse CIC Filter is BYPASSED.
When set to a Logic 0, the Inverse CIC Filter is active.

Spectral Invert—Register Address 0Ch, Bit 1

The quadrature modulator takes the form:
I × cos(ω) + Q × sin(ω) when set to a Logic 1.
I × cos(ω) + Q × sin(ω) when set to a Logic 0.

CIC Interpolation Rate—Register Address 0Ch, Bits 2, 3, 4, 5, 6, 7

00h: Invalid entry.
01h: CIC Filters BYPASSED.
02h–3Fh: CIC interpolation rate (2–63, decimal).

Output Scale Factor—Register Address 0Dh, Bits 0, 1, 2, 3, 4, 5, 6, 7

An 8-bit number that serves as a multiplier for the data pathway before the data is delivered the DAC. It has an LSB weight of 2^{-7} (0.0078125). This yields a multiplier range of 0 to 1.9921875.

PROFILE #2

Tuning Word—Register Address 0Eh, Bits 0, 1, 2, 3, 4, 5, 6, 7

The lower byte of the 32-bit frequency tuning word, Bits 0–7.

Tuning Word—Register Address 0Fh, Bits 0, 1, 2, 3, 4, 5, 6, 7

The second byte of the 32-bit frequency tuning word, Bits 8–15.

Tuning Word—Register Address 10h, Bits 0, 1, 2, 3, 4, 5, 6, 7

The third byte of the 32-bit frequency tuning word, Bits 16–23.

Tuning Word—Register Address 11h, Bits 0, 1, 2, 3, 4, 5, 6, 7

The fourth byte of the 32-bit frequency tuning word, Bits 24–31.

Inverse CIC Bypass—Register Address 12h, Bit 0
When set to a Logic 1, the Inverse CIC Filter is BYPASSED. When set to a Logic 0, the Inverse CIC Filter is active.

Spectral Invert—Register Address 12h, Bit 1
The quadrature modulator takes the form:
$I \times \cos(\omega) + Q \times \sin(\omega)$ when set to a Logic 1.
$I \times \cos(\omega) + Q \times \sin(\omega)$ when set to a Logic 0.

CIC Interpolation Rate—Register Address 12h, Bits 2, 3, 4, 5, 6, 7
00h: Invalid entry.
01h: CIC Filters BYPASSED.
02h–3Fh: CIC interpolation rate (2–63, decimal).

Output Scale Factor—Register Address 13h, Bits 0, 1, 2, 3, 4, 5, 6, 7
An 8-bit number that serves as a multiplier for the data pathway before the data is delivered the DAC. It has an LSB weight of 2^{-7} (0.0078125). This yields a multiplier range of 0 to 1.9921875.

PROFILE #3
Tuning Word—Register Address 14h, Bits 0, 1, 2, 3, 4, 5, 6, 7
The lower byte of the 32-bit frequency tuning word, Bits 0–7.

Tuning Word—Register Address 15h, Bits 0, 1, 2, 3, 4, 5, 6, 7
The second byte of the 32-bit frequency tuning word, Bits 8–15.

Tuning Word—Register Address 16h, Bits 0, 1, 2, 3, 4, 5, 6, 7
The third byte of the 32-bit frequency tuning word, Bits 16–23.

Tuning Word—Register Address 17h, Bits 0, 1, 2, 3, 4, 5, 6, 7
The fourth byte of the 32-bit frequency tuning word, Bits 24–31.

Inverse CIC Bypass—Register Address 18h, Bit 0
When set to a Logic 1, the Inverse CIC Filter is BYPASSED. When set to a Logic 0, the Inverse CIC Filter is active.

Spectral Invert—Register Address 18h, Bit 1
The quadrature modulator takes the form:
$I \times \cos(\omega) + Q \times \sin(\omega)$ when set to a Logic 1.
$I \times \cos(\omega) + Q \times \sin(\omega)$ when set to a Logic 0.

CIC Interpolation Rate—Register Address 18h, Bits 2, 3, 4, 5, 6, 7
00h: Invalid entry.
01h: CIC Filters BYPASSED.
02h–3Fh: CIC interpolation rate (2–63, decimal).

Output Scale Factor—Register Address 19h, Bits 0, 1, 2, 3, 4, 5, 6, 7
An 8-bit number that serves as a multiplier for the data pathway before the data is delivered the DAC. It has an LSB weight of 2^{-7} (0.0078125). This yields a multiplier range of 0 to 1.9921875.

Table IV. Control Register Quick Reference

Reg Address	(MSB) Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	(LSB) Bit 0	Def. Value	Profile
00h	SDIO Input Only	LSB First	PLL Lock Control	**REFCLK Multiplier** 01h: *Bypass* PLL 04h–14h: 4×–20×					21h	N/A
01h	CIC Clear	Inverse SINC Bypass	Reserved: Must Be 0	Reserved	Full Sleep	Auto Power-Down	**Operating Mode** 00h: Quad. Mod. 01h: Single-Tone 02h: Intrp. DAC		00h	N/A
02h	Frequency Tuning Word #1 <7:0>								00h	0
03h	Frequency Tuning Word #1 <15:8>								00h	0
04h	Frequency Tuning Word #1 <23:16>								00h	0
05h	Frequency Tuning Word #1 <31:24>								00h	0
06h	**CIC Interpolation Rate** 01h: Bypass CIC Filter 02h–3Fh: Interpolation Factor (2–63, Decimal)						Spectral Invert	Inverse CIC Bypass	08h	0
07h	**Output Scale Factor** Bit Weighting: MSB = 2^0, LSB = 2^{-7}								B5h	0
08h	Frequency Tuning Word #2 <7:0>								Unset	1
09h	Frequency Tuning Word #2 <15:8>								Unset	1
0Ah	Frequency Tuning Word #2 <23:16>								Unset	1
0Bh	Frequency Tuning Word #2 <31:24>								Unset	1
0Ch	**CIC Interpolation Rate** 01h: Bypass CIC Filter 02h–3Fh: Interpolation Factor (2–63, Decimal)						Spectral Invert	Inverse CIC Bypass	Unset	1
0Dh	**Output Scale Factor** Bit Weighting: MSB = 2^0, LSB = 2^{-7}								Unset	1
0Eh	Frequency Tuning Word #3 <7:0>								Unset	2
0Fh	Frequency Tuning Word #3 <15:8>								Unset	2
10h	Frequency Tuning Word #3 <23:16>								Unset	2
11h	Frequency Tuning Word #3 <31:24>								Unset	2
12h	**CIC Interpolation Rate** 01h: Bypass CIC Filter 02h–3Fh: Interpolation Factor (2–63, Decimal)						Spectral Invert	Inverse CIC Bypass	Unset	2
13h	**Output Scale Factor** Bit Weighting: MSB = 2^0, LSB = 2^{-7}								Unset	2
14h	Frequency Tuning Word #4 <7:0>								Unset	3
15h	Frequency Tuning Word #4 <15:8>								Unset	3
16h	Frequency Tuning Word #4 <23:16>								Unset	3
17h	Frequency Tuning Word #4 <31:24>								Unset	3
18h	**CIC Interpolation Rate** 01h: Bypass CIC Filter 02h–3Fh: Interpolation Factor (2–63, Decimal)						Spectral Invert	Inverse CIC Bypass	Unset	3
19h	**Output Scale Factor** Bit Weighting: MSB = 2^0, LSB = 2^{-7}								Unset	3