TABLE OF CONTENTS

REVISION HISTORY

5/14—Rev. 0 to Rev. A

Change to Table 25 51
Changes to Table 103..... 69

12/13—Revision 0: Initial Version

FUNCTIONAL BLOCK DIAGRAM

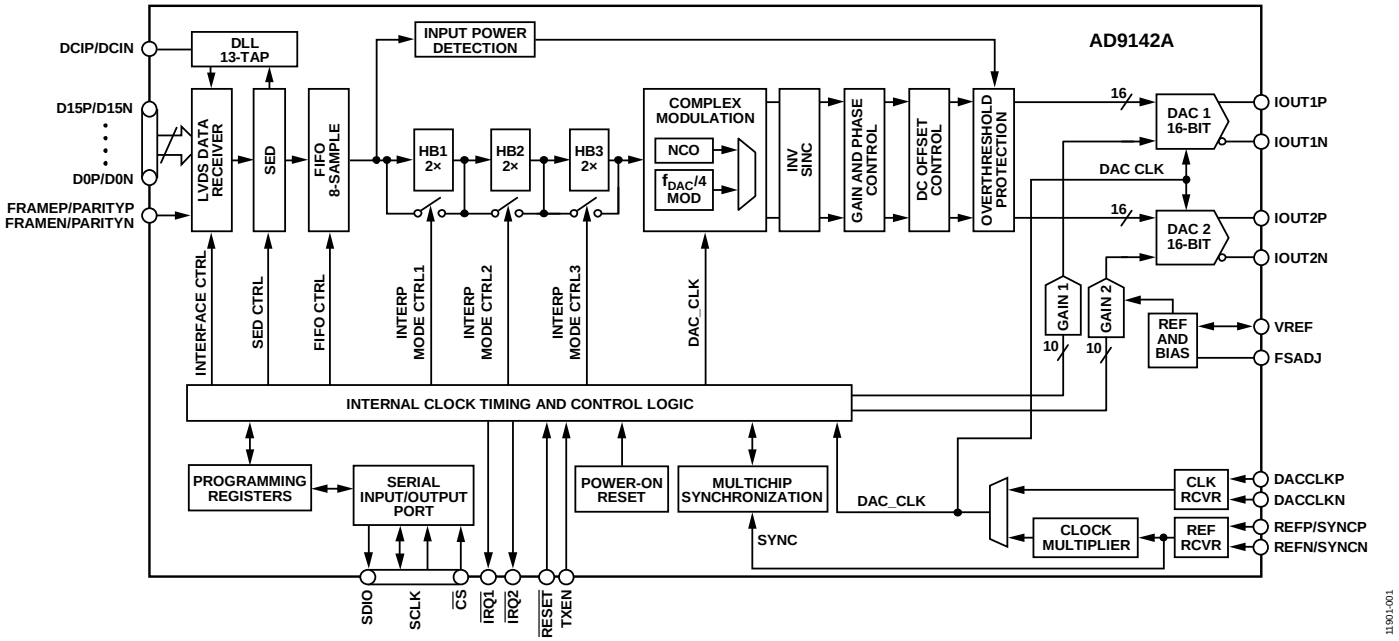


Figure 1.

SPECIFICATIONS

DC SPECIFICATIONS

T_{MIN} to T_{MAX}, AVDD33 = 3.3 V, DVDD18 = 1.8 V, CVDD18 = 1.8 V, I_{OUTFS} = 20 mA, maximum sample rate, unless otherwise noted.

Table 1.

Parameter	Test Conditions/Comments	Min	Typ	Max	Unit
RESOLUTION			16		Bits
ACCURACY					
Differential Nonlinearity (DNL)			±2.1		LSB
Integral Nonlinearity (INL)			±3.7		LSB
MAIN DAC OUTPUTS					
Offset Error		−0.001	0	+0.001	% FSR
Gain Error	With internal reference	−3.2	+2	+4.7	% FSR
Full-Scale Output Current	Based on a 10 kΩ external resistor between FSADJ and AVSS	19.06	19.8	20.6	mA
Output Compliance Range		−1.0		+1.0	V
Output Resistance			10		MΩ
Gain DAC Monotonicity			Guaranteed		
Settling Time to Within ±0.5 LSB			20		ns
MAIN DAC TEMPERATURE DRIFT					
Offset			0.04		ppm/°C
Gain			100		ppm/°C
Reference Voltage			30		ppm/°C
REFERENCE					
Internal Reference Voltage		1.17		1.19	V
Output Resistance			5		kΩ
ANALOG SUPPLY VOLTAGES					
AVDD33		3.13	3.3	3.47	V
CVDD18		1.7	1.8	1.9	V
DIGITAL SUPPLY VOLTAGES					
DVDD18		1.7	1.8	1.9	V
DVDD18 Variation over Operating Conditions[1]		−2.5%		+2.5%	V
POWER CONSUMPTION					
2× Mode	f_{DAC} = 737.28 MSPS				
NCO OFF			925		mW
NCO ON			1217		mW
2× Mode	f_{DAC} = 983.04 MSPS				
NCO OFF			1135		mW
NCO ON			1520		mW
4× Mode	f_{DAC} = 737.28 MSPS				
NCO OFF			852		mW
NCO ON			1144		mW
4× Mode	f_{DAC} = 983.04 MSPS				
NCO OFF			1040		mW
NCO ON			1425		mW
4× Mode	f_{DAC} = 1228.8 MSPS				
NCO OFF			1230		mW
NCO ON			1725		mW
4× Mode	f_{DAC} = 1474.56 MSPS				
NCO OFF			1405		mW
NCO ON			1990		mW

Parameter	Test Conditions/Comments	Min	Typ	Max	Unit
8× Mode	f_{DAC} = 1600 MSPS				
NCO OFF			1350		mW
NCO ON			1984		mW
Phase-Lock Loop (PLL)			70		mW
Inverse Sinc	f_{DAC} = 1474.56 MSPS		113		mW
Reduced Power Mode (Power-Down)				96.6	mW
AVDD33				1.5	mA
CVDD18				42.3	mA
DVDD18				8.6	mA
OPERATING RANGE		−40	+25	+85	°C

[1] This term specifies the maximum allowable variation of DVDD18 over operating conditions compared with the DVDD18 presented to the device at the time the data interface DLL is enabled.

DIGITAL SPECIFICATIONS

T_{MIN} to T_{MAX}, AVDD33 = 3.3 V, DVDD18 = 1.8 V, CVDD18 = 1.8 V, I_{OUTFS} = 20 mA, maximum sample rate, unless otherwise noted.

Table 2.

Parameter	Symbol	Test Conditions/Comments	Min	Typ	Max	Unit
CMOS INPUT LOGIC LEVEL						
Input						
Logic High		DVDD18 = 1.8 V	1.2			V
Logic Low		DVDD18 = 1.8 V			0.6	V
CMOS OUTPUT LOGIC LEVEL						
Output						
Logic High		DVDD18 = 1.8 V	1.4			V
Logic Low		DVDD18 = 1.8 V			0.4	V
LVDS RECEIVER INPUTS		Data, frame signal, and DCI inputs				
Input Voltage Range	V_{IA} or V_{IB}		825		1675	mV
Input Differential Threshold	V_{IDTH}		−175		+175	mV
Input Differential Hysteresis	V_{IDTHH} to V_{IDTHL}			20		mV
Receiver Differential Input Impedance	R_{IN}			100		Ω
DLL SPEED RANGE			250		575	MHz
DAC UPDATE RATE					1600	MSPS
DAC Adjusted Update Rate		2× interpolation			575	MSPS
DAC CLOCK INPUT (DACCLKP, DACCLKN)						
Differential Peak-to-Peak Voltage			100	500	2000	mV
Common-Mode Voltage		Self biased input, ac-coupled		1.25		V
REFCLK/SYNCCLK INPUT (REFP/SYNCP, REFN/SYNCN)						
Differential Peak-to-Peak Voltage			100	500	2000	mV
Common-Mode Voltage				1.25		V
Input Clock Frequency		1.03 GHz ≤ f_{VCO} ≤ 2.07 GHz			450	MHz
SERIAL PORT INTERFACE						
Maximum Clock Rate	SCLK		40			MHz
Minimum Pulse Width						
High	t_{PWH}				12.5	ns
Low	t_{PWL}				12.5	ns
SDIO to SCLK Setup Time	t_{DS}		1.5			ns
SDIO to SCLK Hold Time	t_{DH}		0.68			ns
$\overline{CS}$ to SCLK Setup Time	t_{DCSB}		2.38	1.4		ns
$\overline{CS}$ to SCLK Hold Time	t_{DCSB}		9.6			ns
SDIO to SCLK Delay	t_{DV}	Wait time for valid output from SDIO	11			ns
SDIO High-Z to $\overline{CS}$		Time for SDIO to relinquish the output bus	8.5			ns
SDIO LOGIC LEVEL						
Voltage Input High	V_{IH}		1.2	1.8		V
Voltage Input Low	V_{IL}			0	0.5	V
Voltage Output High	I_{IH}	With 2 mA loading	1.36		2	V
Voltage Output Low	I_{IL}	With 2 mA loading	0		0.45	V

DAC LATENCY SPECIFICATIONS

T_{MIN} to T_{MAX}, AVDD33 = 3.3 V, DVDD18 = 1.8 V, CVDD18 = 1.8 V, I_{OUTFS} = 20 mA, FIFO level is set to 4 (half of the FIFO depth), unless otherwise noted.

Table 3.

Parameter	Test Conditions/Comments	Min	Typ	Max	Unit
WORD INTERFACE MODE	Fine/coarse modulation, inverse sinc, gain/phase compensation off				
2× Interpolation			134		DACCLK cycles
4× Interpolation			244		DACCLK cycles
8× Interpolation			481		DACCLK cycles
BYTE INTERFACE MODE	Fine/coarse modulation, inverse sinc, gain/phase compensation off				
2× Interpolation			145		DACCLK cycles
4× Interpolation			271		DACCLK cycles
8× Interpolation			506		DACCLK cycles
INDIVIDUAL FUNCTION BLOCKS					
Modulation					
Fine			17		DACCLK cycles
Coarse			10		DACCLK cycles
Inverse Sinc			20		DACCLK cycles
Phase Compensation			12		DACCLK cycles
Gain Compensation			16		DACCLK cycles

LATENCY VARIATION SPECIFICATIONS

Table 4.

Parameter	Min	Typ	Max	Unit
DAC LATENCY VARIATION[1]				
SYNC Off		1	2	DACCLK cycles
SYNC On		0	1	DACCLK cycles

[1] DAC latency is defined as the elapsed time from a data sample clocked at the input to the AD9142A until the analog output begins to change.

AC SPECIFICATIONS

T_{MIN} to T_{MAX}, AVDD33 = 3.3 V, DVDD18 = 1.8 V, CVDD18 = 1.8 V, I_{OUTFS} = 20 mA, maximum sample rate, unless otherwise noted.

Table 5.

Parameter	Test Conditions/Comments	Min	Typ	Max	Unit
SPURIOUS-FREE DYNAMIC RANGE (SFDR)	−14 dBFS single tone				
f_{DAC} = 737.28 MSPS	f_{OUT} = 200 MHz				
BW = 125 MHz			85		dBc
BW = 270 MHz			80		dBc
f_{DAC} = 983.04 MSPS	f_{OUT} = 200 MHz				
BW = 360 MHz			85		dBc
f_{DAC} = 1228.8 MSPS	f_{OUT} = 280 MHz				
BW = 200 MHz			85		dBc
BW = 500 MHz			75		dBc
f_{DAC} = 1474.56 MSPS					
BW = 737 MHz	f_{OUT} = 10 MHz		85		dBc
BW = 400 MHz	f_{OUT} = 280 MHz		80		dBc
TWO-TONE INTERMODULATION DISTORTION (IMD)	−12 dBFS each tone				
f_{DAC} = 737.28 MSPS	f_{OUT} = 200 MHz		80		dBc
f_{DAC} = 983.04 MSPS	f_{OUT} = 200 MHz		82		dBc
f_{DAC} = 1228.8 MSPS	f_{OUT} = 280 MHz		80		dBc
f_{DAC} = 1474.56 MSPS	f_{OUT} = 10 MHz		85		dBc
	f_{OUT} = 280 MHz		79		dBc
NOISE SPECTRAL DENSITY (NSD)	Eight-tone, 500 kHz tone spacing				
f_{DAC} = 737.28 MSPS	f_{OUT} = 200 MHz		−160		dBm/Hz
f_{DAC} = 983.04 MSPS	f_{OUT} = 200 MHz		−161.5		dBm/Hz
f_{DAC} = 1228.8 MSPS	f_{OUT} = 280 MHz		−164.5		dBm/Hz
f_{DAC} = 1474.56 MSPS	f_{OUT} = 10 MHz		−166		dBm/Hz
	f_{OUT} = 280 MHz		−162.5		dBm/Hz
W-CDMA ADJACENT CHANNEL LEAKAGE RATIO (ACLR)	Single carrier				
f_{DAC} = 983.04 MSPS	f_{OUT} = 200 MHz		81		dBc
f_{DAC} = 1228.8 MSPS	f_{OUT} = 20 MHz		83		dBc
	f_{OUT} = 280 MHz		80		dBc
f_{DAC} = 1474.56 MSPS	f_{OUT} = 20 MHz		81		dBc
	f_{OUT} = 280 MHz		80		dBc
W-CDMA SECOND (ACLR)	Single carrier				
f_{DAC} = 983.04 MSPS	f_{OUT} = 200 MHz		85		dBc
f_{DAC} = 1228.8 MSPS	f_{OUT} = 20 MHz		86		dBc
	f_{OUT} = 280 MHz		86		dBc
f_{DAC} = 1474.56 MSPS	f_{OUT} = 20 MHz		86		dBc
	f_{OUT} = 280 MHz		85		dBc

OPERATING SPEED SPECIFICATIONS

Table 6.

	DVDD18, CVDD18 = 1.8 V ± 5%		DVDD18, CVDD18 = 1.9 V ± 5% or 1.8 V ± 2%		DVDD18, CVDD18 = 1.9 V ± 2%	
Interpolation Factor	f_{DCI} (MSPS) Maximum	f_{DAC} (MSPS) Maximum	f_{DCI} (MSPS) Maximum	f_{DAC} (MSPS) Maximum	f_{DCI} (MSPS) Maximum	f_{DAC} (MSPS) Maximum
2×	575	1150	575	1150	575	1150
4×	350	1400	375	1500	400	1600
8×	175	1400	187.5	1500	200	1600

ABSOLUTE MAXIMUM RATINGS

Table 7.

Parameter	Rating
AVDD33 to GND	−0.3 V to +3.6 V
DVDD18, CVDD18 to GND	−0.3 V to +2.1 V
FSADJ, VREF, IOUT1P, IOUT1N, IOUT2P, IOUT2N to GND	−0.3 V to AVDD33 + 0.3 V
D15P to D0P, D15N to D0N, FRAMEP/PARITYP, FRAMEN/PARITYN, DCIP, DCIN to GND	−0.3 V to DVDD18 + 0.3 V
DACCLKP, DACCLKN, REFP, SYNCP, REFN, SYNCN to GND	−0.3 V to CVDD18 + 0.3 V
$\overline{\text{RESET}}$, $\overline{\text{IRQ1}}$, $\overline{\text{IRQ2}}$, $\overline{\text{CS}}$, SCLK, SDIO to GND	−0.3 V to DVDD18 + 0.3 V
Junction Temperature	125°C
Storage Temperature Range	−65°C to +150°C

Stresses above those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is a stress rating only; functional operation of the device at these or any other conditions above those indicated in the operational section of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

THERMAL RESISTANCE

The exposed pad (EPAD) must be soldered to the ground plane (AVSS) for the 72-lead LFCSP. The EPAD provides an electrical, thermal, and mechanical connection to the board.

Typical θ_{JA}, θ_{JB}, and θ_{JC} values are specified for a 4-layer board in still air. Airflow increases heat dissipation, effectively reducing θ_{JA} and θ_{JB}.

Table 8. Thermal Resistance

Package	θ_{JA}	θ_{JB}	θ_{JC}	Unit	Conditions
72-Lead LFCSP	20.7	10.9	1.1	°C/W	EPAD soldered to ground plane

ESD CAUTION

ESD (electrostatic discharge) sensitive device. Charged devices and circuit boards can discharge without detection. Although this product features patented or proprietary protection circuitry, damage may occur on devices subjected to high energy ESD. Therefore, proper ESD precautions should be taken to avoid performance degradation or loss of functionality.

PIN CONFIGURATION AND FUNCTION DESCRIPTIONS

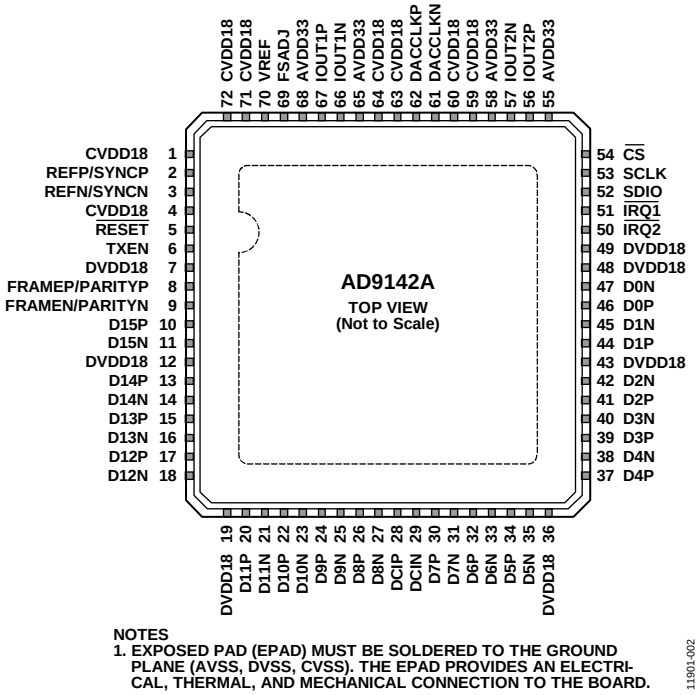


Figure 2. Pin Configuration

Table 9. Pin Function Descriptions

Pin No.	Mnemonic	Description
1	CVDD18	1.8 V PLL Supply. CVDD18 supplies the clock receivers, clock multiplier, and clock distribution.
2	REFP/SYNCP	PLL Reference Clock/Synchronization Clock Input, Positive.
3	REFN/SYNCN	PLL Reference Clock/Synchronization Clock Input, Negative.
4	CVDD18	1.8 V PLL Supply. CVDD18 supplies the clock receivers, clock multiplier, and clock distribution.
5	$\overline{\text{RESET}}$	Reset, Active Low. CMOS levels with respect to DVDD18. Recommended reset pulse length is 1 µs.
6	TXEN	Active High Transmit Path Enable. CMOS levels with respect to DVDD18. A low level on this pin triggers three selectable actions in the DAC. See Table 87 for details.
7	DVDD18	1.8 V Digital Supply. Pin 7 supplies power to the digital core, digital data ports, serial port input/output pins, $\overline{\text{RESET}}$, $\overline{\text{IRQ1}}$, and $\overline{\text{IRQ2}}$.
8	FRAMEP/PARITYP	Frame/Parity Input, Positive.
9	FRAMEN/PARITYN	Frame/Parity Input, Negative.
10	D15P	Data Bit 15 (MSB), Positive.
11	D15N	Data Bit 15 (MSB), Negative.
12	DVDD18	1.8 V Digital Supply. Pin 12 supplies the power to the digital core and digital data ports, serial port input/output pins, $\overline{\text{RESET}}$, $\overline{\text{IRQ1}}$, and $\overline{\text{IRQ2}}$.
13	D14P	Data Bit 14, Positive.
14	D14N	Data Bit 14, Negative.
15	D13P	Data Bit 13, Positive.
16	D13N	Data Bit 13, Negative.
17	D12P	Data Bit 12, Positive.
18	D12N	Data Bit 12, Negative.
19	DVDD18	1.8 V Digital Supply. Pin 19 supplies power to the digital core, digital data ports, serial port input/output pins, $\overline{\text{RESET}}$, $\overline{\text{IRQ1}}$, and $\overline{\text{IRQ2}}$.
20	D11P	Data Bit 11, Positive.
21	D11N	Data Bit 11, Negative.
22	D10P	Data Bit 10, Positive.
23	D10N	Data Bit 10, Negative.

Pin No.	Mnemonic	Description
24	D9P	Data Bit 9, Positive.
25	D9N	Data Bit 9, Negative.
26	D8P	Data Bit 8, Positive.
27	D8N	Data Bit 8, Negative.
28	DCIP	Data Clock Input, Positive.
29	DCIN	Data Clock Input, Negative.
30	D7P	Data Bit 7, Positive.
31	D7N	Data Bit 7, Negative.
32	D6P	Data Bit 6, Positive.
33	D6N	Data Bit 6, Negative.
34	D5P	Data Bit 5, Positive.
35	D5N	Data Bit 5, Negative.
36	DVDD18	1.8 V Digital Supply. Pin 36 supplies power to the digital core, digital data ports, serial port input/output pins, RESET, IRQ1, and IRQ2.
37	D4P	Data Bit 4, Positive.
38	D4N	Data Bit 4, Negative.
39	D3P	Data Bit 3, Positive.
40	D3N	Data Bit 3, Negative.
41	D2P	Data Bit 2, Positive.
42	D2N	Data Bit 2, Negative.
43	DVDD18	1.8 V Digital Supply. Pin 43 supplies power to the digital core, digital data ports, serial port input/output pins, RESET, IRQ1, and IRQ2.
44	D1P	Data Bit 1, Positive.
45	D1N	Data Bit 1, Negative.
46	D0P	Data Bit 0, Positive.
47	D0N	Data Bit 0, Negative.
48	DVDD18	1.8 V Digital Supply. Pin 48 supplies power to the digital core, digital data ports, serial port input/output pins, RESET, IRQ1, and IRQ2.
49	DVDD18	1.8 V Digital Supply. Pin 49 supplies power to the digital core, digital data ports, serial port input/output pins, RESET, IRQ1, and IRQ2.
50	$\overline{\text{IRQ2}}$	Second Interrupt Request. Open-drain, active low output. Connect an external pull-up to DVDD18 through a 10 kΩ resistor.
51	$\overline{\text{IRQ1}}$	First Interrupt Request. Open-drain, active low output. Connect an external pull-up to DVDD18 through a 10 kΩ resistor.
52	SDIO	Serial Port Data Input/Output. CMOS levels with respect to DVDD18.
53	SCLK	Serial Port Clock Input. CMOS levels with respect to DVDD18.
54	$\overline{\text{CS}}$	Serial Port Chip Select. Active low (CMOS levels with respect to DVDD18).
55	AVDD33	3.3 V Analog Supply.
56	IOUT2P	QDAC Positive Current Output.
57	IOUT2N	QDAC Negative Current Output.
58	AVDD33	3.3 V Analog Supply.
59	CVDD18	1.8 V Clock Supply. Supplies clock receivers and clock distribution.
60	CVDD18	1.8 V Clock Supply. Supplies clock receivers and clock distribution.
61	DACCLKN	DAC Clock Input, Negative.
62	DACCLKP	DAC Clock Input, Positive.
63	CVDD18	1.8 V Clock Supply. Supplies clock receivers and clock distribution.
64	CVDD18	1.8 V Clock Supply. Supplies clock receivers and clock distribution.
65	AVDD33	3.3 V Analog Supply.
66	IOUT1N	IDAC Negative Current Output.
67	IOUT1P	IDAC Positive Current Output.
68	AVDD33	3.3 V Analog Supply.
69	FSADJ	Full-Scale Current Output Adjust. Place a 10 kΩ resistor from this pin to GND.
70	VREF	Voltage Reference. Nominally 1.2 V output. Decouple VREF to GND.

Pin No.	Mnemonic	Description
71	CVDD18	1.8 V Clock Supply. Pin 71 supplies the clock receivers, clock multiplier, and clock distribution.
72	CVDD18	1.8 V Clock Supply. Pin 72 supplies the clock receivers, clock multiplier, and clock distribution.
	EPAD	Exposed Pad. The exposed pad (EPAD) must be soldered to the ground plane (AVSS, DVSS, CVSS). The EPAD provides an electrical, thermal, and mechanical connection to the board.

TYPICAL PERFORMANCE CHARACTERISTICS

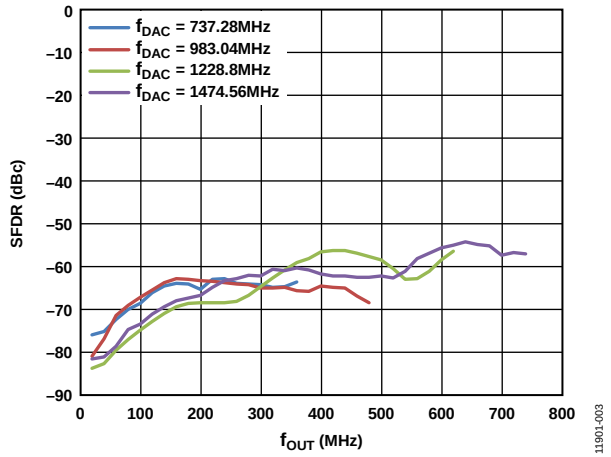


Figure 3. Single Tone (0 dBFS) SFDR vs. f_{OUT} in the First Nyquist Zone over f_{DAC}

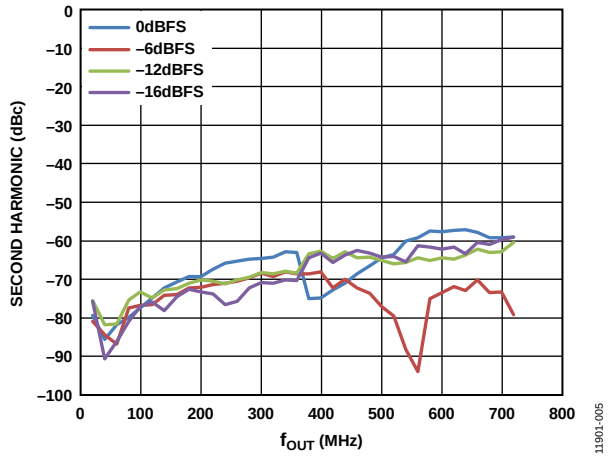


Figure 4. Single Tone Second Harmonic vs. f_{OUT} in the First Nyquist Zone over Digital Back Off, f_{DAC} = 1474.56 MHz

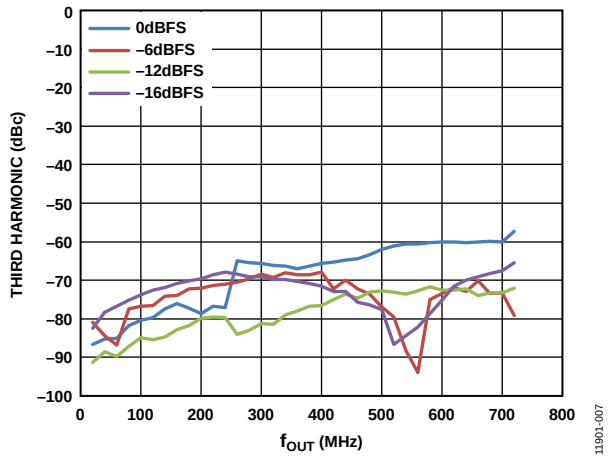


Figure 5. Single Tone Third Harmonic vs. f_{OUT} in the First Nyquist Zone over Digital Back Off, f_{DAC} = 1474.56 MHz

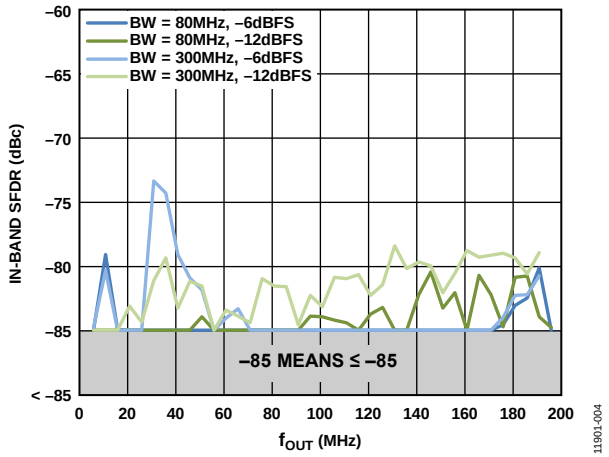


Figure 6. In-Band, Single Tone SFDR (Excluding Second Harmonic) vs. f_{OUT} in 80 MHz and 300 MHz Bandwidths, f_{DAC} = 737.28 MHz

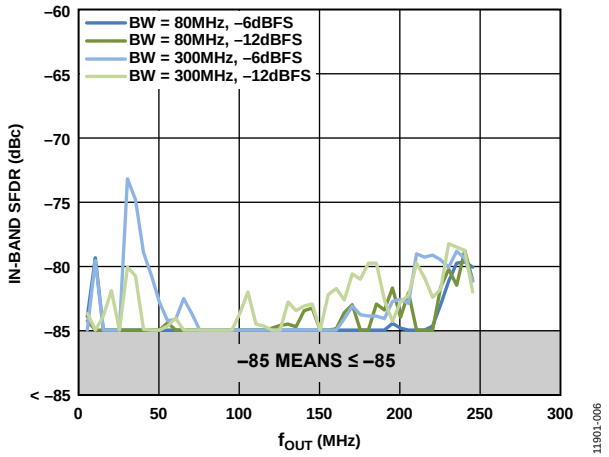


Figure 7. In-Band, Single Tone SFDR (Excluding Second Harmonic) vs. f_{OUT} in 80 MHz and 300 MHz BW, f_{DAC} = 983.04 MHz

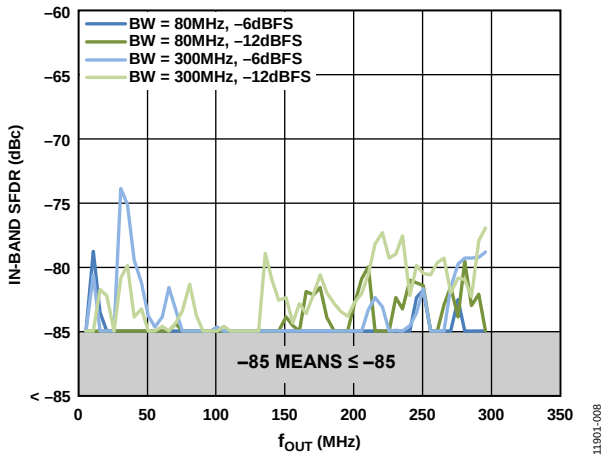


Figure 8. In-Band, Single Tone SFDR (Excluding Second Harmonic) vs. f_{OUT} in 80 MHz and 300 MHz Bandwidths, f_{DAC} = 1228.8 MHz

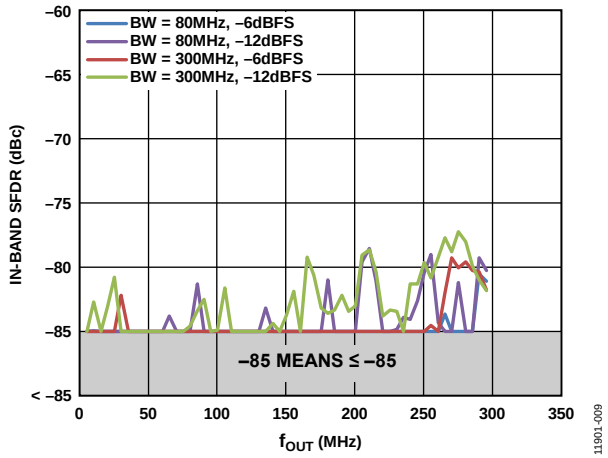


Figure 9. In-Band, Single Tone SFDR (Excluding Second Harmonic) vs. f_{OUT} in 80 MHz and 300 MHz Bandwidths, f_{DAC} = 1474.56 MHz

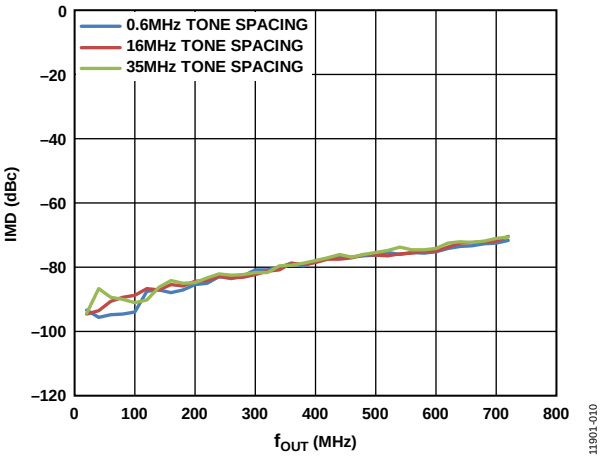


Figure 12. Two Tone, Third IMD vs. f_{OUT} over Tone Spacing, f_{DAC} = 1474.56 MHz

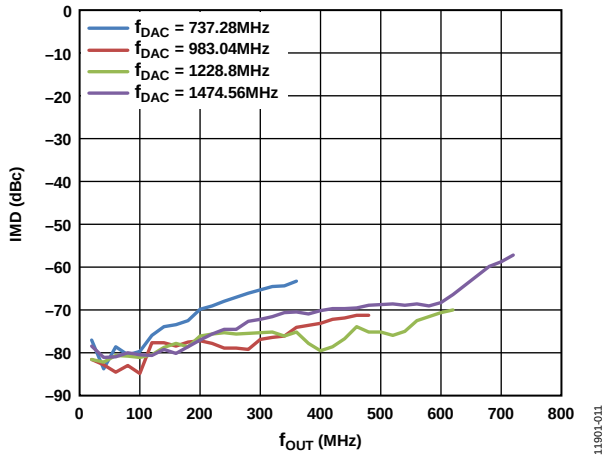


Figure 10. Two Tone, Third IMD vs. f_{OUT} over f_{DAC}

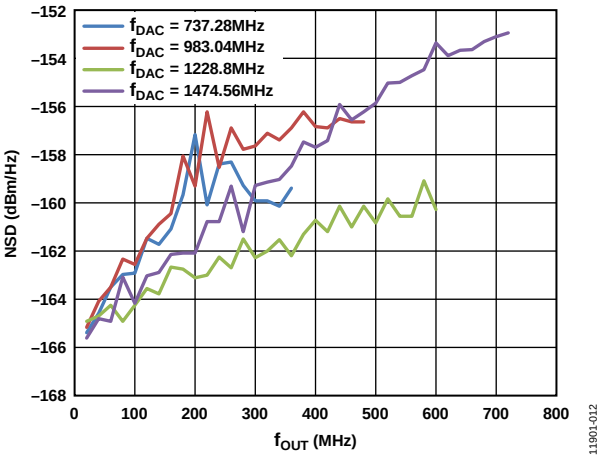


Figure 13. Single Tone (0 dBFS) NSD vs. f_{OUT} over f_{DAC}

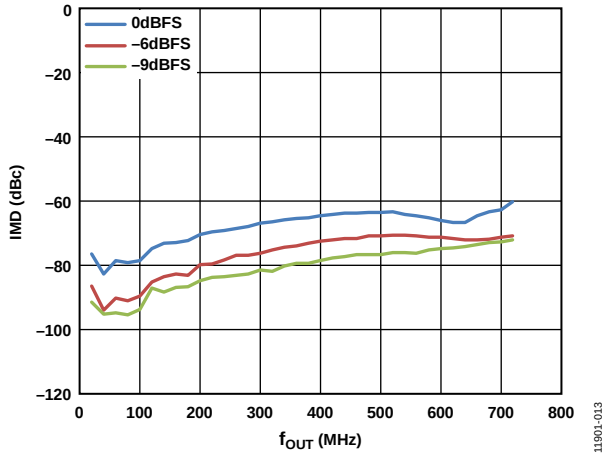


Figure 11. Two Tone, Third IMD vs. f_{OUT} over Digital Back Off, f_{DAC} = 1474.56 MHz

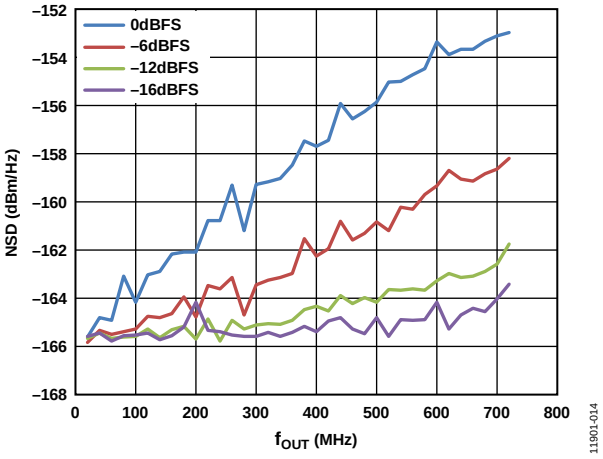


Figure 14. Single Tone NSD vs. f_{OUT} over Digital Back Off, f_{DAC} = 1474.56 MHz

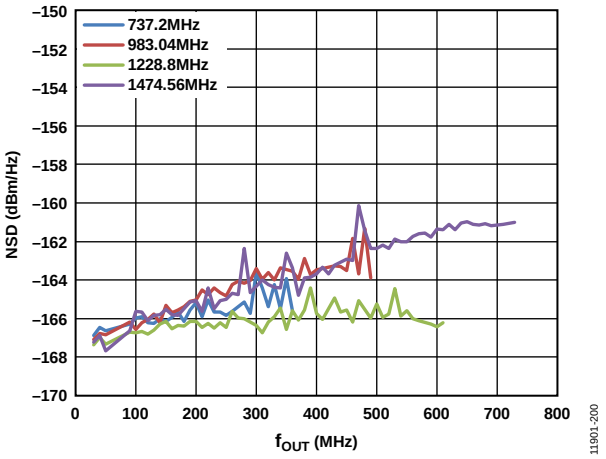


Figure 15. 1C WCDMA NSD vs. f_{OUT}, over f_{DAC}

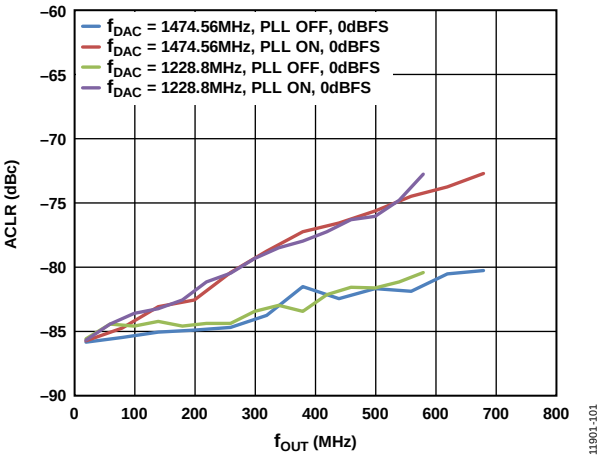


Figure 18. 1C WCDMA, Second Adjacent ACLR vs. f_{OUT}, PLL On and Off

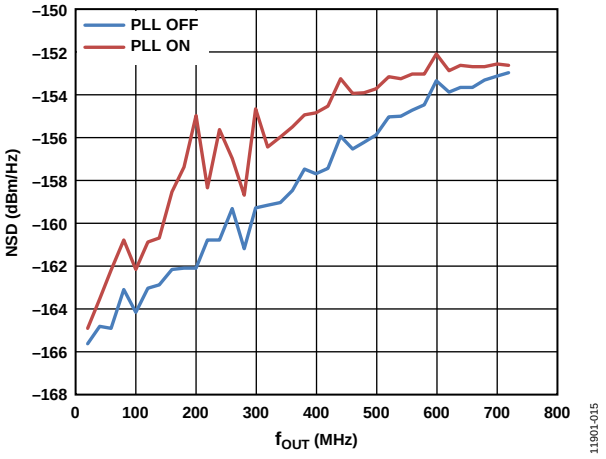


Figure 16. Single Tone NSD vs. f_{OUT}, f_{DAC} = 1474.28 MHz, PLL On and Off

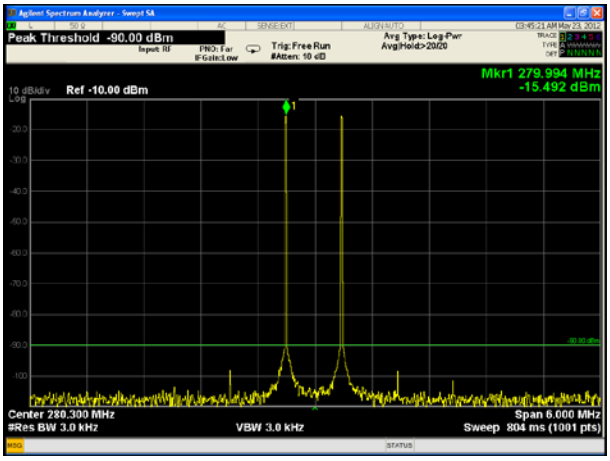


Figure 19. Two Tone, Third IMD Performance, IF = 280 MHz, f_{DAC} = 1474.28 MHz

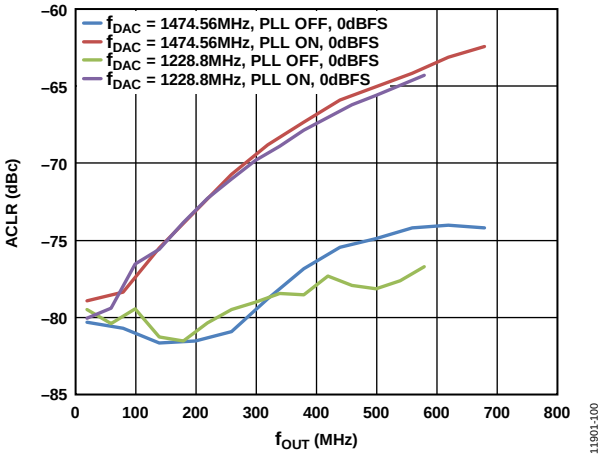


Figure 17. 1C WCDMA, First Adjacent ACLR vs. f_{OUT}, PLL On and Off

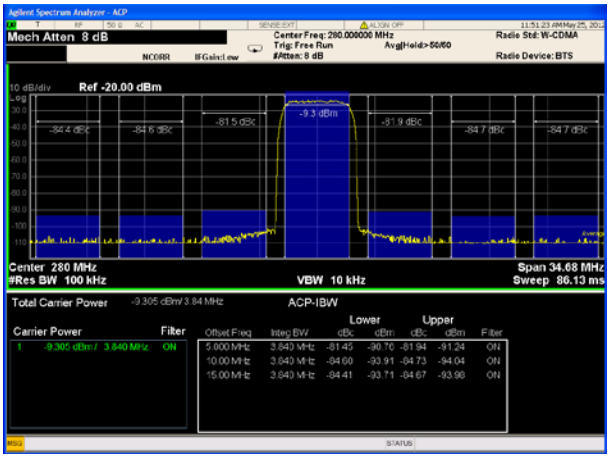


Figure 20. 1C WCDMA ACLR Performance, IF = 280 MHz, f_{DAC} = 1474.28 MHz

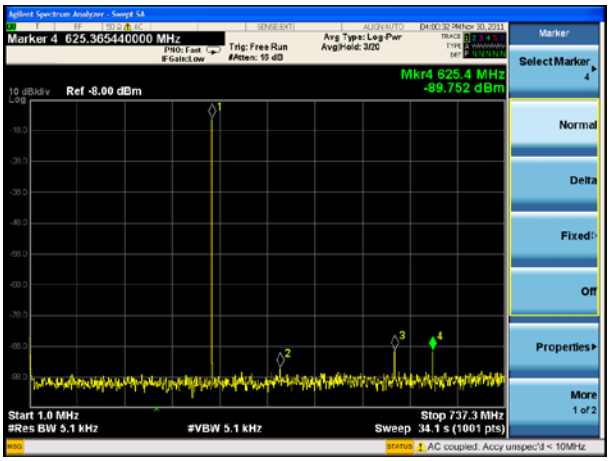


Figure 21. Single Tone f_{DAC} = 1474.56 MHz, f_{OUT} = 280 MHz, −14 dBFS

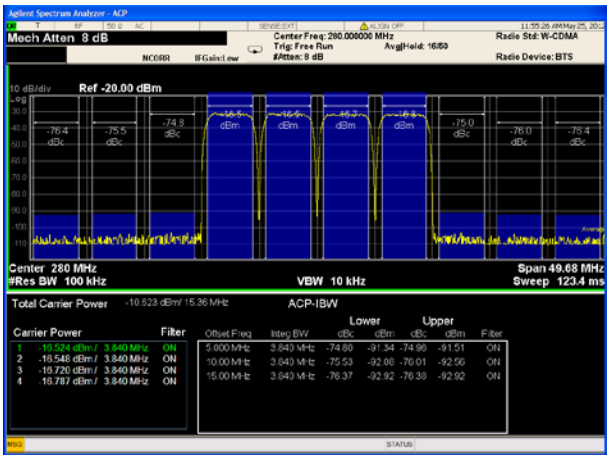


Figure 22. 4C WCDMA ACLR Performance, IF = 280 MHz, f_{DAC} = 1474.28 MHz

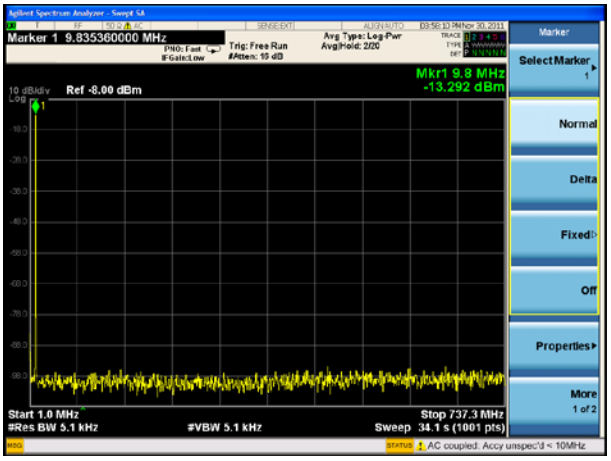


Figure 23. Single Tone SFDR f_{DAC} = 1474.56 MHz, 4× Interpolation, f_{OUT} = 10 MHz, −14 dBFS

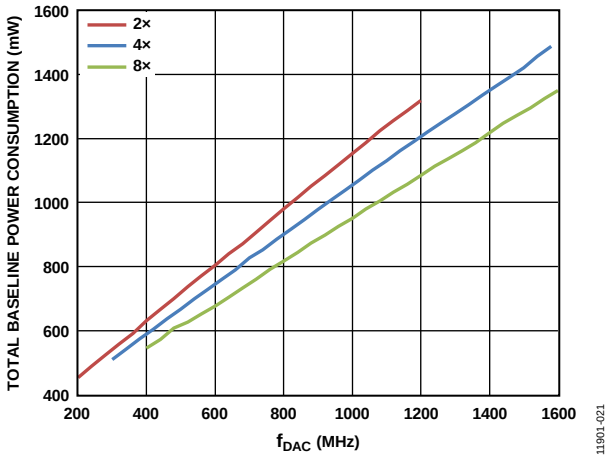


Figure 24. Total Power Baseline Consumption vs. f_{DAC} over Interpolation

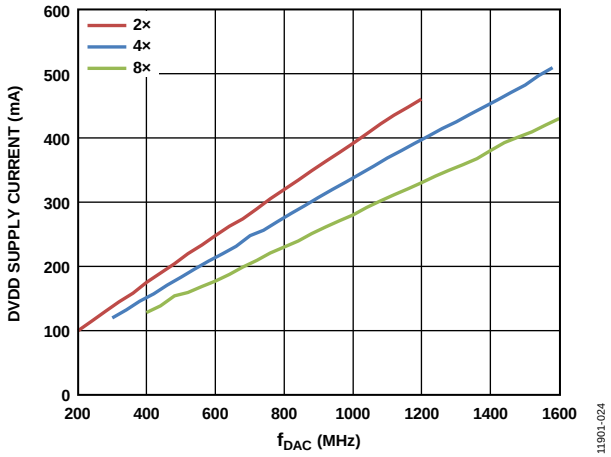


Figure 25. DVDD18 Supply Current vs. f_{DAC} over Interpolation

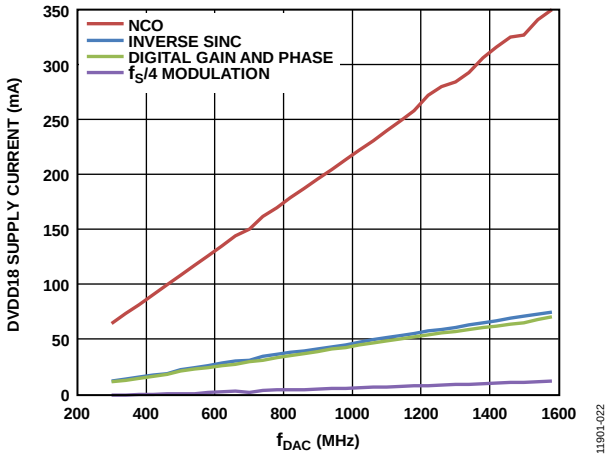


Figure 26. DVDD18 Supply Current vs. f_{DAC} over Digital Functions

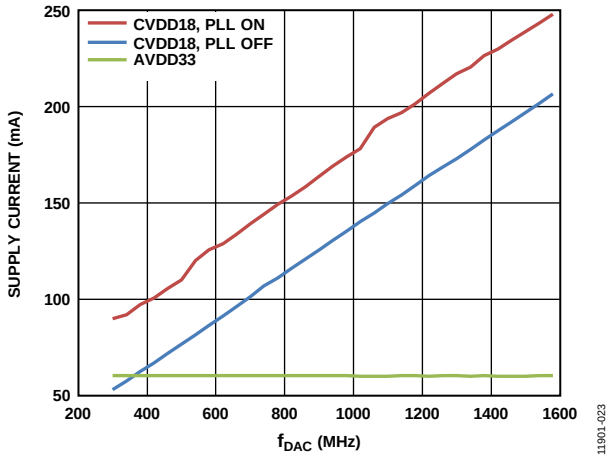


Figure 27. CVDD18, AVDD33 Supply Current vs. f_{DAC}

TERMINOLOGY

Integral Nonlinearity (INL)
INL is the maximum deviation of the actual analog output from the ideal output, determined by a straight line drawn from zero scale to full scale.

Differential Nonlinearity (DNL)
DNL is the measure of the variation in analog value, normalized to full scale, associated with a 1 LSB change in digital input code.

Offset Error
Offset error is the deviation of the output current from the ideal of 0 mA. For IOUT1P, 0 mA output is expected when all inputs are set to 0. For IOUT1N, 0 mA output is expected when all inputs are set to 1.

Gain Error
Gain error is the difference between the actual and ideal output span. The actual span is determined by the difference between the output when all inputs are set to 1 and the output when all inputs are set to 0.

Output Compliance Range
The output compliance range is the range of allowable voltage at the output of a current output DAC. Operation beyond the maximum compliance limits can cause either output stage saturation or breakdown, resulting in nonlinear performance.

Temperature Drift
Temperature drift is specified as the maximum change from the ambient (25°C) value to the value at either T_{MIN} or T_{MAX}. For offset and gain drift, the drift is reported in ppm of full-scale range (FSR) per degree Celsius. For reference drift, the drift is reported in ppm per degree Celsius.

Power Supply Rejection (PSR)
PSR is the maximum change in the full-scale output as the supplies are varied from minimum to maximum specified voltages.

Settling Time
Settling time is the time required for the output to reach and remain within a specified error band around its final value, measured from the start of the output transition.

Spurious-Free Dynamic Range (SFDR)
SFDR is the difference, in decibels, between the peak amplitude of the output signal and the peak spurious signal within the dc to Nyquist frequency of the DAC. Typically, the interpolation filters reject energy in this band. This specification, therefore, defines how well the interpolation filters work and the effect of other parasitic coupling paths on the DAC output.

Signal-to-Noise Ratio (SNR)
SNR is the ratio of the rms value of the measured output signal to the rms sum of all other spectral components below the Nyquist frequency, excluding the first six harmonics and dc. The value for SNR is expressed in decibels.

Interpolation Filter
If the digital inputs to the DAC are sampled at a multiple rate of f_{DATA} (interpolation rate), a digital filter can be constructed that has a sharp transition band near $f_{DATA}/2$. Images that typically appear around f_{DAC} (output data rate) can be greatly suppressed.

Adjacent Channel Leakage Ratio (ACLR)
ACLR is the ratio in decibels relative to the carrier (dBc) between the measured power within a channel relative to its adjacent channel.

Complex Image Rejection
In a traditional two-part upconversion, two images are created around the second IF frequency. These images have the effect of wasting transmitter power and system bandwidth. By placing the real part of a second complex modulator in series with the first complex modulator, either the upper or lower frequency image near the second IF can be rejected.

SERIAL PORT OPERATION

The serial port is a flexible, synchronous serial communications port that allows easy interfacing to many industry standard microcontrollers and microprocessors. The serial I/O is compatible with most synchronous transfer formats, including both the Motorola® SPI and Intel® SSR protocols. The interface allows read/write access to all registers that configure the AD9142A. MSB-first or LSB-first transfer formats are supported. The serial port interface is a 3-wire only interface. The input and output share a single pin input/output (SDIO).

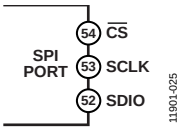


Figure 28. Serial Port Interface Pins

There are two phases to a communication cycle with the AD9142A. Phase 1 is the instruction cycle (the writing of an instruction byte into the device), coincident with the first 16 SCLK rising edges. The instruction word provides the serial port controller with information regarding the data transfer cycle, Phase 2, of the communication cycle. The Phase 1 instruction word defines whether the upcoming data transfer is a read or write, along with the starting register address for the next data transfer in the cycle.

A logic high on the $\overline{CS}$ pin, followed by a logic low, resets the serial port timing to the initial state of the instruction cycle. From this state, the next 16 rising SCLK edges represent the instruction bits of the current I/O operation.

The remaining SCLK edges are for Phase 2 of the communication cycle. Phase 2 is the actual data transfer between the device and the system controller. Phase 2 of the communication cycle is a transfer of one data byte. Registers change immediately upon writing to the last bit of each transfer byte, except for the frequency tuning word and NCO phase offsets, which change only when the frequency tuning word (FTW) update bit is set.

DATA FORMAT

The instruction byte contains the information shown in Table 10.

Table 10. Serial Port Instruction Word

I15 (MSB)	I[14:0]
R/$\overline{W}$	A[14:0]

R/$\overline{W}$ (Bit 15 of the instruction word) determines whether a read or a write data transfer occurs after the instruction word write. Logic 1 indicates a read operation and Logic 0 indicates a write operation.

A14 to A0 (Bit 14 to Bit 0 of the instruction word) determine the register that is accessed during the data transfer portion of the communication cycle. For multibyte transfers, A14 is the starting address; the device generates the remaining register addresses based on the SPI_LSB_FIRST bit.

SERIAL PORT PIN DESCRIPTIONS

Serial Clock (SCLK)

The serial clock pin synchronizes data to and from the device and runs the internal state machines. The maximum frequency of SCLK is 40 MHz. All data input is registered on the rising edge of SCLK. All data is driven out on the falling edge of SCLK.

Chip Select ($\overline{CS}$)

$\overline{CS}$ is an active low input that starts and gates a communication cycle. It allows more than one device to be used on the same serial communications line. The SDIO pins enter a high impedance state when the $\overline{CS}$ input is high. During the communication cycle, $\overline{CS}$ should stay low.

Serial Data I/O (SDIO)

The SDIO pin is a bidirectional data line.

SERIAL PORT OPTIONS

The serial port can support both MSB-first and LSB-first data formats. This functionality is controlled by the SPI_LSB_FIRST bit (Register 0x00, Bit 6). The default is MSB first (LSB_FIRST = 0).

When SPI_LSB_FIRST = 0 (MSB first), the instruction and data bits must be written from MSB to LSB. Multibyte data transfers in MSB-first format start with an instruction word that includes the register address of the most significant data byte. Subsequent data bytes must follow from high address to low address. In MSB-first mode, the serial port internal word address generator decrements for each data byte of the multibyte communication cycle.

When SPI_LSB_FIRST = 1 (LSB first), the instruction and data bits must be written from LSB to MSB. Multibyte data transfers in LSB-first format start with an instruction word that includes the register address of the least significant data byte. Subsequent data bytes must follow from low address to high address. In LSB-first mode, the serial port internal word address generator increments for each data byte of the multibyte communication cycle.

If the MSB-first mode is active, the serial port controller data address decrements from the data address written toward 0x00 for multibyte I/O operations. If the LSB-first mode is active, the serial port controller data address increments from the data address written toward 0xFF for multibyte I/O operations.

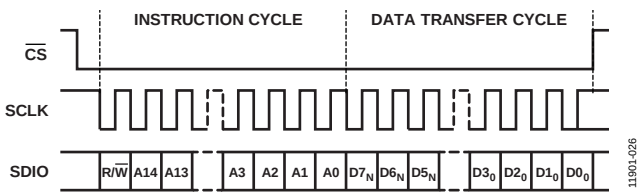


Figure 29. Serial Register Interface Timing, MSB First

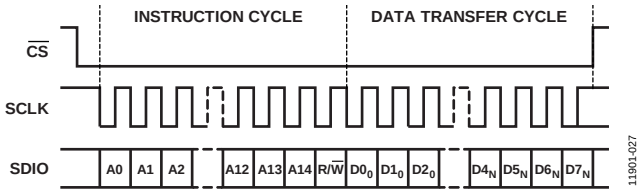


Figure 30. Serial Register Interface Timing, LSB First

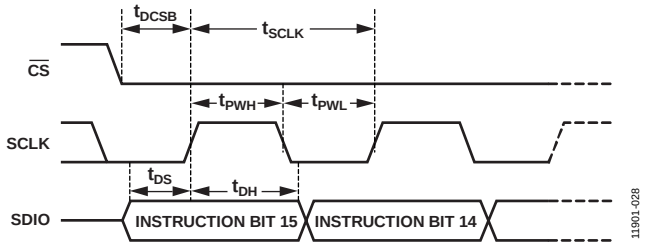


Figure 31. Timing Diagram for Serial Port Register Write

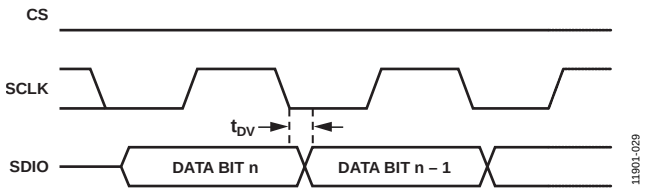


Figure 32. Timing Diagram for Serial Port Register Read

DATA INTERFACE

LVDS INPUT DATA PORTS

The AD9142A has a 16-bit LVDS bus that accepts 16-bit I and Q data either in word (16-bit) or byte (8-bit) formats. In the word interface mode, the data is sent over the entire 16-bit data bus. In the byte interface mode, the data is sent over the lower 8-bit (D7 to D0) LVDS bus. Table 11 lists the pin assignment of the bus and the SPI register configuration for each mode.

Table 11. LVDS Input Data Modes

Interface Mode	Pin Assignment	SPI Register Configuration
Word	D15 to D0	Register 0x26, Bit 0 = 0
Byte	D7 to D0	Register 0x26, Bit 0 = 1

WORD INTERFACE MODE

In word interface mode, the digital clock input (DCI) signal is a reference bit that generates a double data rate (DDR) data sampling clock. Time align the DCI signal with the data. The IDAC data follows the rising edge of the DCI, and the QDAC data follows the falling edge of the DCI, as shown in Figure 33.

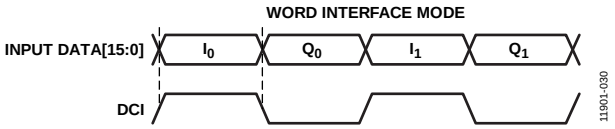


Figure 33. Timing Diagram for Word Interface Mode

BYTE INTERFACE MODE

In byte interface mode, the required sequence of the input data stream is I[15:8], I[7:0], Q[15:8], Q[7:0]. A frame signal is required to align the order of input data bytes properly. Time align both the DCI signal and frame signal with the data. The rising edge of the frame indicates the start of the sequence. The frame can be either a one shot or periodical signal as long as its first rising edge is correctly captured by the device. For a one shot frame, the frame pulse must be held at high for at least one DCI cycle. For a periodical frame, the frequency needs to be

$$f_{DCI}/(2 \times n)$$

where n is a positive integer, that is, 1, 2, 3, …

Figure 34 is an example of signal timing in byte mode.

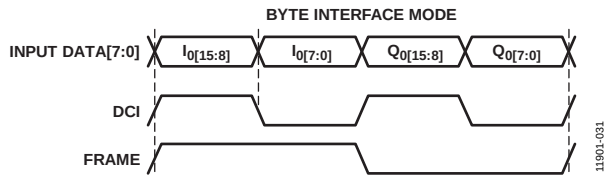


Figure 34. Timing Diagram for Byte Interface Mode

DATA INTERFACE CONFIGURATION OPTIONS

To provide more flexibility for the data interface, some additional options are listed in Table 12.

Table 12. Data Interface Configuration Options

Register 0x26	Description
DATA_FORMAT (Bit 7)	Select between binary and twos complement formats.
DATA_PAIRING (Bit 6)	Indicate I/Q data pairing on data input. This allows the I and Q data that is received to be paired in various ways.
DATA_BUS_INVERT (Bit 5)	Swaps the bit order of the data input port. Remaps the input data from D[15:0] to D[0:15].

DLL INTERFACE MODE

A source synchronous LVDS interface is used between the data host and AD9142A to achieve high data rates while simplifying the interface. The FPGA or ASIC feeds the AD9142A with 16-bit input data. Along with the input data, the FPGA or ASIC provides a DDR (double data rate) data clock input (DCI).

A delay locked loop (DLL) circuit designed to operate with DCI clock rates between 250 and 575 MHz is used to generate a phase-shifted version of the DCI, called DSC (data sampling clock), to register the input data on both the rising and falling edges.

As shown in Figure 35, the DCI clock edges must be coincident with the data bit transitions with minimum skew and jitter. The nominal sampling point of the input data occurs in the middle of the DCI clock edges because this point corresponds to the center of the data eye. This is also equivalent to a nominal phase shift of 90°of the DCI clock.

The data timing requirements are defined by a data valid window (DVW) that is dependent on the data clock input skew, input data jitter, and the variations of the DLL delay line across delay settings. The DVW is defined as

$$DVW = t_{DATA\ PERIOD} - t_{DATA\ SKEW} - t_{DATA\ JITTER}$$

The available margin for data interface timing is given by

$$t_{MARGIN} = DVW - (t_S + t_H)$$

The difference between the setup and hold times, which is also called the keep out window, or KOW, is the area where data transitions should not happen. The timing margin allows tuning of the DLL delay setting by the user, see Figure 36.

From the figure, it can be seen that the ideal location for the DSC signal is 90° out of phase from the DCI input. However, due to skew of the DCI relative to the data, it may be necessary to change the DSC phase offset to sample the data at the center of its eye diagram. The sampling instance can be varied in discrete increments by offsetting the nominal DLL phase shift value of 90° via Register 0x0A, Bits[3:0]. This register is a signed value. The MSB is the sign and the LSBs are the magnitude. The following equation defines the phase offset relationship:

$$Phase\ Offset = 90° \pm n \times 11.25°,\ |n| < 7$$

where n is the DLL phase offset setting.

Figure 35 shows the DSC setup and hold times with respect to the DCI signal and data signals.

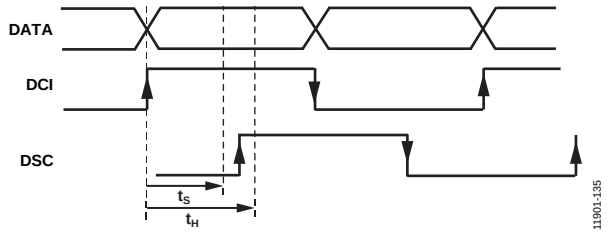


Figure 35. LVDS Data Port Setup and Hold Times

Table 13 lists the values that are guaranteed over the operating conditions. These values were taken with a 50% duty cycle and a DCI swing of 450 mV p-p. For best performance, the duty cycle variation should be kept below ±5%, and the DCI input should be as high as possible, up to 1200 mV p-p.

Table 13. DLL Phase Setup and Hold Times (Guaranteed)

Frequency, f_{DCI} (MHz)	Time (ps)	Data Port Setup and Hold Times (ps) at DLL Phase		
		−3	**0**	**+3**
307	t_S	−125	−385	−695
	t_H	834	1120	1417
368	t_S	−70	−305	−534
	t_H	753	967	1207
491	t_S	−81	−245	−402
	t_H	601	762	928

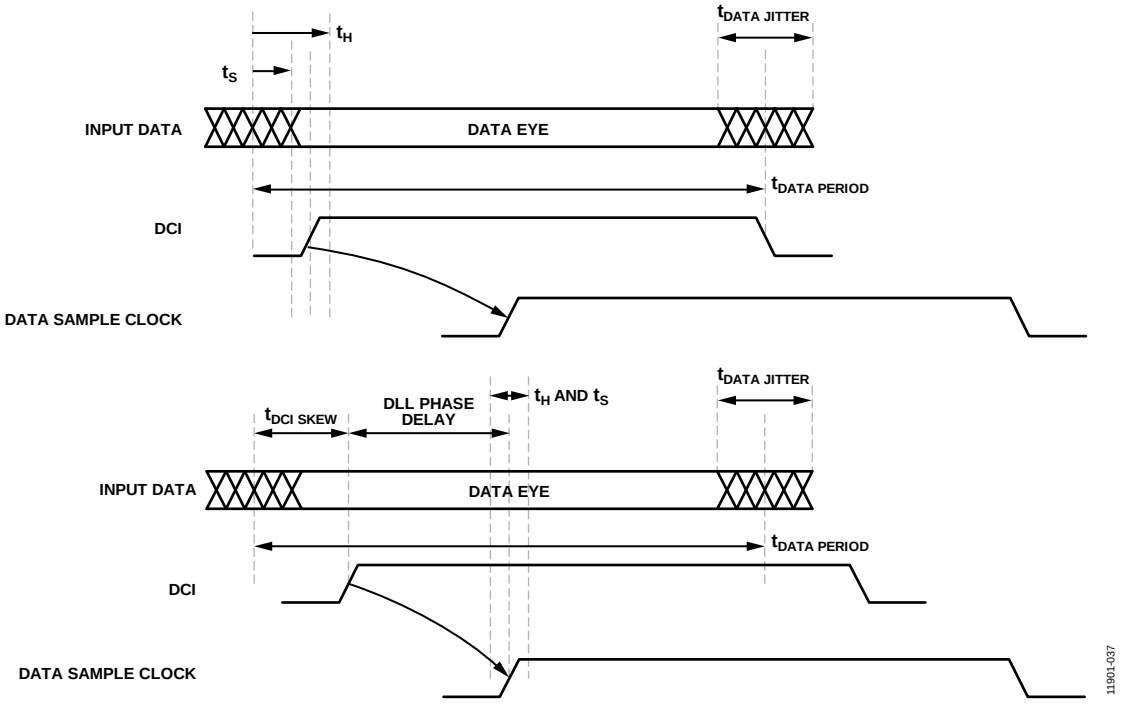


Figure 36. LVDS Data Port Timing Requirements

Table 14. DLL Phase Setup and Hold Times (Typical)

Frequency, f_{DCI}[1] (MHz)	Time (ps)	Data Port Setup and Hold Times (ps) at DLL Phase												
		−6	−5	−4	−3	−2	−1	0	+1	+2	+3	+4	+5	+6
250	t_S	−93	−196	−312	−416	−530	−658	−770	−878	−983	−1093	−1193	−1289	−1412
	t_H	468	579	707	825	947	1067	1188	1315	1442	1570	1697	1777	1876
275	t_S	−87	−172	−264	−364	−464	−556	−653	−756	−859	−956	−1053	−1151	−1251
	t_H	451	537	646	757	878	977	1092	1218	1311	1423	1537	1653	1728
300	t_S	−82	−166	−256	−341	−426	−515	−622	−715	−809	−900	−1001	−1097	−1184
	t_H	422	500	598	703	803	897	1000	1105	1203	1303	1411	1522	1612
325	t_S	−46	−114	−190	−271	−358	−447	−538	−612	−706	−806	−891	−966	−1044
	t_H	405	483	563	647	740	832	914	1000	1100	1200	1292	1380	1476
350	t_S	−23	−92	−180	−252	−328	−409	−491	−574	−654	−731	−819	−889	−959
	t_H	383	451	524	607	682	762	844	930	1011	1097	1186	1277	1358
375	t_S	−7	−82	−150	−225	−315	−391	−461	−526	−595	−661	−726	−786	−853
	t_H	401	466	504	569	641	718	783	863	941	1025	1106	1187	1264
400	t_S	−46	−98	−161	−243	−303	−384	−448	−513	−578	−643	−713	−771	−833
	t_H	385	445	503	546	604	674	748	826	890	965	1039	1110	1178
425	t_S	4	−52	−110	−170	−229	−297	−394	−449	−517	−579	−641	−704	−752
	t_H	358	408	465	524	595	625	692	762	829	900	966	1032	1097
450	t_S	11	−34	−92	−147	−209	−269	−324	−386	−446	−509	−564	−622	−672
	t_H	354	406	457	516	573	637	693	731	792	852	917	983	1042
475	t_S	−15	−51	−95	−147	−198	−255	−313	−366	−425	−480	−530	−585	−640
	t_H	355	399	451	499	556	613	675	727	779	815	873	930	988
500	t_S	9	−28	−77	−128	−183	−233	−288	−333	−390	−438	−495	−545	−594
	t_H	313	354	399	445	500	555	615	668	726	783	825	881	934
525	t_S	−7	−52	−100	−147	−187	−237	−285	−335	−387	−436	−483	−530	−581
	t_H	311	356	395	438	489	537	592	645	692	746	799	850	909
550	t_S	−5	−39	−74	−107	−147	−192	−249	−302	−352	−397	−440	−486	−529
	t_H	300	340	378	423	468	510	560	610	659	710	756	810	865
575	t_S	8	−28	−66	−102	−143	−181	−245	−280	−336	−366	−406	−443	−488
	t_H	312	348	379	414	453	496	544	599	654	708	759	806	847

[1] Table 14 shows characterization data for selected f_{DCI} frequencies. Other frequencies are possible, and Table 14 can be used to estimate performance.

Table 14 shows the typical times for various DCI clock frequencies that are required to calculate the data valid margin. The amount of margin that is available for tuning of the DSC sampling point can be determined using Table 14.

Maximizing the opening of the eye in both the DCI and data signals improves the reliability of the data port interface. Differential controlled impedance traces of equal length (that is, delay) should be used between the host processor and the AD9142A input. To ensure coincident transitions with the data bits, the DCI signal should be implemented as an additional data line with an alternating (010101…) bit sequence from the same output drivers used for the data.

The DCI signal is ac-coupled by default; thus, removing the DCI signal may cause DAC output chatter due to randomness on the DCI input. To avoid this, it is recommended that the DAC output is disabled whenever the DCI signal is not present. To do this, program the DAC output current power down bit in Register 0x01, Bit 7 and Bit 6 to 1. When the DCI signal is again present, the DAC output can be enabled by setting Register 0x01, Bit 7 and Bit 6 to 0.

Register 0x0D optimizes the DLL stability over the operating frequency range. Table 15 shows the recommended setting.

Table 15. DLL Configuration Options

DCI Speed	Register 0x0D
≥350 MHz	0x06
<350 MHz	0x86

The status of the DLL can be polled by reading the data status register at Address 0x0E. Bit 0 indicates that the DLL is running and attempting lock, and Bit 7 is set to when the DLL has locked. Bit 2 is 1 when a valid data clock in is detected. The warning bits in Register 0x0E[6:4] can be used as indicators that the DAC may be operating in a non ideal location in the delay line. Note that these bits are read at the SPI port speed, which is much slower than the actual speed of the DLL. This means they can only show a snapshot of what is happening as opposed to giving real-time feedback.

DLL Configuration Example 1

In the following DLL configuration example, f_{DCI} = 500 MHz, DLL is enabled, and DLL phase offset = 0.

```
1. 0x5E → 0xFE /* Turn off LSB delay cell*/
2. 0x0D → 0x06 /* Select DLL configure
   options */
3. 0x0A → 0xC0 /* Enable DLL and duty cycle
   correction. Set DLL phase offset to 0 */
4. Read 0x0E[7:4] /* Expect 1000b if the DLL
   is locked */
```

DLL Configuration Example 2

In the following DLL configuration example, f_{DCI} = 300 MHz, DLL is enable, and DLL phase offset = 0.

```
1. 0x5E → 0xFE /* Turn off LSB delay cell*/
2. 0x0D → 0x86 /* Select DLL configure
   options */
3. 0x0A → 0xC0 /* Enable DLL and duty cycle
   correction. Set DLL phase offset to 0 */
4. Read 0x0E[7:4] /* Expect 1000b if the DLL
   is locked */
```

PARITY

The data interface can be continuously monitored by enabling the parity bit feature in Register 0x6A, Bit 7 and configuring the frame/parity bit as parity by setting Register 0x09 to 0x21. In this case, the host sends a parity bit along with each data sample. This bit is set according to the following formulas, where n is the data sample that is being checked.

For even parity,

$$XOR[FRM(n), D0(n), D1(n), D2(n), ..., D15(n)] = 0$$

For odd parity,

$$XOR[FRM(n), D0(n), D1(n), D2(n), ..., D15(n)] = 1$$

The parity bit is calculated over 17 bits (including the frame/parity bit).

If a parity error occurs, the parity error counter (Register 0x6B or Register 0x6C) is incremented. Parity errors on the bits sampled by the rising edge of DCI increments the rising edge parity counter (Register 0x6B) and set the PARERRRIS bit (Register 0x6A, Bit 0). Parity errors on the bits sampled by the falling edge of DCI will increment the falling edge parity counter (Register 0x6C) and set the PARERRFAL bit (Register 0x6A, Bit 1). The parity counter continues to accumulate until it is cleared or until it reaches a maximum value of 255. The count can be cleared by writing a 1 to Register 0x6A, Bit 5.

To trigger an IRQ when a parity error occurs, write a 1 to Register 0x04, Bit 7. This IRQ triggers if there is either a rising edge or falling edge parity error. The status of the IRQ can be observed via Register 0x06, Bit 7 or by using the selected $\overline{IRQx}$ pin. Clear the IRQ by writing a 1 to Register 0x06, Bit 7.

Use the parity bit to validate the interface timing. As described previously, the host provides a parity bit with the data samples, as well as configures the AD9142A to generate an IRQ. The user can then sweep the sampling instance of the AD9142A input registers to determine at what point a sampling error occurs. The sampling instance can be varied in discrete increments by offsetting the nominal DLL phase shift value of 90° via Register 0x0A[3:0].

SED OPERATION

The AD9142A provides on-chip sample error detection (SED) circuitry that simplifies verification of the input data interface. The SED compares the input data samples captured at the digital input pins with a set of comparison values. The comparison values are loaded into registers through the SPI port. Differences between the captured values and the comparison values are detected. Options are available for customizing SED test sequencing and error handling.

The SED circuitry allows the application to test a short user defined pattern to confirm that the high speed source synchronous data bus is correctly implemented and meets the timing requirement. Unlike the parity bit, the SED circuitry is expected to be used during initial system calibration, before the AD9142A is in use in the application. The SED circuitry operates on a data set made up of user defined input words, denoted as I0, Q0, I1, and Q1. The user defined pattern consists of sequential data word samples (I0 is sampled on the rising edge of DCI, Q0 is sampled on the following falling edge of DCI, I1 is sampled on the following DCI rising edge, and Q1 is sampled on the following DCI falling edge). The user loads this data pattern in the byte format into Register 0x61 through Register 0x68.

The depth of the user defined pattern is selectable via Bit 4 in the SED_CTRL register (0x60), with the default, 0, meaning a depth of two (using I0 and Q0), and a 1 meaning a depth of four (using I0, Q0, I1, and Q1, and requiring the use of frame signal input to define I0 to the SED state machine). To properly align the input samples using a depth of four, I0 is indicated by asserting the frame signal for a minimum of two complete input samples as shown in Figure 37. The frame signal can be issued once at the start of the data transmission, or it can be asserted repeatedly at intervals coinciding with the S0 word.

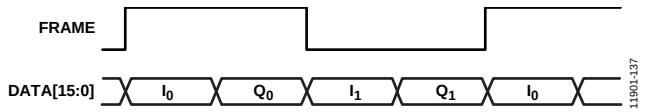


Figure 37. Timing Diagram of Extended FRAME Signal Required to Align Input Data for SED

The SED has three flag bits (Register 0x60, Bit 0, Bit 1, and Bit 2) that indicate the results of the input sample comparisons. The sample error detected bit (Register 0x60, Bit 0) is set when an error is detected and remains set until cleared.

The autosample error detection (AED) mode is an autoclear mode that has two effects: it activates the compare fail bit and the compare pass bit (Register 0x60, Bit 1 and Bit 2). The compare pass bit sets if the last comparison indicated that the sample was error free. The compare fail bit sets if an error is detected. The compare fail bit is automatically cleared by the reception of eight consecutive error-free comparisons, when autoclear mode is enabled.

The sample error flag can be configured to trigger an IRQ when active, if needed. This is done by enabling the appropriate bit in the event flag register (Register 4, Bit 6).

SED EXAMPLE

Normal Operation

The following example illustrates the AD9142A SED configuration sequence for continuously monitoring the input data and assertion of an IRQ when a single error is detected:

1. Write to the following registers to enable the SED and load the comparison values with a 4-deep user pattern. Comparison values can be chosen arbitrarily; however, choosing values that require frequent bit toggling provides the most robust test.
   a. Register 0x61[7:0] → I0[7:0]
   b. Register 0x62[7:0] → I0[15:8]
   c. Register 0x63[7:0] → Q0[7:0]
   d. Register 0x64[7:0] → Q0[15:8]
   e. Register 0x65[7:0] → I1[7:0]
   f. Register 0x66[7:0] → I1[15:8]
   g. Register 0x67[7:0] → Q1[7:0]
   h. Register 0x68[7:0] → Q1[15:8]
2. Enable SED.
   a. Register 0x60 → 0xD0
   b. Register 0x60 → 0x90
3. Enable the SED error detect flag to assert the $\overline{\text{IRQx}}$ pin.
   a. Register 0x04[6] = 1
4. Set up frame parity as the frame signal.
   a. Register 0x09 = 0x12
5. Begin transmitting the input data pattern (frame signal) is also required because the depth of the pattern is 4).

DELAY LINE INTERFACE MODE

The DLL is designed to help ease the interface timing requirements in very high speed data rate applications. The DLL has a minimum supported interface speed of 250 MHz, as shown in Table 2. For interface rates lower than this speed, use the interface delay line. In this mode, the DLL is powered off and a four-tap delay line is provided for the user to adjust the timing between the data bus and the DCI. Table 16 specifies the setup and hold times for each delay tap.

Table 16. Delay Line Setup and Hold Times (Guaranteed)

Delay Setting	0	1	2	3
Register 0x5E[7:0]	**0x00**	**0x80**	**0xF0**	**0xFE**
Register 0x5F[2:0]	**0x60**	**0x67**	**0x67**	**0x67**
t_S (ns)[1]	−0.81	−0.97	−1.13	−1.28
t_H (ns)	1.96	2.20	2.53	2.79
$\lvert t_S + t_H \rvert$ (ns)	1.15	1.23	1.40	1.51

[1] The negative sign indicates the direction of the setup time. The setup time is defined as positive when it is on the left side of the clock edge and negative when it is on the right side of the clock edge.

There is a fixed 1.38 ns delay on the DCI signal when the delay line is enabled. Each tap adds a nominal delay of 200 ps to the fixed delay. To achieve the best timing margin, that is, to center the setup and hold window in the middle of the data eye, the user may need to add a delay on the data bus with respect to the DCI in the data source. Figure 38 is an example of calculating the optimal external delay.

Register 0x0D, Bit 4 configures the DCI signal coupling settings for optimal interface performance over the operating frequency range. It is recommended that this bit be set to 1 (dc-coupled DCI) in the delay line interface mode.

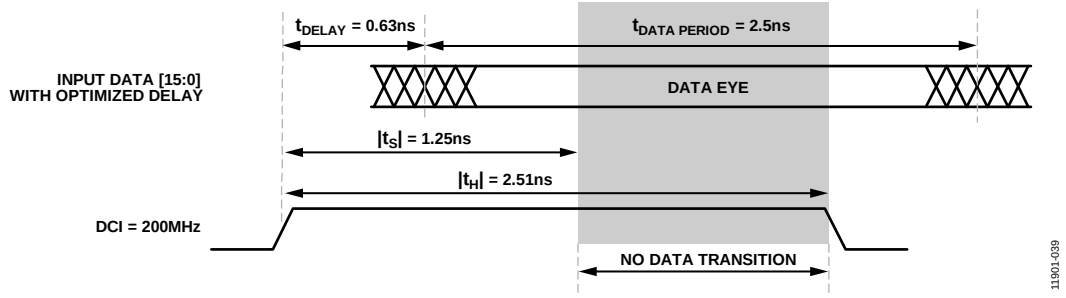


Figure 38. Example of Interfacing Timing in the Delay Line Interface Mode

Interface Timing Requirements

The following example shows how to calculate the optimal delay at the data source to achieve the best sampling timing in the delay line interface mode:

- f_{DCI} = 200 MHz
- Delay setting = 0

The shadow area in Figure 38 is the interface setup and hold time window set to 0. To optimize the interface timing, this window must be placed in the middle of the data transitions. Because the input is double data rate, the available data period is 2.5 ns. Therefore, the optimal data bus delay, with respect to the DCI at the data source, can be calculated as

$$t_{DELAY} = \frac{(|t_S| + |t_H|)}{2} - \frac{t_{DATAPERIOD}}{2} = 1.38 - 1.25 = 0.13 \text{ ns}$$

SPI Sequence to Enable Delay Line Interface Mode

Use the following SPI sequence to enable the delay line interface mode:

1. `0x5E → 0x00 /* Configure the delay setting */`
2. `0x5F → 0x60`
3. `0x0D → 0x16 /* DC couple DCI */`
4. `0x0A → 0x00 /* Turn off DLL and duty cycle correction */`

FIFO OPERATION

As is described in the Data Interface section, the AD9142A adopts source synchronous clocking in the data receiver. The nature of source synchronous clocking is the creation of a separate clock domain at the receiving device. In the DAC, it is the DAC clock domain, that is, the DACCLK. Therefore, there are two clock domains inside of the DAC: the DCI and the DACCLK. Often, these two clock domains are not synchronous, requiring an additional stage to adjust the timing for proper data transfer. In the AD9142A, a FIFO stage is inserted between the DCI and DACCLK domains to transfer the received data into the core clock domain (DACCLK) of the DAC.

The AD9142A contains a 2-channel, 16-bit wide, 8-word deep FIFO. The FIFO acts as a buffer that absorbs timing variations between the two clock domains. The timing budget between the two clock domains in the system is significantly relaxed due to the depth of the FIFO.

Figure 39 shows the block diagram of the datapath through the FIFO. The input data is latched into the device, formatted, and then written into the FIFO register, which is determined by the FIFO write pointer. The value of the write pointer is incremented every time a new word is loaded into the FIFO. Meanwhile, data is read from the FIFO register, which is determined by the read pointer, and fed into the digital datapath. The value of the read pointer is incremented every time data is read into the datapath from the FIFO. The FIFO pointers are incremented at the data rate, which is the DACCLK rate divided by the interpolation rate.

Valid data is transmitted through the FIFO as long as the FIFO does not overflow (full) or underflow (empty). An overflow or underflow condition occurs when the write pointer and read pointer point to the same FIFO slot. This simultaneous access of data leads to unreliable data transfer through the FIFO and must be avoided.

Normally, data is written to and read from the FIFO at the same rate to maintain a constant FIFO depth. If data is written to the FIFO faster than data is read, the FIFO depth increases. If data is read from the FIFO faster than data is written to it, the FIFO depth decreases. For optimal timing margin, maintain the FIFO depth near half full (a difference of four between the write pointer and read pointer values). The FIFO depth represents the FIFO pipeline delay and is part of the overall latency of the AD9142A.

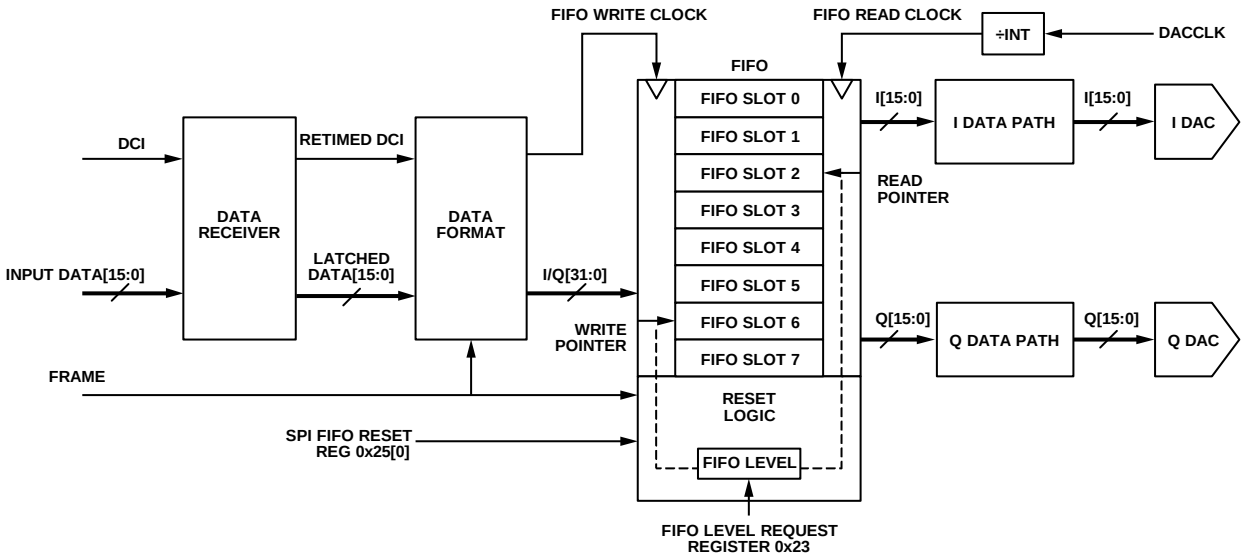


Figure 39. Block Diagram of FIFO

RESETTING THE FIFO

Upon power-on of the device, the read and write pointers start to roll around the FIFO from an arbitrary slot; consequently, the FIFO depth is unknown. To avoid a concurrent read and write to the same FIFO address and to assure a fixed pipeline delay from power-on to power-on, it is important to reset the FIFO pointers to a known state each time the device powers on or wakes up. This state is specified in the requested FIFO level (FIFO depth and FIFO level are used interchangeably in this data sheet), which consists of two sections: the integer FIFO level and the fractional FIFO level.

The integer FIFO level represents the difference of the states between the read and write points in the unit of an input data period ($1/f_{DATA}$). The fractional FIFO level represents the difference of the FIFO pointers that is smaller than the input data period. The resolution of the fractional FIFO level is the input data period divided by the interpolation ratio and, thus, it is equal to one DACCLK cycle.

The exact FIFO level, that is, the FIFO latency, can be calculated by

FIFO Latency = Integer Level + Fractional Level

Because the FIFO has eight data slots, there are eight possible FIFO integer levels. The maximum supported interpolation rate in the AD9142A is 8× interpolation. Therefore, there are eight possible FIFO fractional levels. Two 3-bit registers in Register 0x23 are assigned to represent the two FIFO levels, as follows:

- Bits[6:4] represent the FIFO integer level
- Bits[2:0] represent the FIFO fractional level.

For example, if the interpolation rate is 4× and the total FIFO depth is 4.5 input data periods, set the FIFO_LEVEL_CONFIG (Register 0x23) to 0x42 (4 here means four data cycles and 2 means two DAC cycles, which is half of a data cycle). Note that there are only four possible fractional levels in the case of 4× interpolation. Table 17 shows additional examples of configuring the FIFO level in various interpolation rate modes.

Table 17. Examples of FIFO Level Configuration

Interpolation Rate	Example FIFO Level ($1/f_{DATA}$)	Integer Level (Reg. 0x23[6:4])	Fractional Level (Reg. 0x23[2:0])
2×	3 + 1/2	3	1
4×	4 + 1/4	4	1
8×	4 + 3/8	4	3

By default, the FIFO level is 4.0. It can be programmed to any allowed value from 0.0 to 7.x. The maximum allowed number for x is the interpolation rate minus 1. For example, in 8× interpolation, the maximum value allowed for x is 7.

The following two ways are used to reset the FIFO and initialize the FIFO level:

- Serial port (SPI) initiated FIFO reset.
- Frame initiated FIFO reset.

SERIAL PORT INITIATED FIFO RESET

A SPI initiated FIFO reset is the most common method to reset the FIFO. To initialize the FIFO level through the serial port, toggle FIFO_SPI_RESET_REQUEST (Register 0x25[0]) from 0 to 1 and back to 0. When the write to this register is complete, the FIFO level is initialized to the requested FIFO level and the readback of FIFO_SPI_RESET_ACK (Register 0x25[1]) is set to 1. The FIFO level readback, in the same format as the FIFO level request, should be within ±1 DACCLK cycle of the requested level. For example, if the requested value is 0x40 in 4× interpolation, the readback value should be one of the following: 0x33, 0x40, or 0x41. The range of ±1 DACCLK cycle indicates the default DAC latency uncertainty from power-on to power-on without turning on synchronization.

The recommended procedure for a serial port FIFO reset is as follows:

1. Configure the DAC in the desired interpolation mode (Register 0x28[1:0]).
2. Ensure that the DACCLK and DCI are running and stable at the clock inputs.
3. Program Register 0x23 to the customized value, if the desired value is not 0x40.
4. Request the FIFO level reset by setting Register 0x25[0] to 1.
5. Verify that the device acknowledges the request by setting Register 0x25[1] to 1.
6. Remove the request by setting Register 0x25[0] to 0.
7. Verify that the device drops the acknowledge signal by setting Register 0x25[1] to 0.
8. Read back Register 0x24 multiple times to verify that the actual FIFO level is set to the requested level and that the readback values are stable. By design, the readback is within ±1 DACCLK around the requested level.

FRAME INITIATED FIFO RESET

The frame input has two functions. One function is to indicate the beginning of a byte stream in the byte interface mode, as discussed in the Data Interface section. The other function is to initialize the FIFO level by asserting the frame signal high for at least the time interval required to load complete data to the I and Q DACs. This corresponds to one DCI period in word interface mode and two DCI periods in byte interface mode. Note that this requirement of the frame pulse length is longer than that of the frame signal when it serves only to assemble the byte stream. The device accepts either a continuous frame or a one shot frame signal.

In the continuous reset mode, the FIFO responds to every valid frame pulse and resets itself. In the one shot reset mode, the FIFO responds only to the first valid frame pulse after the FRAME_RESET_MODE bits (Register 0x22[1:0]) are set. Therefore, even with a continuous frame input, the FIFO resets one time only; this prevents the FIFO from toggling between the two states from periodic resets. The one shot frame reset mode is the default and the recommended mode.

The recommended procedure for a frame initiated FIFO reset is as follows:

1. Configure the DAC in the desired interpolation mode (Register 0x28[1:0]).
2. Ensure that the DACCLK and DCI are running and stable at the clock inputs.
3. Ensure that the DLL is locked (if using DLL mode) or the DCI clock is being sent properly (if using bypass mode). Program Register 0x23 to the customized value, if the desired value is not 0x40.
4. Configure the FRAME_RESET_MODE bits (Register 0x22, Bits [1:0]) to 00b.
5. Choose whether to use continuous or one shot mode by writing 0 or 1 to EN_CON_FRAME_RESET (Register 0x22, Bit 2).
6. Toggle the frame input from 0 to 1 and back to 0. The pulse width needs to be longer than the minimum requirement.
   a. If the frame input is a continuous clock, turn on the signal.
7. Read back Register 0x24 multiple times to verify that the actual FIFO level is set to the requested level and the readback values are stable. By design, the readback is within ±1 DACCLK around the requested level.

Monitoring the FIFO Status

The real-time FIFO status can be monitored from the SPI Register 0x24 and reflects the real-time FIFO depth after a FIFO reset. Without timing drifts in the system, this readback does not change from that which resulted from the FIFO reset. When there is a timing drift or other abnormal clocking situation, the FIFO level readback can change. However, as long as the FIFO does not overflow or underflow, there is no error in data transmission. Three status bits in Register 0x06, Bits[2:0], indicate if there are FIFO underflows, overflows, or similar situations. The status of the three bits can be latched and used to trigger hardware interrupts, $\overline{\text{IRQ1}}$ and $\overline{\text{IRQ2}}$. To enable latching and interrupts, configure the corresponding bits in Register 0x03 and Register 0x04.

DIGITAL DATAPATH

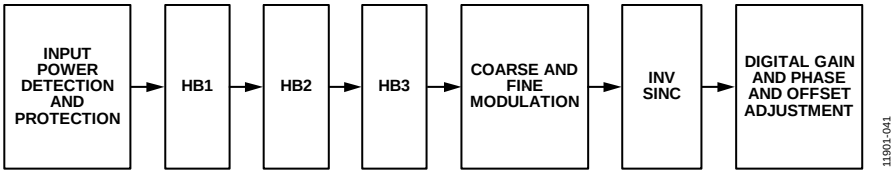


Figure 40. Block Diagram of Digital Datapath

The block diagram in Figure 40 shows the functionality of the digital datapath. The digital processing includes

- An input power detection block
- Three half-band interpolation filters
- A quadrature modulator consisting of a fine resolution NCO and an $f_S/4$ coarse modulation block
- An inverse sinc filter
- A gain and phase and offset adjustment block

The interpolation filters accept I and Q data streams and process them as two independent data streams, whereas the quadrature modulator and phase adjustment block accepts I and Q data streams as a quadrature data stream. Therefore, quadrature input data is required when digital modulation and phase adjustment functions are used.

INTERPOLATION FILTERS

The transmit path contains three interpolation filters. Each of the three interpolation filters provides a 2× increase in output data rate and a low-pass function. The half-band (HB) filters are cascaded to provide 4× or 8× interpolation ratios.

The AD9142A provides three interpolation modes (see Table 6). Each mode offers a different usable signal bandwidth in an operating mode. Which mode to select depends on the required signal bandwidth and the DAC update rate. Refer to Table 6 for the maximum speed and signal bandwidth of each interpolation mode.

The usable bandwidth is defined as the frequency band over which the filters have a pass-band ripple of less than ±0.001 dB and a stop band rejection of greater than 85 dB.

2× Interpolation Mode

Figure 41 and Figure 42 show the pass-band and all-band filter response for 2× mode. Note that the transition from the transition band to the stop band is much sharper than the transition from the pass band to the transition band. Therefore, when the desired output signal moves out of the defined pass band, the signal image, which is supposed to be suppressed by the stop band, grows faster than the droop of the signal itself due to the degraded pass-band flatness. In cases where the degraded image rejection is acceptable or can be compensated by the analog low-pass filter at the DAC output, it is possible to let the output signal extend beyond the specified usable signal bandwidth.

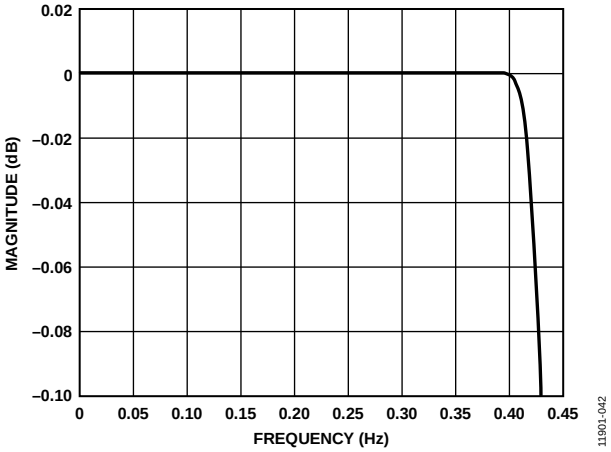


Figure 41. Pass-Band Detail of 2× Mode

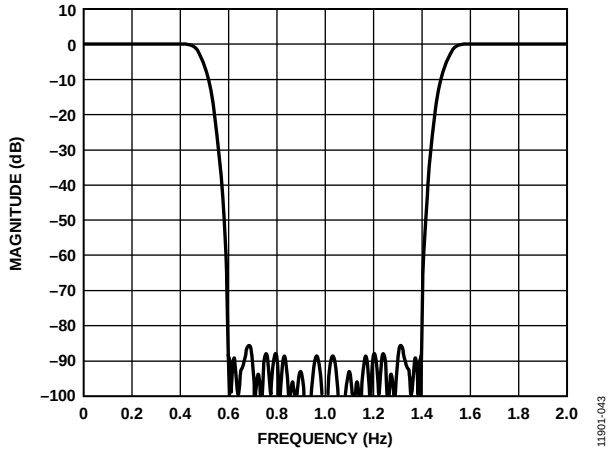


Figure 42. All-Band Response of 2× Mode

4× Interpolation Mode

Figure 43 and Figure 44 show the pass-band and all-band filter responses for 4× mode.

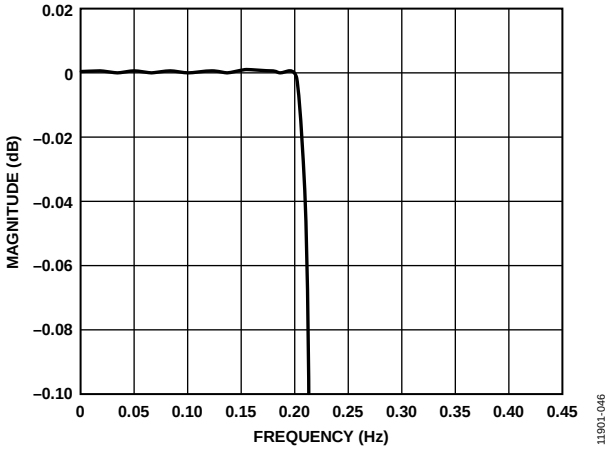


Figure 43. Pass-Band Detail of 4× Mode

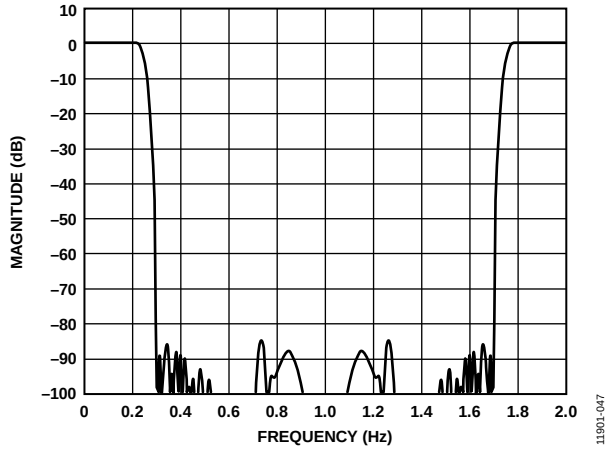


Figure 44. All-Band Response of 4× Mode

8× Interpolation Mode

Figure 45 and Figure 46 show the pass-band and all-band filter responses for 8× mode. The maximum DAC update rate is 1.6 GHz, and the maximum input data rate that is supported in this mode is 200 MHz (1.6 GHz/8).

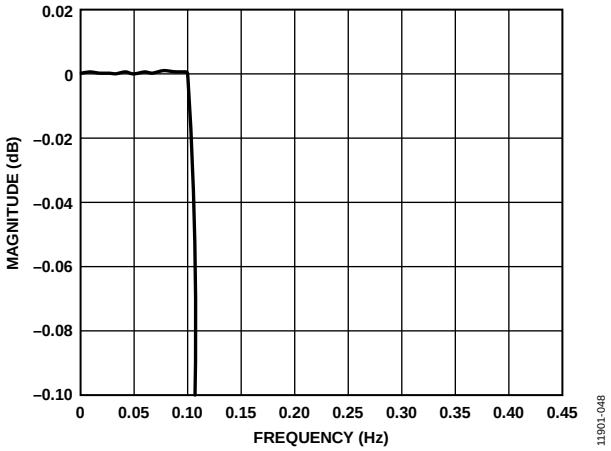


Figure 45. Pass-Band Detail of 8× Mode

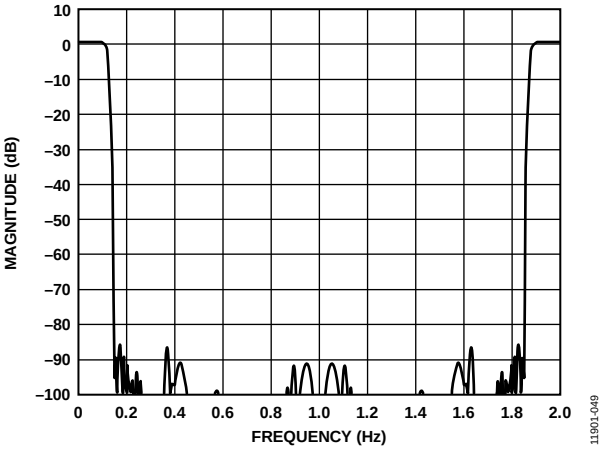


Figure 46. All-Band Response of 8× Mode

Table 18. Half-Band Filter 1 Coefficient

Lower Coefficient	Upper Coefficient	Integer Value
H(1)	H(55)	−4
H(2)	H(54)	0
H(3)	H(53)	+13
H(4)	H(52)	0
H(5)	H(51)	−32
H(6)	H(50)	0
H(7)	H(49)	+69
H(8)	H(48)	0
H(9)	H(47)	−134
H(10)	H(46)	0
H(11)	H(45)	+239
H(12)	H(44)	0
H(13)	H(43)	−401
H(14)	H(42)	0
H(15)	H(41)	+642
H(16)	H(40)	0
H(17)	H(39)	−994
H(18)	H(38)	0
H(19)	H(37)	+1512
H(20)	H(36)	0
H(21)	H(35)	−2307
H(22)	H(34)	0
H(23)	H(33)	+3665
H(24)	H(32)	0
H(25)	H(31)	−6638
H(26)	H(30)	0
H(27)	H(29)	+20,754
H(28)		+32,768

Table 19. Half-Band Filter 2 Coefficient

Lower Coefficient	Upper Coefficient	Integer Value
H(1)	H(23)	−2
H(2)	H(22)	0
H(3)	H(21)	+17
H(4)	H(20)	0
H(5)	H(19)	−75
H(6)	H(18)	0
H(7)	H(17)	+238
H(8)	H(16)	0
H(9)	H(15)	−660
H(10)	H(14)	0
H(11)	H(13)	+2530
H(12)		+4096

Table 20. Half-Band Filter 3 Coefficient

Lower Coefficient	Upper Coefficient	Integer Value
H(1)	H(11)	+29
H(2)	H(10)	0
H(3)	H(9)	−214
H(4)	H(8)	0
H(5)	H(7)	+1209
H(6)		+2048

DIGITAL MODULATION

The AD9142A provides two modes to modulate the baseband quadrature signal to the desired DAC output frequency.

- Coarse ($f_S/4$) modulation
- Fine (NCO) modulation

$f_S/4$ Modulation

The $f_S/4$ modulation is a convenient and low power modulation mode to translate the input baseband frequency to a fixed $f_S/4$ IF frequency, f_S being the DAC sampling rate. When modulation frequencies other than this frequency are required, the NCO modulation mode must be used.

NCO Modulation

The NCO modulation mode makes use of a numerically controlled oscillator (NCO), a phase shifter, and a complex modulator to provide a means for modulating the signal by a programmable carrier signal. A block diagram of the digital modulator is shown in Figure 47. The NCO modulation allows the DAC output signal to be placed anywhere in the output spectrum with very fine frequency resolution.

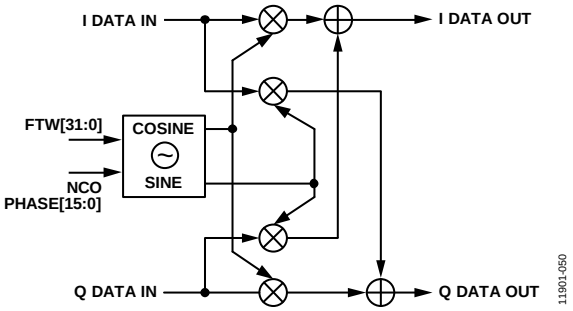


Figure 47. NCO Modulator Block Diagram

The NCO modulator mixes the carrier signal generated by the NCO with the I and Q signals. The NCO produces a quadrature carrier signal to translate the input signal to a new center frequency. A complex carrier signal is a pair of sinusoidal waveforms of the same frequency, offset 90° from each other. The frequency of the complex carrier signal is set via NCO_FTW3 to NCO_FTW0 in Register 0x31 through Register 0x34.

The NCO operating frequency, f_{NCO}, is always equal to f_{DAC}, the DACCLK frequency. The frequency of the complex carrier signal can be set from dc up to $\pm 0.5 \times f_{NCO}$.

The frequency tuning word (FTW) is in twos complement format. It can be calculated as

$$-\frac{f_{DAC}}{2} \le f_{CARRIER} \le \frac{f_{DAC}}{2}$$

$$FTW = \frac{f_{CARRIER}}{f_{DAC}} \times \left(2^{32}\right)\left(f_{CARRIER} \ge 0\right)$$

$$FTW = \left(1 - \frac{\left|f_{CARRIER}\right|}{f_{DAC}}\right) \times \left(2^{32}\right)\left(f_{CARRIER} < 0\right)$$

The generated quadrature carrier signal is mixed with the I and Q data. The quadrature products are then summed into the I and Q data paths, as shown in Figure 47.

Updating the Frequency Tuning Word

The frequency tuning word registers are not updated immediately upon writing, as are other configuration registers. Similar to FIFO reset, the NCO update can be triggered in two ways.

- SPI initiated update
- Frame initiated update

SPI Initiated Update

In the SPI initiated update method, the user simply toggles Register 0x30[0] (NCO_SPI_UPDATE_REQ) after configuring the NCO settings. The NCO is updated on the rising edge (from 0 to 1) in this bit. Register 0x30[1] (NCO_SPI_UPDATE_ACK) goes high when the NCO is updated. A falling edge (from 1 to 0) in Register 0x30[0] clears Bit 1 of Register 0x30 and prepares the NCO for the next update operation. This update method is recommended when there is no requirement to align the DAC output from multiple devices because SPI writes to multiple devices are asynchronous.

Frame Initiated Update

When the DAC output from multiple devices must be well aligned with NCO turned on, the frame initiated update is recommended. In this method, the NCOs from multiple devices are updated at the same time upon the rising edge of the frame signal. To use this update method, the FRAME_RESET_MODE (Register 0x22[1:0]) must be set in NCO only or FIFO and NCO, depending on whether a FIFO reset is needed at the same time. The second step is to ensure that the reset mode is in one shot mode (EN_CON_FRAME_RESET, Register 0x22[2] = 0). When this is completed, the NCO waits for a valid frame pulse and updates the FTW accordingly. The user can verify if the frame pulse is correctly received by reading Register 0x30[6] (NCO_FRAME_UPDATE_ACK) wherein a 1 indicates a complete update operation. See the FIFO Operation section for information to generate a valid frame pulse.

DATAPATH CONFIGURATION

Configuring the AD9142A datapath starts with the following four parameters:

- The application requirements of the input data rate
- The interpolation ratio
- The output signal center frequency
- The output signal bandwidth

Given these four parameters, the first step to configure the datapath is to verify that the device supports the desired input data rate, the DAC sampling rate, and the bandwidth requirements. After this verification, the modes of the interpolation filters can be chosen. If the output signal center frequency is different from the baseband input center frequency, additional frequency offset requirements are determined and applied with on-chip digital modulation.

DIGITAL QUADRATURE GAIN AND PHASE ADJUSTMENT

The digital quadrature gain and phase adjustment function enables compensation of the gain and phase imbalance of the I and Q paths caused by analog mismatches between DAC I/Q outputs, quadrature modulator I/Q baseband inputs, and DAC/modulator interface I/Q paths. The undesired imbalances cause unwanted sideband signal to appear at the quadrature modulator output with significant energy. Tuning the quadrature gain and phase adjust values optimizes image rejection in single sideband radios.

Quadrature Gain Adjustment

Ordinarily, the I and Q channels have the same gain or signal magnitude. The quadrature gain adjustment is used to balance the gain between the I and Q channels. The digital gain of the I and Q channels can be adjusted independently through two 6-bit registers, IDAC_GAIN_ADJ (Register 0x3F[5:0]) and QDAC_GAIN_ADJ (Register 0x40[5:0]). The range of the adjustment is [0, 2] or [−∞, 6 dB] with a step size of 2^{-5} (−30 dB). The default setting is 0x20, corresponding to a gain equal to 1 or 0 dB.

Quadrature Phase Adjustment

Under normal circumstances, I and Q channels have an angle of precisely 90° between them. The quadrature phase adjustment is used to change the angle between the I and Q channels. IQ_PHASE_ADJ_MSB and IQ_PHASE_ADJ_LSB (Register 0x37, Bits [7:0] and Register 0x38, Bits [4:0]) provide an adjustment range of ±14° with a resolution of 0.0035°. If the original angle is precisely 90°, setting IQ_PHASE_ADJ_MSB and IQ_PHASE_ADJ_LSB to 0x0FFF adds approximately 14° between I and QDAC outputs, creating an angle of 104° between the channels. Likewise, if the original angle is precisely 90°, setting IQ_PHASE_ADJ_MSB and IQ_PHASE_ADJ_LSB to 0x1000 adds approximately −14° between the I and QDAC outputs, creating an angle of 76° between the channels.

DC OFFSET ADJUSTMENT

The dc value of the I datapath and the Q datapath can be controlled independently by adjusting the values in the two IDAC dc offset 16-bit registers, IDAC_DC_OFFSET_LSB, IDAC_DC_OFFSET_MSB, QDAC_DC_OFFSET_LSB, and QDAC_DC_OFFSET_MSB (Register 0x3B through Register 0x3E). These values are added directly to the datapath values. Take care not to overrange the transmitted values.

As shown in Figure 48, the DAC offset current varies as a function of the I/QDAC dc offset values. Figure 48 shows the nominal current of the positive node of the DAC output, I_{OUTP}, when the digital inputs are fixed at midscale (0x0000, twos complement data format) and the DAC offset value is swept from 0x0000 to 0xFFFF. Because I_{OUTP} and I_{OUTN} are complementary current outputs, the sum of I_{OUTP} and I_{OUTN} is always 20 mA.

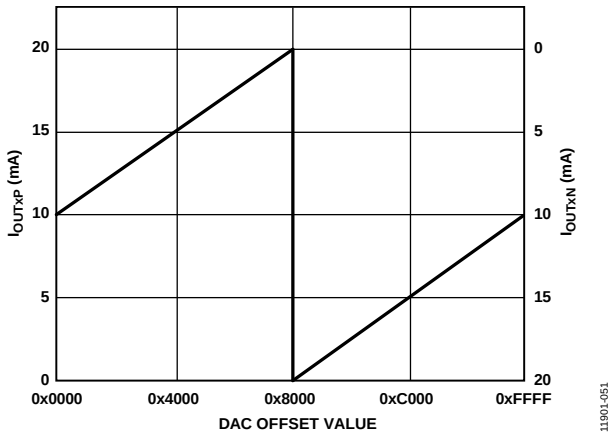


Figure 48. DAC Output Currents vs. DAC Offset Value

INVERSE SINC FILTER

The AD9142A provides a digital inverse sinc filter to compensate for the DAC roll-off over frequency. The inverse sinc ($sinc^{-1}$) filter is a seven-tap FIR filter. Figure 49 shows the frequency response of sin(x)/x roll-off, the inverse sinc filter, and their composite response. The composite response has less than ±0.05 dB pass-band ripple up to a frequency of 0.4 × f_{DAC}.

To provide the necessary peaking at the upper end of the pass band, the inverse sinc filter has an intrinsic insertion loss of about 3.8 dB. The loss of the digital gain can be offset by increasing the quadrature gain adjustment setting on both the I and Q data paths to minimize the impact on the output signal-to-noise ratio. However, care is needed to ensure that the additional digital gain does not cause signal saturation, especially at high output frequencies. The $sinc^{-1}$ filter is disabled by default; it can be enabled by setting the INVSINC_ENABLE bit to 1 in Register 0x27[7]).

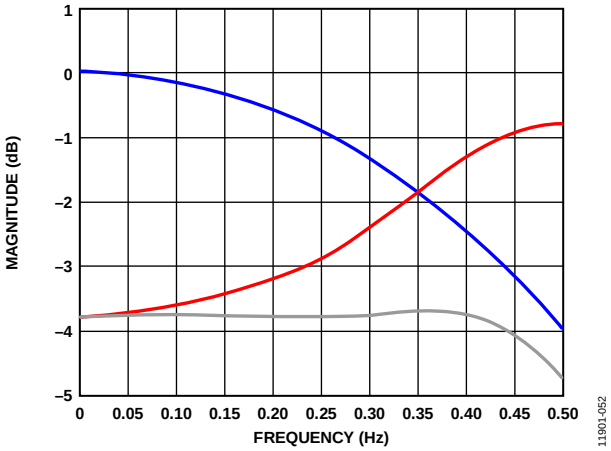


Figure 49. Responses of sin(x)/x Roll Off (Blue), the $Sinc^{-1}$ Filter (Red), and Composite of Both (Black)

Table 21. Inverse Sinc Filter

Lower Coefficient	Upper Coefficient	Integer Value
H(1)	H(7)	−1
H(2)	H(6)	+4
H(3)	H(5)	−16
H(4)		+192

INPUT SIGNAL POWER DETECTION AND PROTECTION

The input signal power detection and protection function detects the average power of the DAC input signal and prevents overrange signals from being passed to the next stage. An overrange DAC output signal can cause destructive breakdown on power sensitive devices, such as power amplifiers. The power detection and protection feature of the AD9142A detects overrange signals in the DAC. When an overrange signal is detected, the protection function either attenuates or mutes the signal to protect the downstream devices from abnormal power surges in the signal.

Figure 50 shows the block diagram of the power detection and protection function. The protection block is at the very last stage of the data path and the detection block uses a separate path from the data path. The design of the detection block guarantees that the worst-case latency of power detecting is shorter than that of the data path. This ensures that the protection circuit initiates before the overrange signal reaches the analog DAC core.

The sum of I^2 and Q^2 is calculated as a representation of the input signal power. Only the upper six MSBs, D[15:10], of data samples are used in the calculation; consequently, samples whose power is 36 dB below the full-scale peak power are not detected.

The calculated sample power numbers accumulate through a moving average filter. Its output is the average of the input signal power in a certain number of data clock cycles. The length of the filter is configurable through the SAMPLE_WINDOW_LENGTH (Register 0x2B[3:0]). To determine whether the input average power is over range, the device averages the power of the samples in the filter and compares the average power with a user defined threshold, THRESHOLD_LEVEL_REQUEST_LSB and THRESHOLD_LEVEL_REQUEST_MSB (Register 0x29[7:0] and Register 0x2A[4:0]). When the output of the averaging filter is larger than the threshold, the DAC output is either attenuated or muted.

The appropriate filter length and average power threshold for effective protection are application dependent. It is recommended that experiments be performed with real-world vectors to determine the values of these parameters.

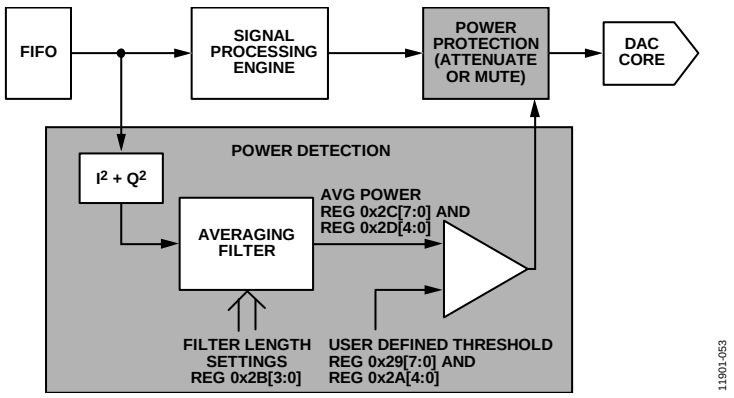


Figure 50. Block Diagram of Input Signal Power Detection and Protection Function

TRANSMIT ENABLE FUNCTION

The transmit enable (TXEN) function provides the user with a hardware switch of the DAC output. The function accepts a CMOS signal via Pin 6 (TXEN). When this signal is detected high, the transmit path is enabled and the DAC transmits the data normally. When this signal is detected low, one of the three actions related to the DAC output is triggered. This can be configured in Register 0x43.

1. The DAC output is gradually attenuated from full scale gain to 0. The attenuation step size is set in Register 0x42[5:0].
2. The DAC is put in sleep mode and the output current is turned off. Other areas of the DAC are still running in this mode.
3. The DAC is put in power-down mode. In this mode, not only the DAC output current is turned off but the rest of the DAC is powered down. This minimizes the power consumption of the DAC when the data is not transmitting but it takes a bit longer than the first two modes to start to retransmit data due to the device power-up time.

The TXEN function also provides a gain ramp-up function that lets the user turn on the DAC output gradually when the TXEN signal switches from low to high. The ramp-up gain step can be configured using Register 0x41[5:0].

DIGITAL FUNCTION CONFIGURATION

Each of the digital gain and phase adjust functions and the inverse sinc filter can be enabled and adjusted independently. The pipeline latencies these blocks add into the data path are different between enabled and disabled. If fixed DAC pipeline latency is desired during operation, leave these functions always on or always off after initial configuration.

The digital dc adjust function is always on. The default value is 0; that is, there is no additional dc offset. The pipeline latency that this block adds is a constant, no matter the value of the dc offset.

There is also a latency difference between using and not using the input signal power detection and protection function. Therefore, to keep the overall latency fixed, leave this function always on or always off after the initial configuration.

MULTIDEVICE SYNCHRONIZATION AND FIXED LATENCY

A DAC introduces a variation of pipeline latency to a system. The latency variation causes the phase of a DAC output to vary from power-on to power-on. Therefore, the output from different DAC devices may not be perfectly aligned even with well aligned clocks and digital inputs. The skew between multiple DAC outputs varies from power-on to power-on.

In applications such as transmit diversity or digital predistortion, where deterministic latency is desired, the variation of the pipeline latency must be minimized. Deterministic latency in this data sheet is defined as a fixed time delay from the digital input to the analog output in a DAC from power-on to power-on. Multiple DAC devices are considered synchronized to each other when each DAC in this group has the same constant latency from power-on to power-on. Three conditions must be identical in all of the ready-to-sync devices before these devices are considered synchronized:

- The phase of DAC internal clocks
- The FIFO level
- The alignment of the input data

VERY SMALL INHERENT LATENCY VARIATION

The innovative architecture of the AD9142A minimizes the inherent latency variation. The worst-case variation in the AD9142A is two DAC clock cycles. For example, in the case of a 1.5 GHz sample rate, the variation is less than 1.4 ns in any scenario. Therefore, without turning on the synchronization engine, the DAC outputs from multiple AD9142A devices are guaranteed to be aligned within two DAC clock cycles, regardless of the timing between the DCI and the DACCLK. No additional clocks are required to achieve this accuracy. The user must reset the FIFO in each DAC device through the SPI at startup. Therefore, the AD9142A can decrease the complexity of system design in multitransmit channel applications.

Note the alignment of the DCI signals in the design. The DCI is used as a reference in the AD9142A design to align the FIFO and the phase of internal clocks in multiple devices. The achieved DAC output alignment depends on how well the DCI signals are aligned at the input of each device. The following equation is the expression of the worst-case DAC output alignment accuracy in the case of DCI signal mismatches.

$$t_{SK\,(OUT)} = t_{SK\,(DCI)} + 2/f_{DAC}$$

where:

$t_{SK\,(OUT)}$ is the worst-case skew between the DAC output from two AD9142A devices.

$t_{SK\,(DCI)}$ is the skew between two DCI signals at the DCI input of the two AD9142A devices.

f_{DAC} is the DACCLK frequency.

The better the alignment of the DCI signals, the smaller is the overall skew between two DAC outputs.

FURTHER REDUCING THE LATENCY VARIATION

For applications that require finer synchronization accuracy (DAC latency variation < 2 DAC clock cycles), the AD9142A has a provision for enabling multiple devices to be synchronized to each other within a single DAC clock cycle.

To further reduce the latency variation in the DAC, the synchronization machine needs to be turned on and two external clocks (frame and sync) need to be generated in the system and fed to all the DAC devices.

Set Up and Hold Timing Requirement

The sync clock (f_{SYNC}) serves as a reference clock in the system to reset the clock generation circuitry in multiple AD9142A devices simultaneously. Inside the DAC, the sync clock is sampled by the DACCLK to generate a reference point for aligning the internal clocks, so there is a setup and hold timing requirement between the sync clock and the DAC clock.

If the user adopts the continuous frame reset mode, that is, the FIFO and sync engine periodically reset, the timing requirements between the sync clock and the DAC clock must be met. Otherwise, the device can lose lock and corrupt the output. In the one shot frame reset mode, it is still recommended that this timing be met at the time when the sync routine is run because not meeting the timing can degrade the sync alignment accuracy by one DAC cycle, as shown in Table 22.

For users who want to synchronize the device in a one-shot manner and continue to monitor the synchronization status, the AD9142A provides a sync monitoring mode. It provides a continuous sync and frame clock to synchronize the part once and ignore the clock cycles after the first valid frame pulse is detected. In this way, the user can monitor the sync status without periodically resynchronizing the device; to engage the sync monitoring mode, set Register 0x22[1:0] (FRAME_RESET_MODE) to 11b.

Table 22. Sync Clock and DAC Clock Setup and Hold Times

Falling Edge Sync Timing (default)	Max (ps)
t_S (ns)	324
t_H (ns)[1]	−92
$\lvert t_S + t_H \rvert$ (ns)	232

[1] The negative sign indicates the direction of the setup time. The setup time is defined as positive when it is on the left side of the clock edge and negative when it is on the right side of the clock edge.

SYNCHRONIZATION IMPLEMENTATION

The AD9142A lets the user choose either the rising or falling edge of the DAC clock to sample the sync clock, which makes it easier to meet the timing requirements. Ensure that the sync clock, f_{SYNC}, is 1/8 × f_{DATA} or slower by a factor of 2n, n being an integer (1, 2, 3…). Note that there is a limit on how slow the sync clock can be received because of the ac coupling nature of the sync clock receiver. Choose an appropriate value of the ac coupling capacitors to ensure that the signal swing meets the data sheet specification, as listed in Table 2.

The frame clock resets the FIFO in multiple AD9142A devices. The frame can be either a one shot or continuous clock. In either case, the pulse width of the frame must be longer than one DCI cycle in the word interface mode and two DCI cycles in the byte interface mode. When the frame is a continuous clock, f_{FRAME}, ensure that it is 1/8 × f_{DATA} or slower by a factor of 2n, n being an integer (1, 2, 3…). Table 23 lists the requirements of the frame clock in various conditions. Byte interface mode is not supported when the frame signal is used in synchronization.

Table 23. Frame Clock Speed and Pulse Width Requirement

Sync Clock	Maximum Speed	Minimum Pulse Width
One Shot	N/A[1]	For both one shot and continuous sync clocks, word interface mode = one DCI cycle and byte interface mode = two DCI cycles.
Continuous	f_{DATA}/8	

[1] N/A means not applicable.

SYNCHRONIZATION PROCEDURES

When the sync accuracy of an application is less precise than two DAC clock cycles, it is recommended to turn off the synchronization machine because there are no additional steps required, other than the regular start-up procedure sequence.

For applications that require more precise sync accuracy than two DAC clock cycles, it is recommended that the procedure in the Synchronization Procedure for PLL Off or Synchronization Procedure for PLL On sections be followed to set up the system and configure the device. For more information about the details of the synchronization scheme in the AD9142A and using the synchronization function to correct system skews and drifts, see the DAC Latency and System Skews section.

Synchronization Procedure for PLL Off

1. Configure the DAC interpolation mode and, if NCO is used, configure the NCO FTW.
2. Set up the DAC data interface according to the procedure outlined in the Data Interface section and verify that the DLL is locked.
3. Choose the appropriate mode in the FRAME_RESET_MODE bits (Register 0x22[1:0]).
   a. If NCO is not used, choose FIFO only mode.
   b. If NCO is used, it must be synchronized. FIFO and NCO mode can then be used.
4. Configure Bit 2 in Register 0x22 for continuous or one shot reset mode. One shot reset mode is recommended.
5. Ensure that the DACCLK, DCI, and sync clock to all of the AD9142A devices are running and stable.
6. Enable the sync engine by writing 1 to Register 0x21[0].
7. Send a valid frame pulse(s) to all of the AD9142A devices.
8. Verify that the frame pulse is received by each device by reading back Register 0x22[3]. All the readback values are 1. At this point, the devices should be synchronized.

Synchronization Procedure for PLL On

Note that, because the sync clock and PLL reference clock share the same clock and the maximum sync clock rate is f_{DATA}/8, the same limit also applies to the reference clock. Therefore, only 2× interpolation is supported for synchronization with PLL on.

1. Set up the PLL according to the procedure in the Clock Multiplication section and ensure that the PLL is locked.
2. Configure the DAC interpolation mode and, if NCO is used, configure the NCO FTW.
3. Set up the DAC data interface according to the procedure in the Data Interface section and verify that the DLL is locked.
4. Choose the appropriate mode in the FRAME_RESET_MODE bits (Register 0x22[1:0])
   a. If NCO is not used, choose the FIFO only mode.
   b. If NCO is used, it must be synchronized. FIFO and NCO mode can then be used.
5. Configure Bit 2 in Register 0x22 for continuous or one shot reset mode. One shot reset mode is recommended.
6. Ensure that DACCLK, DCI, and sync clock to all of the AD9142A devices are running.
7. Enable the sync engine by writing 1 to Register 0x21[0].
8. Send a valid frame pulse(s) to all of the AD9142A devices.
9. Verify that the frame pulse is received by each device by reading back Register 0x22[3]. All the readback values are 1. At this point, the devices should be synchronized.

INTERRUPT REQUEST OPERATION

The AD9142A provides an interrupt request output signal on Pin 50 and Pin 51 ($\overline{IRQ2}$ and $\overline{IRQ1}$, respectively) that can be used to notify an external host processor of significant device events. Upon assertion of the interrupt, query the device to determine the precise event that occurred. The $\overline{IRQ1}$ pin is an open-drain, active low output. Pull the $\overline{IRQ1}$ pin high external to the device. This pin can be tied to the interrupt pins of other devices with open-drain outputs to wire-OR these pins together.

Ten event flags provide visibility into the device. These flags are located in the two event flag registers, Register 0x05 and Register 0x06. The behavior of each event flag is independently selected in the interrupt enable registers, Register 0x03 and Register 0x04. When the flag interrupt enable is active, the event flag latches and triggers an external interrupt. When the flag interrupt is disabled, the event flag monitors the source signal, but the $\overline{IRQ1}$ and $\overline{IRQ2}$ pins remain inactive.

INTERRUPT WORKING MECHANISM

Figure 51 shows the interrupt related circuitry and how the event flag signals propagate to the $\overline{IRQx}$ output. The INTERRUPT_ENABLE signal represents one bit from the interrupt enable register. The EVENT_FLAG_SOURCE signal represents one bit from the event flag register. The EVENT_FLAG_SOURCE signal represents one of the device signals that can be monitored, such as the PLL_LOCK signal from the PLL phase detector or the FIFO_WARNING_1 signal from the FIFO controller.

When an interrupt enable bit is set high, the corresponding event flag bit reflects a positively tripped version of the EVENT_FLAG_SOURCE signal; that is, the event flag bit is latched on the rising edge of the EVENT_FLAG_SOURCE signal. This signal also asserts the external $\overline{IRQ}$ pins.

When an interrupt enable bit is set low, the event flag bit reflects the present status of the EVENT_FLAG_SOURCE signal, and the event flag has no effect on the external $\overline{IRQ}$ pins.

Clear the latched version of an event flag (the INTERRUPT_SOURCE signal) in one of two ways. The recommended method is by writing 1 to the corresponding event flag bit. The second method is to use a hardware or software reset to clear the INTERRUPT_SOURCE signal.

The $\overline{IRQ2}$ circuitry works in the same way as the $\overline{IRQ1}$ circuitry. Any one or multiple event flags can be enabled to trigger the $\overline{IRQ1}$ and $\overline{IRQ2}$ pins. The user can select one or both hardware interrupt pins for the enabled event flags. Register 0x07 and Register 0x08 determine the pin to which each event flag is routed. Set Register 0x07 and Register 0x08 to 0 for $\overline{IRQ1}$ and set these registers to 1 for $\overline{IRQ2}$.

INTERRUPT SERVICE ROUTINE

Interrupt request management starts by selecting the set of event flags that require host intervention or monitoring. Enable the events that require host action so that the host is notified when they occur. For events requiring host intervention upon $\overline{IRQx}$ activation, run the following routine to clear an interrupt request:

1. Read the status of the event flag bits that are being monitored.
2. Set the interrupt enable bit low so that the unlatched EVENT_FLAG_SOURCE signal can be monitored directly.
3. Perform any actions that may be required to clear the EVENT_FLAG_SOURCE signal. In many cases, no specific actions may be required.
4. Read the event flag to verify that the actions taken have cleared the EVENT_FLAG_SOURCE signal.
5. Clear the interrupt by writing 1 to the event flag bit.
6. Set the interrupt enable bits of the events to be monitored.

Note that some EVENT_FLAG_SOURCE signals are latched signals. These signals are cleared by writing to the corresponding event flag bit. For more information about each of the event flags, see the Device Configuration Register Map section.

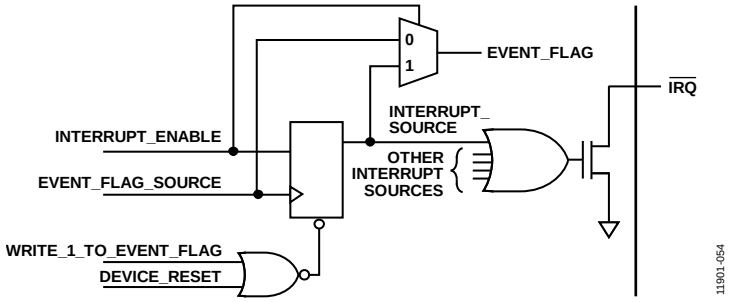


Figure 51. Simplified Schematic of $\overline{IRQ}$ Circuitry

TEMPERATURE SENSOR

The AD9142A has a diode-based temperature sensor for measuring the temperature of the die. The temperature reading is accessed using Register 0x1D and Register 0x1E. The temperature of the die can be calculated as

$$T_{DIE} = \frac{(DieTemp[15:0] - 41,237)}{106}$$

where T_{DIE} is the die temperature in degrees Celsius.

The temperature accuracy is ±7°C typical over the +85°C to −40°C range with one point temperature calibration against a known temperature. A typical plot of the die temperature code readback vs. die temperature is shown in Figure 52.

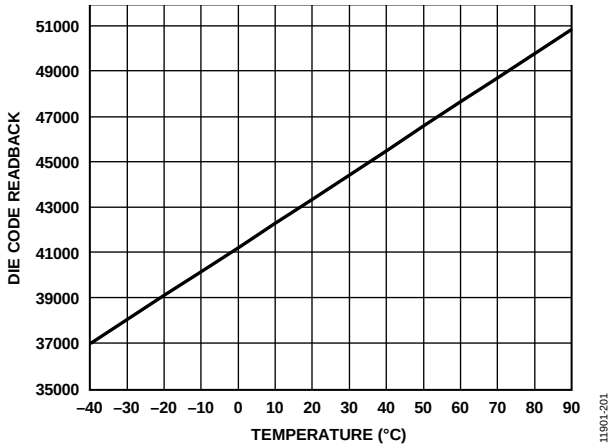


Figure 52. Die Temperature Code Readback vs. Die Temperature

Estimates of the ambient temperature can be made if the power dissipation of the device is known. For example, if the device power dissipation is 800 mW and the measured die temperature is 50°C, then the ambient temperature can be calculated as

$$T_A = T_{DIE} - P_D \times \theta_{JA} = 50 - 0.8 \times 20.7 = 33.4°C$$

where:
T_A is the ambient temperature in degrees Celsius.
T_{DIE} is the die temperature in degrees Celsius.
P_D is power consumption of the device.
θ_{JA} is the thermal resistance from junction to ambient of the AD9142A as shown in Table 8.

To use the temperature sensor, it must be enabled by setting Register 0x1C[0] to 1. In addition, to obtain accurate readings, set the die temperature control register (Register 0x1C) to 0x03.

DAC INPUT CLOCK CONFIGURATIONS

The AD9142A DAC sample clock (DACCLK) can be sourced directly or by clock multiplying. Clock multiplying employs the on-chip PLL that accepts a reference clock operating at a submultiple of the desired DACCLK rate. The PLL then multiplies the reference clock up to the desired DACCLK frequency, which can then be used to generate all of the internal clocks required by the DAC. The clock multiplier provides a high quality clock that meets the performance requirements of most applications. Using the on-chip clock multiplier removes the burden of generating and distributing the high speed DACCLK.

The second mode bypasses the clock multiplier circuitry and lets DACCLK be sourced directly to the DAC core. This mode lets the user source a very high quality clock directly to the DAC core.

DRIVING THE DACCLK AND REFCLK INPUTS

The DACCLKx and REFx/SYNCx differential inputs share similar clock receiver input circuitry. Figure 53 shows a simplified circuit diagram of the input. The on-chip clock receiver has a differential input impedance of about 10 kΩ. It is self biased to a common-mode voltage of about 1.25 V. The inputs can be driven by differential PECL or LVDS drivers with ac coupling between the clock source and the receiver.

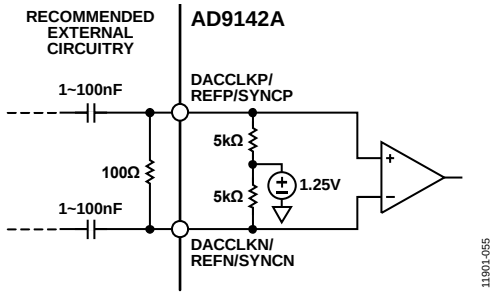


Figure 53. Clock Receiver Input Simplified Equivalent Circuit

The minimum input drive level to the differential clock input is 100 mV p-p differential. The optimal performance is achieved when the clock input signal is between 800 mV p-p differential and 1.6 V p-p differential. Whether using the on-chip clock multiplier or sourcing the DACCLK directly, the input clock signal to the device must have low jitter and fast edge rates to optimize the DAC noise performance.

DIRECT CLOCKING

Direct clocking with a low noise clock produces the lowest noise spectral density at the DAC outputs. To select the differential CLK inputs as the source for the DAC sampling clock, set the PLL enable bit (Register 0x12[7]) to 0. This powers down the internal PLL clock multiplier and selects the input from the DACCLKP and DACCLKN pins as the source for the internal DAC sampling clock. The REFCLKx input can remain floating.

The device also has clock duty cycle correction circuitry and differential input level correction circuitry. Enabling these circuits can provide improved performance in some cases. The control bits for these functions are in Register 0x10 and Register 0x11.

CLOCK MULTIPLICATION

The on-chip PLL clock multiplier circuit generates the DAC sample rate clock from a lower frequency reference clock. When the PLL enable bit (Register 0x12[7]) is set to 1, the clock multiplication circuit generates the DAC sampling clock from the lower rate REFx/SYNCx input and the DACCLKx input is left floating. The functional diagram of the clock multiplier is shown in Figure 54.

The clock multiplier circuit operates such that the VCO outputs a frequency, f_{VCO}, equal to the REFx/SYNCx input signal frequency multiplied by N1 × N0. N1 is the divide ratio of the loop divider; N0 is the divide ratio of the VCO divider.

$$f_{VCO} = f_{REFCLK} \times (N1 \times N0)$$

The DAC sample clock frequency, f_{DACCLK}, is equal to

$$f_{DACCLK} = f_{REFCLK} \times N1$$

The output frequency of the VCO must be chosen to keep f_{VCO} in the optimal operating range of 1.03 GHz to 2.07 GHz. It is important to select a frequency of the reference clock and values of N1 and N0 so that the desired DACCLK frequency can be synthesized and the VCO output frequency is in the correct range.

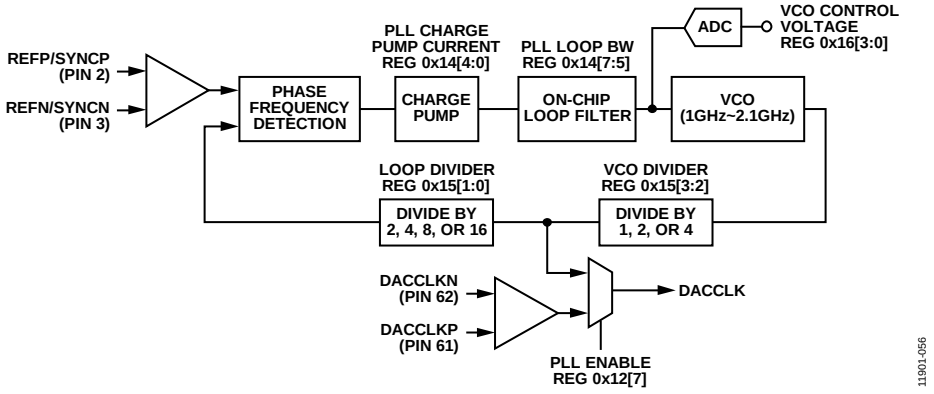


Figure 54. PLL Clock Multiplication Circuit

PLL SETTINGS

The PLL circuitry requires three settings to be programmed to their nominal values. The PLL values shown in Table 24 are the recommended settings for these parameters.

Table 24. PLL Settings

PLL SPI Control Register	Register Address	Optimal Setting (Binary)
PLL Loop Bandwidth	0x14[7:5]	111
PLL Charge Pump Current	0x14[4:0]	00111
PLL Cross Point Control Enable	0x15[4]	0

CONFIGURING THE VCO TUNING BAND

The PLL VCO has a valid operating range from approximately 1.03 GHz to 2.07 GHz covered in 64 overlapping frequency bands. For any desired VCO output frequency, there may be several valid PLL band select values. The frequency bands of a typical device are shown in Figure 55. Device-to-device variations and operating temperature affect the actual band frequency range. Therefore, it is required that the optimal PLL band select value be determined for each individual device.

AUTOMATIC VCO BAND SELECT

The device has an automatic VCO band select feature on chip. Using the automatic VCO band select feature is a simple and reliable method of configuring the VCO frequency band. This feature is enabled by starting the PLL in manual mode, and then placing the PLL in autoband select mode by setting Register 0x12 to a value of 0xC0 and then to a value of 0x80. When these values are written, the device executes an automated routine that determines the optimal VCO band setting for the device.

The setting selected by the device ensures that the PLL remains locked over the full −40°C to +85°C operating temperature range of the device without further adjustment. The PLL remains locked over the full temperature range even if the temperature during initialization is at one of the temperature extremes.

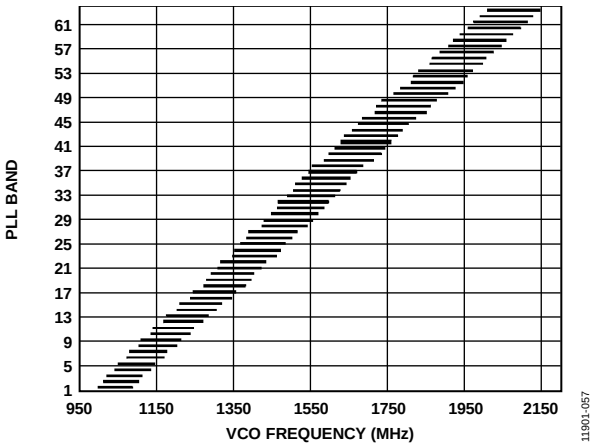


Figure 55. PLL Lock Range for a Typical Device

MANUAL VCO BAND SELECT

The device includes a manual band select mode (PLL auto manual enable, Register 0x12[6] = 1) that lets the user select the VCO tuning band. In manual mode, the VCO band is set directly with the value written to the manual VCO band bits (Register 0x12[5:0]).

PLL ENABLE SEQUENCE

To enable the PLL in automatic or manual mode properly, the following sequence must be followed:

Automatic Mode Sequence

4. Configure the loop divider and the VCO divider registers for the desired divide ratios.
5. Set 00111b to PLL charge pump current and 111b to PLL loop bandwidth for the best performance. Register 0x14 = 0xE7 (default).
6. Set the PLL mode to manual using Register 0x12[6] = 1.
7. Enable the PLL using Register 0x12[7] = 1.
8. Set the PLL mode to automatic using Register 0x12[6] = 0.

Manual Mode

1. Configure the loop divider and the VCO divider registers for the desired divide ratios.
2. Set 00111b to PLL charge pump current and 111 to PLL loop bandwidth for the best performance. Register 0x14 = 0xE7 (default).
3. Select the desired band using Register 0x12[5:0].
4. Set the PLL mode to manual using Register 0x12[6] = 1.
5. Enable the PLL using Register 0x12[7] = 1.

ANALOG OUTPUTS

TRANSMIT DAC OPERATION

Figure 56 shows a simplified block diagram of the transmit path DACs. The DAC core consists of a current source array, a switch core, digital control logic, and full-scale output current control. The DAC full-scale output current (I_{OUTFS}) is nominally 20 mA. The output currents from the IOUT1P/IOUT2P and IOUT1N/IOUT2N pins are complementary, meaning that the sum of the two currents always equals the full-scale current of the DAC. The digital input code to the DAC determines the effective differential current delivered to the load.

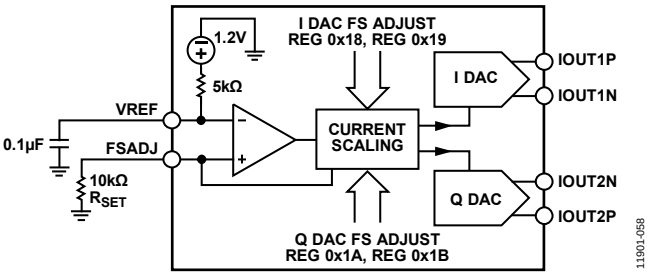


Figure 56. Simplified Block Diagram of DAC Core

The DAC has a 1.2 V band gap reference with an output impedance of 5 kΩ. The reference output voltage appears on the VREF pin. When using the internal reference, decouple the VREF pin to AVSS with a 0.1 µF capacitor. Use the internal reference only for external circuits that draw dc currents of 2 µA or less. For dynamic loads or static loads greater than 2 µA, buffer the VREF pin. If desired, the internal reference can be overdriven by applying an external reference (from 1.10 V to 1.30 V) to the VREF pin.

A 10 kΩ external resistor, R_{SET}, must be connected from the FSADJ pin to AVSS. This resistor, together with the reference control amplifier, sets up the correct internal bias currents for the DAC. Because the full-scale current is inversely proportional to this resistor, the tolerance of R_{SET} is reflected in the full-scale output amplitude.

The full-scale current equation, where the DAC gain is individually set for the Q and I DACs in Register 0x40 and Register 0x44, respectively, is as follows:

$$I_{FS} = \frac{V_{REF}}{R_{SET}} \times \left(72 + \left(\frac{3}{16} \times DAC\ gain\right)\right)$$

For nominal values of V_{REF} (1.2 V), R_{SET} (10 kΩ), and DAC gain (512), the full-scale current of the DAC is typically 20 mA. The DAC full-scale current can be adjusted from 8.64 mA to 31.68 mA by setting the DAC gain parameter, as shown in Figure 57.

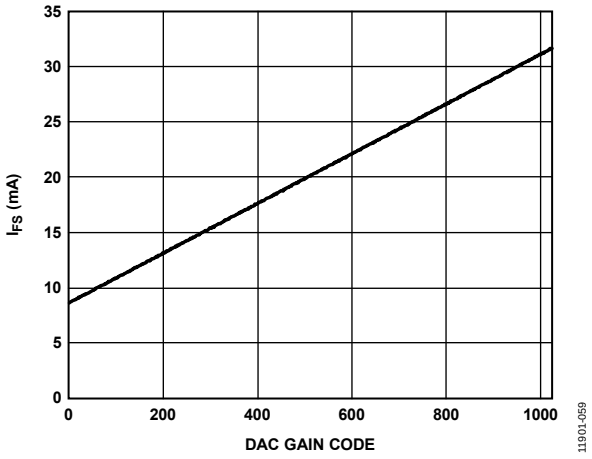


Figure 57. DAC Full-Scale Current vs. DAC Gain Code

Transmit DAC Transfer Function

The output currents from the IOUT1P/IOUT2P and IOUT1N/IOUT2N pins are complementary, meaning that the sum of the two currents always equals the full-scale current of the DAC. The digital input code to the DAC determines the effective differential current delivered to the load. IOUT1P/IOUT2P provide maximum output current when all bits are high. The output currents vs. DACCODE for the DAC outputs is expressed as

$$I_{OUTP} = \left[\frac{DACCODE}{2^N}\right] \times I_{OUTFS} \quad (1)$$

$$I_{OUTN} = I_{OUTFS} - I_{OUTP} \quad (2)$$

where $DACCODE = 0$ to $2^N - 1$.

Transmit DAC Output Configurations

The optimum noise and distortion performance of the AD9142A is realized when it is configured for differential operation. The common-mode rejection of a transformer or differential amplifier significantly reduces the common-mode error sources of the DAC outputs. These common-mode error sources include even-order distortion products and noise. The enhancement in distortion performance becomes more significant as the frequency content of the reconstructed waveform increases and/or its amplitude increases. This is due to the first-order cancellation of various dynamic common-mode distortion mechanisms, digital feedthrough, and noise.

Figure 58 shows the most basic DAC output circuitry. A pair of resistors, R_O, converts each of the complementary output currents to a differential voltage output, V_{OUT}. Because the current outputs of the DAC are high impedance, the differential driving point impedance of the DAC outputs, R_{OUT}, is equal to $2 \times R_O$. See Figure 59 for the output voltage waveforms.

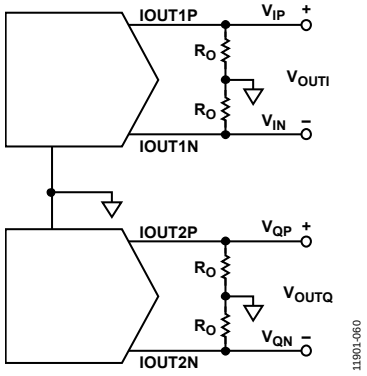


Figure 58. Basic Transmit DAC Output Circuit

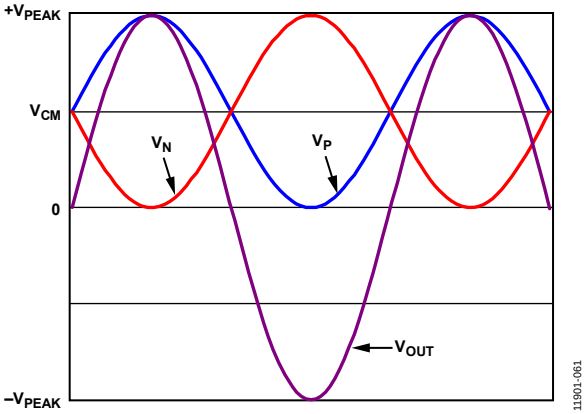


Figure 59. Output Voltage Waveforms

The common-mode signal voltage, V_{CM}, is calculated as

$$V_{CM} = \frac{I_{FS}}{2} \times R_O$$

The peak output voltage, V_{PEAK}, is calculated as

$$V_{PEAK} = I_{FS} \times R_O$$

In this circuit configuration, the single-ended peak voltage is the same as the peak differential output voltage.

INTERFACING TO MODULATORS

The AD9142A interfaces to the ADL537x family of modulators with a minimal number of components. An example of the recommended interface circuitry is shown in Figure 60.

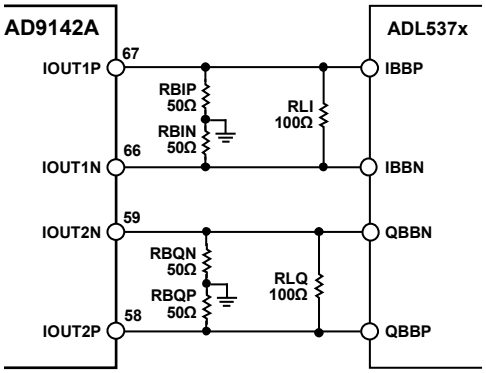


Figure 60. Typical Interface Circuitry Between the AD9142A and the ADL537x Family of Modulators

The baseband inputs of the ADL537x family require a dc bias of 500 mV. The nominal midscale output current on each output of the DAC is 10 mA (one-half the full-scale current). Therefore, a single 50 Ω resistor to ground from each of the DAC outputs results in the desired 500 mV dc common-mode bias for the inputs to the ADL537x. The addition of the load resistor in parallel with the modulator inputs reduces the signal level. The peak-to-peak voltage swing of the transmitted signal is

$$V_{SIGNAL} = I_{FS} \times \frac{(2 \times R_B \times R_L)}{(2 \times R_B + R_L)}$$

Baseband Filter Implementation

Most applications require a baseband anti-imaging filter between the DAC and the modulator to filter out Nyquist images and broadband DAC noise. The filter can be inserted between the I-V resistors at the DAC output and the signal level setting resistor across the modulator input. This configuration establishes the input and output impedances for the filter.

Figure 61 shows a fifth-order, low-pass filter. A common-mode choke is placed between the I-V resistors and the remainder of the filter to remove the common-mode signal produced by the DAC and to prevent the common-mode signal from being converted to a differential signal, which can appear as unwanted spurious signals in the output spectrum. Splitting the first filter capacitor into two and grounding the center point creates a common-mode low-pass filter, which provides additional common-mode rejection of high frequency signals. A purely differential filter can pass common-mode signals.

For more details about interfacing the AD9142A DAC to an IQ modulator, refer to the Circuits from the Lab™ Circuit Note CN-0205, *Interfacing the ADL5375 I/Q Modulator to the AD9122 Dual Channel, 1.2 GSPS High Speed DAC* on the Analog Devices website.

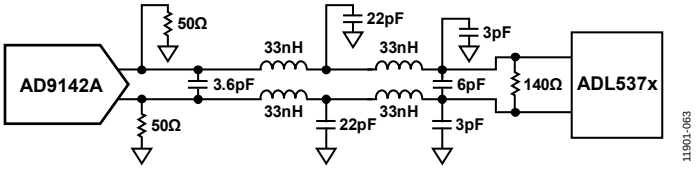


Figure 61. DAC Modulator Interface with Fifth-Order, Low-Pass Filter

REDUCING LO LEAKAGE AND UNWANTED SIDEBANDS

Analog quadrature modulators can introduce unwanted signals at the local oscillator (LO) frequency due to dc offset voltages in the I and Q baseband inputs, as well as feedthrough paths from the LO input to the output. The LO feedthrough can be nulled by applying the correct dc offset voltages at the DAC output using the digital dc offset adjustments (Register 0x3B through Register 0x3E).

Effective sideband suppression requires both gain and phase matching of the I and Q signals. The I/Q phase adjust registers (Register 0x37 and Register 0x38) and the DAC FS adjust registers (Register 0x18 through Register 0x1B) can be used to calibrate the I and Q transmit paths to optimize sideband suppression.

For more information about suppressing LO leakage and sideband image, refer to the AN-1039 Application Note, *Correcting Imperfections in IQ Modulators to Improve RF Signal Fidelity* and the AN-1100 Application Note, *Wireless Transmitter IQ Balance and Sideband Suppression* from the Analog Devices website.

EXAMPLE START-UP ROUTINE

To ensure reliable startup of the AD9142A, certain sequences must be followed.

DEVICE CONFIGURATION AND START-UP SEQUENCE 1

1. Set f_{DCI} = 375 MHz, f_{OUT} = 250 MHz, and interpolation to 4×.
2. Disable the PLL.
3. Enable fine NCO and the inverse sinc filter.
4. Use the DLL-based interface mode with DLL phase offset = 0.

Derived NCO Settings

The following NCO settings can be derived from the device configuration:

- f_{DAC} = 375 × 4 = 1500 MHz.
- $f_{CARRIER}$ = f_{OUT} = 250 MHz.
- FTW = $f_{CARRIER}/f_{DAC} \times 2^{32}$ = 0x2AAAAAAA.

Start-Up Sequence 1

1. Power up the device (no specific power supply sequence is required).
2. Apply stable DAC clock.
3. Apply stable DCI clock.
4. Feed stable input data.
5. Issue hardware reset (optional).

```
/* Device configuration register write sequence */
0x00 → 0x20 /* Issue software reset */
0x20 → 0x01 /* Device startup configuration */

/* Configure data interface */
0x5E → 0xFE /* Turn off LSB delay cell */
0x0A → 0xC0 /* Enable the DLL and duty cycle correction. Set DLL phase offset to 0 */
Read 0x0E[7:4] /* Expect 1000b if the DLL is locked */

/* Configure interpolation filter */
0x28 → 0x02 /* 4× interpolation */

/* Reset FIFO */
0x25 → 0x01
Read 0x25[1] /* Expect 1b if the FIFO reset is complete */
Read 0x24 /* The readback should be one of the three values: 0x33, 0x40, or 0x41 */

/* Configure NCO */
0x27→ 0x40 /* Enable NCO */
0x31 → 0xAA
0x32 → 0xAA
0x33 → 0xAA
0x34 → 0x2A
0x30 → 0x01
Read 0x30[1] /* Expect 1b if the NCO update is complete */

/* Enable inverse sinc filter */
0x27 → 0xC0

/* Power up DAC outputs */
0x01 → 0x00
```

DEVICE CONFIGURATION AND START-UP SEQUENCE 2

1. Set f_{DCI} = 200 MHz and interpolation to 8×.
2. Enable the PLL, and set f_{REF} = 200 MHz.
3. Enable the inverse sinc filter.
4. Use the delay line-based interface mode with a delay setting of 0.

Derived PLL Settings

The following PLL settings can be derived from the device configuration:

- f_{DAC} = 200 × 8 = 1600 MHz.
- f_{VCO}= f_{DAC} = 1600 MHz (1.03 GHz < f_{VCO} < 2.07 GHz).
- VCO divider = f_{VCO}/f_{DAC} = 1.
- Loop divider = f_{DAC}/f_{REF} = 8.

Start-Up Sequence 2

1. Power up the device (no specific power supply sequence is required).
2. Apply stable DAC clock.
3. Apply stable DCI clock.
4. Feed stable input data.
5. Issue hardware reset (optional).

```
/* Device configuration register write sequence */
0x00 → 0x20 /* Issue software reset */
0x20 → 0x01 /* Device startup configuration */

/* Configure PLL */
0x14 → 0xE7 /* Configure PLL loop BW and charge pump current */
0x15 → 0xC2 /* Configure VCO divider and loop divider */
0x12 → 0xC0 /*Enable the PLL */
0x12 → 0x80
Wait 10ms for autoband selection to finish
Read 0x16[7] /* Expect 1b if the PLL is locked */
```

```
/* Configure data interface */
0x5E → 0x00 /* Configure the delay setting */

0x5F → 0x60
0x0D → 0x16 /* DC couple DCI */
0x0A → 0x00 /* Turn off DLL and duty cycle
correction */

/* Configure interpolation filter */
0x28 → 0x03 /* 8× interpolation */

/* Reset FIFO */
0x25 → 0x01
Read 0x25[1] /* Expect 1b if the FIFO reset is
complete */
Read 0x24 /* The readback should be one of the
three values: 0x37, 0x40, or 0x41 */

/* Enable inverse sinc filter */
0x27 → 0x80

/* Power up DAC outputs */
0x01 → 0x00
```

DEVICE CONFIGURATION REGISTER MAP

Table 25. Device Configuration Register Map

Reg	Name	Bits	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset	RW
0x00	Common	[7:0]	Reserved	SPI_LSB_FIRST	DEVICE_RESET	Reserved					0x00	RW
0x01	PD_CONTROL	[7:0]	PD_IDAC	PD_QDAC	PD_DATARCV	Reserved		PD_DEVICE	PD_DACCLK	PD_FRAME	0xC0	RW
0x03	INTERRUPT_ENABLE0	[7:0]	Reserved	ENABLE_SYNC_LOST	ENABLE_SYNC_LOCKED	ENABLE_SYNC_DONE	ENABLE_PLL_LOST	ENABLE_PLL_LOCKED	ENABLE_OVER_THRESHOLD	ENABLE_DACOUT_MUTED	0x00	RW
0x04	INTERRUPT_ENABLE1	[7:0]	ENABLE_PARITY_FAIL	ENABLE_SED_FAIL	ENABLE_DLL_WARNING	ENABLE_DLL_LOCKED	Reserved	ENABLE_FIFO_UNDERFLOW	ENABLE_FIFO_OVERFLOW	ENABLE_FIFO_WARNING	0x00	RW
0x05	INTERRUPT_FLAG0	[7:0]	Reserved	SYNC_LOST	SYNC_LOCKED	SYNC_DONE	PLL_LOST	PLL_LOCKED	OVER_THRESHOLD	DACOUT_MUTED	0x00	R
0x06	INTERRUPT_FLAG1	[7:0]	PARITY_FAIL	SED_FAIL	DLL_WARNING	DLL_LOCKED	Reserved	FIFO_UNDERFLOW	FIFO_OVERFLOW	FIFO_WARNING	0x00	R
0x07	IRQ_SEL0	[7:0]	Reserved	SEL_SYNC_LOST	SEL_SYNC_LOCKED	SEL_SYNC_DONE	SEL_PLL_LOST	SEL_PLL_LOCKED	SEL_OVER_THRESHOLD	SEL_DACOUT_MUTED	0x00	RW
0x08	IRQ_SEL1	[7:0]	SEL_PARITY_FAIL	SEL_SED_FAIL	SEL_DLL_WARNING	SEL_DLL_LOCKED	Reserved	SEL_FIFO_UNDERFLOW	SEL_FIFO_OVERFLOW	SEL_FIFO_WARNING	0x00	RW
0x09	FRAME_MODE	[7:0]	Reserved		PARUSAGE	FRMUSAGE	Reserved		FRAME_PIN_USAGE		0x00	RW
0x0A	DATA_CNTR_0	[7:0]	DLL_ENABLE	DUTY_CORRECTION_ENABLE	Reserved		DLL_PHASE_OFFSET				0x40	RW
0x0B	DATA_CNTR_1	[7:0]	CLEAR_WARN	Reserved							0x39	RW
0x0C	DATA_CNTR_2	[7:0]	Reserved								0x64	RW
0x0D	DATA_CNTR_3	[7:0]	LOW_DCI_EN	Reserved		DC_COUPLE_LOW_EN	Reserved				0x06	RW
0x0E	DATA_STAT_0	[7:0]	DLL_LOCK	DLL_WARN	DLL_START_WARNING	DLL_END_WARNING	Reserved	DCI_ON	Reserved	DLL_RUNNING	0x00	R
0x10	DACCLK_RECEIVER_CTRL	[7:0]	DACCLK_DUTYCYCLE_CORRECTION	Reserved	DACCLK_CROSSPOINT_CTRL_ENABLE	DACCLK_CROSSPOINT_LEVEL					0xFF	RW
0x11	REFCLK_RECEIVER_CTRL	[7:0]	DUTYCYCLE_CORRECTION	Reserved	REFCLK_CROSSPOINT_CTRL_ENABLE	REFCLK_CROSSPOINT_LEVEL					0x5F	RW
0x12	PLL_CTRL0	[7:0]	PLL_ENABLE	AUTO_MANUAL_SEL	PLL_MANUAL_BAND						0x00	RW
0x14	PLL_CTRL2	[7:0]	PLL_LOOP_BW			PLL_CP_CURRENT					0xE7	RW
0x15	PLL_CTRL3	[7:0]	DIGLOGIC_DIVIDER		Reserved	CROSSPOINT_CTRL_EN	VCO_DIVIDER		LOOP_DIVIDER		0xC9	RW
0x16	PLL_STATUS0	[7:0]	PLL_LOCK	Reserved		VCO_CTRL_VOLTAGE_READBACK					0x00	R
0x17	PLL_STATUS1	[7:0]	Reserved		PLL_BAND_READBACK						0x00	R
0x18	IDAC_FS_ADJ0	[7:0]	IDAC_FULLSCALE_ADJUST_LSB								0xF9	RW
0x19	IDAC_FS_ADJ1	[7:0]	Reserved						IDAC_FULLSCALE_ADJUST_MSB		0xE1	RW
0x1A	QDAC_FS_ADJ0	[7:0]	QDAC_FULLSCALE_ADJUST_LSB								0xF9	RW
0x1B	QDAC_FS_ADJ1	[7:0]	Reserved						QDAC_FULLSCALE_ADJUST_MSB		0x01	RW
0x1C	DIE_TEMP_SENSOR_CTRL	[7:0]	Reserved	FS_CURRENT			REF_CURRENT			DIE_TEMP_SENSOR_EN	0x02	RW
0x1D	DIE_TEMP_LSB	[7:0]	DIE_TEMP_LSB								0x00	R
0x1E	DIE_TEMP_MSB	[7:0]	DIE_TEMP_MSB								0x00	R
0x1F	CHIP_ID	[7:0]	CHIP_ID								0x0A	R
0x20	INTERRUPT_CONFIG	[7:0]	INTERRUPT_CONFIGURATION								0x00	RW
0x21	SYNC_CTRL	[7:0]	Reserved						SYNC_CLK_EDGE_SEL	SYNC_ENABLE	0x00	RW
0x22	FRAME_RST_CTRL	[7:0]	Reserved				ARM_FRAME	EN_CON_FRAME_RESET	FRAME_RESET_MODE		0x12	RW

Reg	Name	Bits	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset	RW
0x23	FIFO_LEVEL_CONFIG	[7:0]	Reserved	INTEGER_FIFO_LEVEL_REQUEST			Reserved	FRACTIONAL_FIFO_LEVEL_REQUEST			0x40	RW
0x24	FIFO_LEVEL_READBACK	[7:0]	Reserved	INTEGER_FIFO_LEVEL_READBACK			Reserved	FRACTIONAL_FIFO_LEVEL_READBACK			0x00	R
0x25	FIFO_CTRL	[7:0]	Reserved						FIFO_SPI_RESET_ACK	FIFO_SPI_RESET_REQUEST	0x00	RW
0x26	DATA_FORMAT	[7:0]	DATA_FORMAT	DATA_PAIRING	DATA_BUS_INVERT	Reserved				DATA_BUS_WIDTH	0x00	RW
0x27	DATAPATH_CTRL	[7:0]	INVSINC_ENABLE	NCO_ENABLE	IQ_GAIN_ADJ_DCOFFSET_ENABLE	IQ_PHASE_ADJ_ENABLE	Reserved	FS4_MODULATION_ENABLE	NCO_SIDEBAND_SEL	SEND_IDATA_TO_QDAC	0x00	RW
0x28	INTERPOLATION_CTRL	[7:0]	Reserved						INTERPOLATION_MODE		0x00	RW
0x29	OVER_THRESHOLD_CTRL0	[7:0]	THRESHOLD_LEVEL_REQUEST_LSB								0x00	RW
0x2A	OVER_THRESHOLD_CTRL1	[7:0]	Reserved			THRESHOLD_LEVEL_REQUEST_MSB					0x00	RW
0x2B	OVER_THRESHOLD_CTRL2	[7:0]	ENABLE_PROTECTION	IQ_DATA_SWAP	Reserved		SAMPLE_WINDOW_LENGTH				0x00	RW
0x2C	INPUT_POWER_READBACK_LSB	[7:0]	INPUT_POWER_READBACK_LSB								0x00	R
0x2D	INPUT_POWER_READBACK_MSB	[7:0]	Reserved			INPUT_POWER_READBACK_MSB					0x00	R
0x30	NCO_CTRL	[7:0]	Reserved	NCO_FRAME_UPDATE_ACK	SPI_NCO_PHASE_RST_ACK	SPI_NCO_PHASE_RST_REQ	Reserved		NCO_SPI_UPDATE_ACK	NCO_SPI_UPDATE_REQ	0x00	RW
0x31	NCO_FREQ_TUNING_WORD0	[7:0]	NCO_FTW0								0x00	RW
0x32	NCO_FREQ_TUNING_WORD1	[7:0]	NCO_FTW1								0x00	RW
0x33	NCO_FREQ_TUNING_WORD2	[7:0]	NCO_FTW2								0x00	RW
0x34	NCO_FREQ_TUNING_WORD3	[7:0]	NCO_FTW3								0x10	RW
0x35	NCO_PHASE_OFFSET0	[7:0]	NCO_PHASE_OFFSET_LSB								0x00	RW
0x36	NCO_PHASE_OFFSET1	[7:0]	NCO_PHASE_OFFSET_MSB								0x00	RW
0x37	IQ_PHASE_ADJ0	[7:0]	IQ_PHASE_ADJ_LSB								0x00	RW
0x38	IQ_PHASE_ADJ1	[7:0]	Reserved			IQ_PHASE_ADJ_MSB					0x00	RW
0x39	LVDS_IN_PWR_DOWN_0	[7:0]	PWR_DOWN_DATA_INPUT_BITS								0x00	RW
0x3B	IDAC_DC_OFFSET0	[7:0]	IDAC_DC_OFFSET_LSB								0x00	RW
0x3C	IDAC_DC_OFFSET1	[7:0]	IDAC_DC_OFFSET_MSB								0x00	RW
0x3D	QDAC_DC_OFFSET0	[7:0]	QDAC_DC_OFFSET_LSB								0x00	RW
0x3E	QDAC_DC_OFFSET1	[7:0]	QDAC_DC_OFFSET_MSB								0x00	RW
0x3F	IDAC_GAIN_ADJ	[7:0]	Reserved		IDAC_GAIN_ADJ						0x20	RW
0x40	QDAC_GAIN_ADJ	[7:0]	Reserved		QDAC_GAIN_ADJ						0x20	RW
0x41	GAIN_STEP_CTRL0		Reserved		RAMP_UP_STEP						0x01	RW

Reg	Name	Bits	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset	RW
0x42	GAIN_STEP_CTRL1		DAC_OUTPUT_OFF	DAC_OUTPUT_STATUS	RAMP_DOWN_STEP						0x41	RW
0x43	TX_ENABLE_CTRL	[7:0]	Reserved					TXENABLE_GAINSTEP_EN	TXENABLE_SLEEP_EN	TXENABLE_POWER_DOWN_EN	0x07	RW
0x44	DAC_OUTPUT_CTRL	[7:0]	DAC_OUTPUT_CTRL_EN	Reserved			FIFO_WARNING_SHUTDOWN_EN	OVERTHRESHOLD_SHUTDOWN_EN	Reserved	FIFO_ERROR_SHUTDOWN_EN	0x8D	RW
0x5E	ENABLE_DLL_DELAY_CELL0	[7:0]	DELAY_CELL_ENABLE [7:0]								0xFF	
0x5F	ENABLE_DLL_DELAY_CELL1	[7:0]	Reserved					DELAY_CELL_ENABLE [10:8]			0x67	RW
0x60	SED_CTRL	[7:0]	SED_ENABLE	SED_ERR_CLEAR	AED_ENABLE	SED_DEPTH	Reserved	AED_PASS	AED_FAIL	SED_FAIL	0x00	RW
0x61	SED_PATT_L_I0	[7:0]	SED_PATTERN_RISE_I0[7:0]								0x00	RW
0x62	SED_PATT_H_I0	[7:0]	SED_PATTERN_RISE_I0[15:8]								0x00	RW
0x63	SED_PATT_L_Q0	[7:0]	SED_PATTERN_FALL_Q0[7:0]								0x00	RW
0x64	SED_PATT_H_Q0	[7:0]	SED_PATTERN_FALL_Q0[15:8]								0x00	RW
0x65	SED_PATT_L_I1	[7:0]	SED_PATTERN_RISE_I1[7:0]								0x00	RW
0x66	SED_PATT_H_I1	[7:0]	SED_PATTERN_RISE_I1[15:8]								0x00	RW
0x67	SED_PATT_L_Q1	[7:0]	SED_PATTERN_FALL_Q1[7:0]								0x00	RW
0x68	SED_PATT_H_Q1	[7:0]	SED_PATTERN_FALL_Q1[15:8]								0x00	RW
0x6A	PARITY_CTRL	[7:0]	PARITY_ENABLE	PARITY_EVEN	PARITY_ERR CLEAR	Reserved			PARERRFAL	PARERRIS	0x00	RW
0x6B	PARITY_ERR_RISING	[7:0]	Parity Rising Edge Error Count								0x00	R
0x6C	PARITY_ERR_FALLING	[7:0]	Parity Falling Edge Error Count								0x00	R
0x7F	Version	[7:0]	Version								0x0B	R

REGISTER DESCRIPTIONS

Defined reserved bits are those whose reset values are not 0x00. Access indicates the read and/or write nature of the register.

SPI CONFIGURE REGISTER

Address: 0x00, Reset: 0x00, Name: Common

Table 26. Bit Descriptions for Common

Bits	Bit Name	Settings	Description	Reset	Access
6	SPI_LSB_FIRST		Serial port communication, MSB-first or LSB-first selection.	0	RW
		0	MSB first.		
		1	LSB first.		
5	DEVICE_RESET		The device resets when 1 is written to this bit. DEVICE_RESET is a self clear bit. After the reset, the bit returns to 0 automatically. The readback is always 0.	0	RW

POWER-DOWN CONTROL REGISTER

Address: 0x01, Reset: 0xC0, Name: PD_CONTROL

Table 27. Bit Descriptions for PD_CONTROL

Bits	Bit Name	Settings	Description	Reset	Access
7	PD_IDAC		The IDAC is powered down when PD_IDAC is set to 1. This bit powers down only the analog portion of the IDAC. The IDAC digital data path is not affected.	1	RW
6	PD_QDAC		The QDAC is powered down when PD_QDAC is set to 1. This bit powers down only the analog portion of the QDAC. The QDAC digital datapath is not affected.	1	RW
5	PD_DATARCV		The data interface circuitry is powered down when PD_DATARCV is set to 1. This bit powers down the data interface and the write side of the FIFO.	0	RW
2	PD_DEVICE		The band gap circuitry is powered down when set to 1. This bit powers down the entire chip.	0	RW
1	PD_DACCLK		The DAC clock powers down when PD_DEVICE is set to 1. This bit powers down the DAC clocking path and, thus, the majority of the digital functions.	0	RW
0	PD_FRAME		The frame receiver powers down when PD_FRAME is set to 1. The frame signal is internally pulled low. Set to 1 when the frame is not used.	0	RW

INTERRUPT ENABLE0 REGISTER

Address: 0x03, Reset: 0x00, Name: INTERRUPT_ENABLE0

Table 28. Bit Descriptions for INTERRUPT_ENABLE0

Bits	Bit Name	Settings	Description	Reset	Access
6	ENABLE_SYNC_LOST		Enable interrupt for sync lost.	0	RW
5	ENABLE_SYNC_LOCKED		Enable interrupt for sync lock.	0	RW
4	ENABLE_SYNC_DONE		Enable interrupt for sync done.	0	RW
3	ENABLE_PLL_LOST		Enable interrupt for PLL lost.	0	RW
2	ENABLE_PLL_LOCKED		Enable interrupt for PLL locked.	0	RW
1	ENABLE_OVER_THRESHOLD		Enable interrupt for overthreshold.	0	RW
0	ENABLE_DACOUT_MUTED		Enable interrupt for DACOUT muted.	0	RW

INTERRUPT ENABLE1 REGISTER

Address: 0x04, Reset: 0x00, Name: INTERRUPT_ENABLE1

Table 29. Bit Descriptions for INTERRUPT_ENABLE1

Bits	Bit Name	Settings	Description	Reset	Access
7	ENABLE_PARITY_FAIL		Enable interrupt for parity failure.	0	RW
6	ENABLE_SED_FAIL		Enable interrupt for SED failure.	0	RW
5	ENABLE_DLL_WARNING		Enable interrupt for DLL warning.	0	RW
4	ENABLE_DLL_LOCKED		Enable interrupt for DLL locked.	0	RW
2	ENABLE_FIFO_UNDERFLOW		Enable interrupt for FIFO underflow.	0	RW
1	ENABLE_FIFO_OVERFLOW		Enable interrupt for FIFO overflow.	0	RW
0	ENABLE_FIFO_WARNING		Enable interrupt for FIFO warning.	0	RW

INTERRUPT FLAG0 REGISTER

Address: 0x05, Reset: 0x00, Name: INTERRUPT_FLAG0

Table 30. Bit Descriptions for INTERRUPT_FLAG0

Bits	Bit Name	Settings	Description	Reset	Access
6	SYNC_LOST		SYNC_LOST is set to 1 when sync is lost.	0	R
5	SYNC_LOCKED		SYNC_LOCKED is set to 1 when sync is locked.	0	R
4	SYNC_DONE		SYNC_DONE is set to 1 when sync is done.	0	R
3	PLL_LOST		PLL_LOST is set to 1 when PLL loses lock.	0	R
2	PLL_LOCKED		PLL_LOCKED is set to 1 when PLL is locked.	0	R
1	OVER_THRESHOLD		OVER_THRESHOLD is set to 1 when input power is overthreshold.	0	R
0	DACOUT_MUTED		DACOUT_MUTED is set to 1 when the DAC output is muted (midscale dc).	0	R

INTERRUPT FLAG1 REGISTER

Address: 0x06, Reset: 0x00, Name: INTERRUPT_FLAG1

Table 31. Bit Descriptions for INTERRUPT_FLAG1

Bits	Bit Name	Settings	Description	Reset	Access
7	PARITY_FAIL		PARITY_FAIL is set to 1 when the parity check fails.	0	R
6	SED_FAIL		SED_FAIL is set to 1 when the SED comparison fails.	0	R
5	DLL_WARNING		DLL_WARNING is set to 1 when the DLL raises a warning.	0	R
4	DLL_LOCKED		DLL_LOCKED is set to 1 when the DLL is locked.	0	R
2	FIFO_UNDERFLOW		FIFO_UNDERFLOW is set to 1 when the FIFO read pointer catches the FIFO write pointer.	0	R
1	FIFO_OVERFLOW		FIFO_OVERFLOW is set to 1 when the when the FIFO read pointer catches the FIFO read pointer.	0	R
0	FIFO_WARNING		FIFO_WARNING is set to 1 when the FIFO is one slot from empty (≤1) or full (≥6).	0	R

INTERRUPT SELECT0 REGISTER

Address: 0x07, Reset: 0x00, Name: IRQ_SEL0

Table 32. Bit Descriptions for IRQ_SEL0

Bits	Bit Name	Settings	Description	Reset	Access
6	SEL_SYNC_LOST	0	Selects the $\overline{\text{IRQ1}}$ pin.	0	RW
		1	Selects the $\overline{\text{IRQ2}}$ pin.		
5	SEL_SYNC_LOCKED	0	Selects the $\overline{\text{IRQ1}}$ pin.	0	RW
		1	Selects the $\overline{\text{IRQ2}}$ pin.		
4	SEL_SYNC_DONE	0	Selects the $\overline{\text{IRQ1}}$ pin.	0	RW
		1	Selects the $\overline{\text{IRQ2}}$ pin.		
3	SEL_PLL_LOST	0	Selects the $\overline{\text{IRQ1}}$ pin.	0	RW
		1	Selects the $\overline{\text{IRQ2}}$ pin.		
2	SEL_PLL_LOCKED	0	Selects the $\overline{\text{IRQ1}}$ pin.	0	RW
		1	Selects the $\overline{\text{IRQ2}}$ pin.		
1	SEL_OVER_THRESHOLD	0	Selects the $\overline{\text{IRQ1}}$ pin.	0	RW
		1	Selects the $\overline{\text{IRQ2}}$ pin.		
0	SEL_DACOUT_MUTED	0	Selects the $\overline{\text{IRQ1}}$ pin.	0	RW

INTERRUPT SELECT1 REGISTER

Address: 0x08, Reset: 0x00, Name: IRQ_SEL1

Table 33. Bit Descriptions for IRQ_SEL1

Bits	Bit Name	Settings	Description	Reset	Access
7	SEL_PARITY_FAIL	1	Selects the $\overline{\text{IRQ2}}$ pin.	0	RW
		0	Selects the $\overline{\text{IRQ1}}$ pin.		
6	SEL_SED_FAIL	1	Selects the $\overline{\text{IRQ2}}$ pin.	0	RW
		0	Selects the $\overline{\text{IRQ1}}$ pin.		
5	SEL_DLL_WARNING	0	Selects the $\overline{\text{IRQ1}}$ pin.	0	RW
4	SEL_DLL_LOCKED	1	Selects the $\overline{\text{IRQ2}}$ pin.	0	RW
		0	Selects the $\overline{\text{IRQ1}}$ pin.		
2	SEL_FIFO_UNDERFLOW	1	Selects the $\overline{\text{IRQ2}}$ pin.	0	RW
		0	Selects the $\overline{\text{IRQ1}}$ pin.		
1	SEL_FIFO_OVERFLOW	1	Selects the $\overline{\text{IRQ2}}$ pin.	0	RW
		0	Selects the $\overline{\text{IRQ1}}$ pin.		
0	SEL_FIFO_WARNING	1	Selects the $\overline{\text{IRQ2}}$ pin.	0	RW
		0	Selects the $\overline{\text{IRQ1}}$ pin.		

FRAME MODE REGISTER

Address: 0x09, Reset: 0x00, Name: FRAME_MODE

Table 34. Bit Descriptions for FRAME_MODE

Bits	Bit Name	Description	Reset	Access
5	PARUSAGE	Must set to 1 when parity is used	0	RW
4	FRMUSAGE	Must set to 1 when frame is used.	0	RW
[1:0]	FRAME_PIN_USAGE	0 = no effect. 1 = parity. 2 = frame. 3 = reserved.	0x0	RW

DATA CONTROL 0 REGISTER

Address: 0x0A, Reset: 0x40, Name: DATA_CNTR_0

Table 35. Bit Descriptions for DATA_CNTR_0

Bits	Bit Name	Description	Reset	Access
7	DLL_ENABLE	1 = enable DLL. 0 = disable DLL.	0	RW
6	DUTY_CORRECTION_ENABLE	1 = enable duty cycle correction. 0 = disable duty cycle correction.	1	RW
[3:0]	DLL_PHASE_OFFSET	Locked phase = 90° + n ×11.25°, where n is the 4 bit signed magnitude number. Valid phase setting ranges from −6 to +6, 13 phases in total.	0x0	RW

DATA CONTROL 1 REGISTER

Address: 0x0B, Reset: 0x39, Name: DATA_CNTR_1

Table 36. Bit Descriptions for DATA_CNTR_1

Bits	Bit Name	Description	Reset	Access
7	CLEAR_WARN	1= clears data receiver warning bits (Register 0x0E[6:4]).	0	RW
[6:0]	Reserved	Must write the default value for optimal performance.	0x39	RW

DATA CONTROL 2 REGISTER

Address: 0x0C, Reset: 0x64, Name: DATA_CNTR_2

Table 37. Bit Descriptions for DATA_CNTR_2

Bits	Bit Name	Description	Reset	Access
[7:0]	Reserved	Must write the default value for optimal performance.	0x64	RW

DATA CONTROL 3 REGISTER

Address: 0x0D, Reset: 0x06, Name: DATA_CNTR_3

Table 38. Bit Descriptions for DATA_CNTR_3

Bits	Bit Name	Description	Reset	Access
7	LOW_DCI_EN	Set to 0 when DLL is enabled and DCI rate is ≥350 MHz. Set to 1 when DLL is enabled and DCI rate is <350 MHz.	0	RW
4	DC_COUPLE_LOW_EN	Set to 0 when DLL is enabled and delay line is disabled. Set to 1 when DLL is disabled and delay line is enabled. It is recommended that DLL mode be used for a DCI rate faster than 250 MHz and the delay line mode be used for DCI rate slower than 250 MHz.	0	RW
[3:0]	Reserved	Must write the default value for optimal performance.	0x6	RW

DATA STATUS 0 REGISTER

Address: 0x0E, Reset: 0x00, Name: DATA_STAT_0

Table 39. Bit Descriptions for DATA_STAT_0

Bits	Bit Name	Description	Reset	Access
7	DLL_LOCK	1 = DLL lock.	0	R
6	DLL_WARN	1 = DLL near beginning/end of delay line.	0	R
5	DLL_START_WARNING	1 = DLL at beginning of delay line.	0	R
4	DLL_END_WARNING	1 = DLL at end of delay line.	0	R
3	Reserved	Reserved.	0	R
2	DCI_ON	1 = user has provided a clock >100 MHz.	0	R
1	Reserved	Reserved.	0	R
0	DLL_RUNNING	1 = closed loop DLL attempting to lock. 0 = delay fixed at middle of delay line.	0	R

DAC CLOCK RECEIVER CONTROL REGISTER

Address: 0x10, Reset: 0xFF, Name: DACCLK_RECEIVER_CTRL

Table 40. Bit Descriptions for DACCLK_RECEIVER_CTRL

Bits	Bit Name	Settings	Description	Reset	Access
7	DACCLK_DUTYCYCLE_CORRECTION		Enables duty cycle correction at the DACCLK input. For best performance, the default and recommended status is turned on.	1	RW
6	Reserved		Must write the default value for optimal performance	1	RW
5	DACCLK_CROSSPOINT_CTRL_ENABLE		Enables crosspoint control at the DACCLK input. For best performance, the default and recommended status is turned on.	1	RW
[4:0]	DACCLK_CROSSPOINT_LEVEL		A twos complement value. For best performance, it is recommended to set DACCLK_CROSSPOINT_LEVEL to the default value.	0x1F	RW
		01111	Highest crosspoint.		
		11111	Lowest crosspoint.		

REF CLOCK RECEIVER CONTROL REGISTER

Address: 0x11, Reset: 0x5F, Name: REFCLK_RECEIVER_CTRL

Table 41. Bit Descriptions for REFCLK_RECEIVER_CTRL

Bits	Bit Name	Settings	Description	Reset	Access
7	DUTYCYCLE_CORRECTION		Enables duty cycle correction at the REFx/SYNCx input. For best performance, the default and recommended status is turned off.	0	RW
6	Reserved		Must write the default value for optimal performance	1	RW
5	REFCLK_CROSSPOINT_CTRL_ENABLE		Enables crosspoint control at the REFx/SYNCx input. For best performance, the default and recommended status is turned off.	0	RW
[4:0]	REFCLK_CROSSPOINT_LEVEL		A twos complement value. For best performance, it is recommended to set REFCLK_CROSSPOINT_LEVEL to the default value.	0x1F	RW
		01111	Highest crosspoint.		
		11111	Lowest crosspoint.		

PLL CONTROL 0 REGISTER

Address: 0x12, Reset: 0x00, Name: PLL_CTRL0

Table 42. Bit Descriptions for PLL_CTRL0

Bits	Bit Name	Settings	Description	Reset	Access
7	PLL_ENABLE		Enables PLL clock multiplier.	0	RW
6	AUTO_MANUAL_SEL		PLL band selection mode.	0	RW
		0	Automatic mode.		
		1	Manual mode.		
[5:0]	PLL_MANUAL_BAND		PLL band setting in manual mode. 64 bands in total, covering a 1 GHz to 2.1 GHz VCO range.	0x00	RW
		000000	Lowest band (1 GHz).		
		111111	Highest band (2.1 GHz).		

PLL CONTROL 2 REGISTER

Address: 0x14, Reset: 0xE7, Name: PLL_CTRL2

Table 43. Bit Descriptions for PLL_CTRL2

Bits	Bit Name	Settings	Description	Reset	Access
[7:5]	PLL_LOOP_BW		Selects the PLL filter bandwidth. The default and recommended setting is 111 for optimal PLL performance.	0x7	RW
		0x00	Lowest setting.		
		0x1F	Highest setting.		
[4:0]	PLL_CP_CURRENT		Sets nominal PLL charge pump current. The default and recommended setting is 00111 for optimal PLL performance.	0x07	RW
		0x00	Lowest setting.		
		0x1F	Highest setting.		

PLL CONTROL 3 REGISTER

Address: 0x15, Reset: 0xC9, Name: PLL_CTRL3

Table 44. Bit Descriptions for PLL_CTRL3

Bits	Bit Name	Settings	Description	Reset	Access
[7:6]	DIGLOGIC_DIVIDER		REFCLK to PLL digital clock divide ratio. The PLL digital clock drives the internal PLL logics. The divide ratio must be set to ensure that the PLL digital clock is less than 75 MHz.	0x3	RW
		00	f_{REFCLK}/f_{DIG} = 2.		
		01	f_{REFCLK}/f_{DIG} = 4.		
		10	f_{REFCLK}/f_{DIG} = 8.		
		11	f_{REFCLK}/f_{DIG} = 16.		
4	CROSSPOINT_CTRL_EN		Enable loop divider crosspoint control. The default and recommended setting is set to 0 for optimal PLL performance.	0	RW
[3:2]	VCO_DIVIDER		PLL VCO divider. This divider determines the ratio of the VCO frequency to the DACCLK frequency.	0x2	RW
		00	f_{VCO}/f_{DACCLK} = 1.		
		01	f_{VCO}/f_{DACCLK} = 2.		
		10	f_{VCO}/f_{DACCLK} = 4.		
		11	f_{VCO}/f_{DACCLK} = 4.		
[1:0]	LOOP_DIVIDER		PLL divider. This divider determines the ratio of the DACCLK frequency to the REFCLK frequency.	0x1	RW
		00	f_{DACCLK}/f_{REFCLK} = 2.		
		01	f_{DACCLK}/f_{REFCLK} = 4.		
		10	f_{DACCLK}/f_{REFCLK} = 8.		
		11	f_{DACCLK}/f_{REFCLK} = 16.		

PLL STATUS 0 REGISTER

Address: 0x16, Reset: 0x00, Name: PLL_STATUS0

Table 45. Bit Descriptions for PLL_STATUS0

Bits	Bit Name	Settings	Description	Reset	Access
7	PLL_LOCK		PLL clock multiplier output is stable.	0	R
[3:0]	VCO_CTRL_VOLTAGE_READBACK		VCO control voltage readback. A binary value.	0x0	R
		1111	The highest VCO control voltage.		
		0111	The midvalue when a proper VCO band is selected. When the PLL is locked, selecting a higher VCO band decreases this value and selecting a lower VCO band increases this value.		
		0000	The lowest VCO control voltage.		

PLL STATUS 1 REGISTER

Address: 0x17, Reset: 0x00, Name: PLL_STATUS1

Table 46. Bit Descriptions for PLL_STATUS1

Bits	Bit Name	Settings	Description	Reset	Access
[5:0]	PLL_BAND_READBACK		Indicates the VCO band currently selected.	0x00	R

IDAC FS ADJUST LSB REGISTER

Address: 0x18, Reset: 0xF9, Name: IDAC_FS_ADJ0

Table 47. Bit Descriptions for IDAC_FS_ADJ0

Bits	Bit Name	Settings	Description	Reset	Access
[7:0]	IDAC_FULLSCALE_ADJUST_LSB		IDAC full-scale adjust, these bits, along with Bits[1:0] in Register 0x19, set the full-scale current of the IDAC. The full-scale current can be adjusted from 8.64 mA to 31.68 mA. The default value (0x1F9) sets the full-scale current to 20 mA.	0xF9	RW

IDAC FS ADJUST MSB REGISTER

Address: 0x19, Reset: 0xE1, Name: IDAC_FS_ADJ1

Table 48. Bit Descriptions for IDAC_FS_ADJ1

Bits	Bit Name	Settings	Description	Reset	Access
[7:5]	Reserved		Set to default value for optimal performance.	0x7	RW
[1:0]	IDAC_FULLSCALE_ADJUST_MSB		IDAC full-scale adjust, these bits, along with Bits[7:0] in Register 0x18,the full-scale current of the IDAC. The full-scale current can be adjusted from 8.64 mA to 31.68 mA. The default value (0x1F9) sets the full-scale current to 20 mA.	0x1	RW

QDAC FS ADJUST LSB REGISTER

Address: 0x1A, Reset: 0xF9, Name: QDAC_FS_ADJ0

Table 49. Bit Descriptions for QDAC_FS_ADJ0

Bits	Bit Name	Settings	Description	Reset	Access
[7:0]	QDAC_FULLSCALE_ADJUST_LSB		QDAC Full-Scale Adjust, these bits, along with Bits[1:0] in Register 0x1B, set the full-scale current of the QDAC. The full-scale current can be adjusted from 8.64 mA to 31.68 mA. The default value (0x1F9) sets the full-scale current to 20 mA.	0xF9	RW

QDAC FS ADJUST MSB REGISTER

Address: 0x1B, Reset: 0x01, Name: QDAC_FS_ADJ1

Table 50. Bit Descriptions for QDAC_FS_ADJ1

Bits	Bit Name	Settings	Description	Reset	Access
[1:0]	QDAC_FULLSCALE_ADJUST_MSB		QDAC Full-Scale Adjust, these bits, along with Bits[7:0] in Register 0x1A, set the full-scale current of the QDAC. The full-scale current can be adjusted from 8.64 mA to 31.68 mA. The default value (0x1F9) sets the full-scale current to 20 mA.	0x1	RW

DIE TEMPERATURE SENSOR CONTROL REGISTER

Address: 0x1C, Reset: 0x02, Name: DIE_TEMP_SENSOR_CTRL

Table 51. Bit Descriptions for DIE_TEMP_SENSOR_CTRL

Bits	Bit Name	Settings	Description	Reset	Access
[6:4]	FS_CURRENT		Temperature sensor ADC full-scale current. Using the default setting is recommended.	0x0	RW
		000	50 µA.		
		001	62.5 µA.		
		…			
		110	125 µA.		
		111	137.5 µA.		
[3:1]	REF_CURRENT		Temperature sensor ADC reference current. Using the default setting is recommended.	0x1	RW
		000	12.5 µA.		
		001	19 µA.		
		…			
		110	50 µA.		
		111	56.5 µA.		
0	DIE_TEMP_SENSOR_EN		Enable the on-chip temperature sensor.	0x0	RW

DIE TEMPERATURE LSB REGISTER

Address: 0x1D, Reset: 0x00, Name: DIE_TEMP_LSB

Table 52. Bit Descriptions for DIE_TEMP_LSB

Bits	Bit Name	Settings	Description	Reset	Access
[7:0]	DIE_TEMP_LSB		Die temperature, these bits, along with Bits[7:0] in Register 0x1E, indicate the approximate die temperature. For more information, see the Temperature Sensor section.	0x00	R

DIE TEMPERATURE MSB REGISTER

Address: 0x1E, Reset: 0x00, Name: DIE_TEMP_MSB

Table 53. Bit Descriptions for DIE_TEMP_MSB

Bits	Bit Name	Settings	Description	Reset	Access
[7:0]	DIE_TEMP_MSB		Die temperature, these bits, along with Bits[7:0] in Register 0x1D, indicate the approximate die temperature. For more information, see the Temperature Sensor section.	0x00	R

CHIP ID REGISTER

Address: 0x1F, Reset: 0x0A, Name: CHIP_ID

Table 54. Bit Descriptions for CHIP_ID

Bits	Bit Name	Settings	Description	Reset	Access
[7:0]	CHIP_ID		The AD9142A chip ID is 0x0A.	0x0A	R

INTERRUPT CONFIGUATION REGISTER

Address: 0x20, Reset: 0x00, Name: INTERRUPT_CONFIG

Table 55. Bit Descriptions for INTERRUPT_CONFIG

Bits	Bit Name	Settings	Description	Reset	Access
[7:0]	INTERRUPT_CONFIGURATION	0x00	Test mode.	0x00	RW
		0x01	Recommended mode (described in the Interrupt Request Operation section).		

SYNC CONTROL REGISTER

Address: 0x21, Reset: 0x00, Name: SYNC_CTRL

Table 56. Bit Descriptions for SYNC_CTRL

Bits	Bit Name	Settings	Description	Reset	Access
1	SYNC_CLK_EDGE_SEL		Selects the sampling edge of the DACCLK on the sync clock.	0	RW
		0	SYNC CLK is sampled by the falling edges of DACCLK.		
		1	SYNC CLK is sampled by the rising edges of DACCLK.		
0	SYNC_ENABLE		Enables multichip synchronization.	0	RW

FRAME RESET CONTROL REGISTER

Address: 0x22, Reset: 0x12, Name: FRAME_RST_CTRL

Table 57. Bit Descriptions for FRAME_RST_CTRL

Bits	Bit Name	Settings	Description	Reset	Access
3	ARM_FRAME		This bit is used to retrigger a frame reset in one shot mode (when Bit 2 is set to 0). Setting this bit to 1 requests the device to respond to the next valid frame pulse.	0	RW
2	EN_CON_FRAME_RESET		Frame reset mode selection.	0	RW
		0	Responds to the first valid frame pulse and resets the FIFO one time only. This is the default and recommended mode.		
		1	Responds to every valid frame pulse and resets the FIFO continuously.		
[1:0]	FRAME_RESET_MODE		These bits determine what is to be reset when the device receives a valid frame signal.	0x2	RW
		00	FIFO only.		
		01	NCO only.		
		10	FIFO and NCO.		
		11	None.		

FIFO LEVEL CONFIGURATION REGISTER

Address: 0x23, Reset: 0x40, Name: FIFO_LEVEL_CONFIG

Table 58. Bit Descriptions for FIFO_LEVEL_CONFIG

Bits	Bit Name	Settings	Description	Reset	Access
[6:4]	INTEGER_FIFO_LEVEL_REQUEST		These bits set the integer FIFO level. This is the difference between the read pointer and the write pointer values in the unit of input data rate (f_{DATA}). The default and recommended FIFO level is integer level = 4 and fractional level = 0. See the FIFO Operation section for details.	0x4	RW
		000	0.		
		001	1.		
		…			
		111	7.		
[2:0]	FRACTIONAL_FIFO_LEVEL_REQUEST		Set the fractional FIFO level. This is the difference between the read pointer and the write pointer values in the unit of DACCLK rate (f_{DAC}). The maximum allowed setting value = interpolation rate − 1. See the FIFO Operation section for details.	0x0	RW
		000	0.		
		001	1.		

FIFO LEVEL READBACK REGISTER

Address: 0x24, Reset: 0x00, Name: FIFO_LEVEL_READBACK

Table 59. Bit Descriptions for FIFO_LEVEL_READBACK

Bits	Bit Name	Settings	Description	Reset	Access
[6:4]	INTEGER_FIFO_LEVEL_READBACK		The integer FIFO level read back. The difference between the overall FIFO level request and readback should be within two DACCLK cycles. See the FIFO Operation section for details.	0x0	R
[2:0]	FRACTIONAL_FIFO_LEVEL_READBACK		The fractional FIFO level read back. This value should be used in combination with the readback in Bits[6:4].	0x0	R

FIFO CONTROL REGISTER

Address: 0x25, Reset: 0x00, Name: FIFO_CTRL

Table 60. Bit Descriptions for FIFO_CTRL

Bits	Bit Name	Settings	Description	Reset	Access
1	FIFO_SPI_RESET_ACK		Acknowledge a serial port initialized FIFO reset.	0x0	R
0	FIFO_SPI_RESET_REQUEST		Initialize a FIFO reset via the serial port.	0x0	RW

DATA FORMAT SELECT REGISTER

Address: 0x26, Reset: 0x00, Name: DATA_FORMAT_SEL

Table 61. Bit Descriptions for DATA_FORMAT_SEL

Bits	Bit Name	Settings	Description	Reset	Access
7	DATA_FORMAT		Select binary or twos complement data format.	0x0	RW
		0	Input data in twos complement format.		
		1	Input data in binary format.		
6	DATA_PAIRING		Indicate I/Q data pairing on data input.	0x0	RW
		0	I samples are paired with the next Q samples.		
		1	I samples are paired with the prior Q samples.		
5	DATA_BUS_INVERT		Swap the bit order of the data input port. MSBs become the LSBs: D[15:0] changes to D[0:15].	0x0	RW
		0	The order of the data bits corresponds to the pin descriptions in Table 9.		
		1	The order of the data bits is inverted.		
0	DATA_BUS_WIDTH		Data interface mode. See the LVDS Input Data Ports section for information about the operation of the different interface modes.	0x0	RW
		0	Word interface mode; 16-bit interface bus width.		
		1	Byte interface mode; 8-bit interface bus width.		

DATAPATH CONTROL REGISTER

Address: 0x27, Reset: 0x00, Name: DATAPATH_CTRL

Table 62. Bit Descriptions for DATAPATH_CTRL

Bits	Bit Name	Settings	Description	Reset	Access
7	INVSINC_ENABLE		Enable the inverse sinc filter.	0x0	RW
6	NCO_ENABLE		Enable the NCO.	0x0	RW
5	IQ_GAIN_ADJ_DCOFFSET_ENABLE		Enable digital IQ gain adjustment and dc offset.	0x0	RW
4	IQ_PHASE_ADJ_ENABLE		Enable digital IQ phase adjustment.	0x0	RW
2	FS4_MODULATION_ENABLE		Enable $f_S/4$ modulation function.	0x0	RW
1	NCO_SIDEBAND_SEL		Selects the single-side NCO modulation image.	0x0	RW
		0	The NCO outputs the high-side image.		
		1	The NCO outputs the low-side image.		
0	SEND_IDATA_TO_QDAC		Send the IDATA to the QDAC. When enabled, I data is sent to both the IDAC and the QDAC. The Q data path still runs, and the Q data is ignored.	0x0	RW

INTERPOLATION CONTROL REGISTER

Address: 0x28, Reset: 0x00, Name: INTERPOLATION_CTRL

Table 63. Bit Descriptions for INTERPOLATION_CTRL

Bits	Bit Name	Settings	Description	Reset	Access
[1:0]	INTERPOLATION_MODE		Interpolation rate and mode selection.	0x0	RW
		00	2× Mode; use HB1 filter.		
		10	4× mode; use HB1 and HB2 filters.		
		11	8× mode; use all three filters (HB1, HB2, and HB3).		

OVER THRESHOLD CONTROL 0 REGISTER

Address: 0x29, Reset: 0x00, Name: OVER_THRESHOLD_CTRL0

Table 64. Bit Descriptions for OVER_THRESHOLD_CTRL0

Bits	Bit Name	Settings	Description	Reset	Access
[7:0]	THRESHOLD_LEVEL_REQUEST_LSB		These bits, along with Bits[4:0] in Register 0x2A, set the minimum average input power ($I^2 + Q^2$) to trigger the input power protection function.	0x0	RW

OVER THRESHOLD CONTROL 1 REGISTER

Address: 0x2A, Reset: 0x00, Name: OVER_THRESHOLD_CTRL1

Table 65. Bit Descriptions for OVER_THRESHOLD_CTRL1

Bits	Bit Name	Settings	Description	Reset	Access
[4:0]	THRESHOLD_LEVEL_REQUEST_MSB		These bits, along with Bits[7:0] in Register 0x29, set the minimum average input power ($I^2 + Q^2$) to trigger the input power protection function.	0x00	RW

OVER THRESHOLD CONTROL 2 REGISTER

Address: 0x2B, Reset: 0x00, Name: OVER_THRESHOLD_CTRL2

Table 66. Bit Descriptions for OVER_THRESHOLD_CTRL2

Bits	Bit Name	Settings	Description	Reset	Access
7	ENABLE_PROTECTION		Enable input power protection.	0x0	RW
6	IQ_DATA_SWAP		Swap I and Q data in average power calculation.	0x0	RW
[3:0]	SAMPLE_WINDOW_LENGTH		Number of data input samples for power averaging.	0x0	RW
		0000	512 IQ data sample pairs.		
		0001	1024 IQ data sample pairs.		
		…			
		1010	2^{19} IQ data sample pairs.		
		1011 to 1111	invalid.		

INPUT POWER READBACK LSB REGISTER

Address: 0x2C, Reset: 0x00, Name: INPUT_POWER_READBACK_LSB

Table 67. Bit Descriptions for INPUT_POWER_READBACK_LSB

Bits	Bit Name	Settings	Description	Reset	Access
[7:0]	INPUT_POWER_READBACK_LSB		These bits, along with Bits[4:0] in Register 0x2D, set the input signal average power readback.	0x0	R

INPUT POWER READBACK MSB REGISTER

Address: 0x2D, Reset: 0x00, Name: INPUT_POWER_READBACK_MSB

Table 68. Bit Descriptions for INPUT_POWER_READBACK_MSB

Bits	Bit Name	Settings	Description	Reset	Access
[4:0]	INPUT_POWER_READBACK_MSB		These bits, along with Bits[7:0] in Register 0x2C, set the input signal average power readback.	0x00	R

NCO CONTROL REGISTER

Address: 0x30, Reset: 0x00, Name: NCO_CTRL

Table 69. Bit Descriptions for NCO_CTRL

Bits	Bit Name	Settings	Description	Reset	Access
6	NCO_FRAME_UPDATE_ACK		Frequency tuning word update request from frame.	0x0	R
5	SPI_NCO_PHASE_RST_ACK		NCO phase SPI reset acknowledge.	0x0	R
4	SPI_NCO_PHASE_RST_REQ		NCO phase SPI reset request.	0x0	RW
1	NCO_SPI_UPDATE_ACK		Frequency tuning word update acknowledge.	0x0	R
0	NCO_SPI_UPDATE_REQ		Frequency tuning word update request from SPI.	0x0	RW

NCO FREQUENCY TUNING WORD 0 REGISTER

Address: 0x31, Reset: 0x00, Name: NCO_FREQ_TUNING_WORD0

Table 70. Bit Descriptions for NCO_FREQ_TUNING_WORD0

Bits	Bit Name	Settings	Description	Reset	Access
[7:0]	NCO_FTW0		Bits[7:0] together with the bits in Register 0x32, Register 0x33, and Register 0x34 form the 32-bit frequency tuning word that determines the frequency of the complex carrier generated by the on-chip NCO. The frequency is not updated when the FTW registers are written. The values are only updated when a serial port update or frame update is initialized in Register 0x30. It is in twos complement format.	0x00	RW

NCO FREQUENCY TUNING WORD 1 REGISTER

Address: 0x32, Reset: 0x00, Name: NCO_FREQ_TUNING_WORD1

Table 71. Bit Descriptions for NCO_FREQ_TUNING_WORD1

Bits	Bit Name	Settings	Description	Reset	Access
[7:0]	NCO_FTW1		Bits[7:0] together with the bits in Register 0x31, Register 0x33, and Register 0x34 form the 32-bit frequency tuning word that determines the frequency of the complex carrier generated by the on-chip NCO. The frequency is not updated when the FTW registers are written. The values are only updated when a serial port update or frame update is initialized in Register 0x30. It is in twos complement format.	0x00	RW

NCO FREQUENCY TUNING WORD 2 REGISTER

Address: 0x33, Reset: 0x00, Name: NCO_FREQ_TUNING_WORD2

Table 72. Bit Descriptions for NCO_FREQ_TUNING_WORD2

Bits	Bit Name	Settings	Description	Reset	Access
[7:0]	NCO_FTW2		Bits[7:0] together with the bits in Register 0x31, Register 0x32, and Register 0x34 form the 32-bit frequency tuning word that determines the frequency of the complex carrier generated by the on-chip NCO. The frequency is not updated when the FTW registers are written. The values are only updated when a serial port update or frame update is initialized in Register 0x30. It is in twos complement format.	0x00	RW

NCO FREQUENCY TUNING WORD 3 REGISTER

Address: 0x34, Reset: 0x10, Name: NCO_FREQ_TUNING_WORD3

Table 73. Bit Descriptions for NCO_FREQ_TUNING_WORD3

Bits	Bit Name	Settings	Description	Reset	Access
[7:0]	NCO_FTW3		Bits[7:0] together with the bits in Register 0x31 through Register 0x33 form the 32-bit frequency tuning word that determines the frequency of the complex carrier generated by the on-chip NCO. The frequency is not updated when the FTW registers are written. The values are only updated when a serial port update or frame update is initialized in Register 0x30. It is in twos complement format.	0x10	RW

NCO PHASE OFFSET 0 REGISTER

Address: 0x35, Reset: 0x00, Name: NCO_PHASE_OFFSET0

Table 74. Bit Descriptions for NCO_PHASE_OFFSET0

Bits	Bit Name	Settings	Description	Reset	Access
[7:0]	NCO_PHASE_OFFSET_LSB		This register, together with Register 0x36, sets the initial phase of the complex carrier signal upon reset. The phase offset spans from 0° to 360°. Each bit represents an offset of 0.0055°. This value is in twos complement format.	0x00	RW

NCO PHASE OFFSET 1 REGISTER

Address: 0x36, Reset: 0x00, Name: NCO_PHASE_OFFSET1

Table 75. Bit Descriptions for NCO_PHASE_OFFSET1

Bits	Bit Name	Settings	Description	Reset	Access
[7:0]	NCO_PHASE_OFFSET_MSB		This register, together with Register 0x35, sets the initial phase of the complex carrier signal upon reset. The phase offset spans from 0° to 360°. Each bit represents an offset of 0.0055°. This value is in twos complement format.	0x00	RW

IQ PHASE ADJUST 0 REGISTER

Address: 0x37, Reset: 0x00, Name: IQ_PHASE_ADJ0

Table 76. Bit Descriptions for IQ_PHASE_ADJ0

Bits	Bit Name	Settings	Description	Reset	Access
[7:0]	IQ_PHASE_ADJ_LSB		Q phase adjust, Bits[7:0] along with Bits[4:0] in Register 0x38, is used to insert a phase offset between the I and Q datapaths. It provides an adjustment range of ±14° with a step of 0.0035°. This value is in twos complement. See the Quadrature Phase Adjustment section for more information.	0x00	RW

IQ PHASE ADJUST 1 REGISTER

Address: 0x38, Reset: 0x00, Name: IQ_PHASE_ADJ1

Table 77. Bit Descriptions for IQ_PHASE_ADJ1

Bits	Bit Name	Settings	Description	Reset	Access
[4:0]	IQ_PHASE_ADJ_MSB		IQ phase adjust, Bits[4:0] along with Bits[7:0] in Register 0x37, is used to insert a phase offset between the I and Q datapaths. It provides an adjustment range of ±14° with a step of 0.0035°. This value is in twos complement. See the Quadrature Phase Adjustment section for more information.	0x0	RW

POWER DOWN DATA INPUT 0 REGISTER

Address: 0x39, Reset: 0x00, Name: LVDS_IN_PWR_DOWN_0

Table 78. Bit Descriptions for LVDS_IN_PWR_DOWN_0

Bits	Bit Name	Settings	Description	Reset	Access
[3:0]	PWR_DOWN_DATA_INPUT_BITS		Powers down data input D[3:0]. Each bit controls one data input bit. These bits can be powered down individually.	0x0	RW

IDAC DC OFFSET 0 REGISTER

Address: 0x3B, Reset: 0x00, Name: IDAC_DC_OFFSET0

Table 79. Bit Descriptions for IDAC_DC_OFFSET0

Bits	Bit Name	Settings	Description	Reset	Access
[7:0]	IDAC_DC_OFFSET_LSB		DAC dc offset, Bits[7:0] along with Bits[7:0] in Register 0x3C, is a dc value that is added directly to the sample values written to the DAC.	0x00	RW

IDAC DC OFFSET 1 REGISTER

Address: 0x3C, Reset: 0x00, Name: IDAC_DC_OFFSET1

Table 80. Bit Descriptions for IDAC_DC_OFFSET1

Bits	Bit Name	Settings	Description	Reset	Access
[7:0]	IDAC_DC_OFFSET_MSB		DAC dc offset, Bits[7:0] along with Bits[7:0] in Register 0x3B, is a dc value that is added directly to the sample values written to the DAC.	0x00	RW

QDAC DC OFFSET 0 REGISTER

Address: 0x3D, Reset: 0x00, Name: QDAC_DC_OFFSET0

Table 81. Bit Descriptions for QDAC_DC_OFFSET0

Bits	Bit Name	Settings	Description	Reset	Access
[7:0]	QDAC_DC_OFFSET_LSB		QDAC dc offset, Bits[7:0] along with Bits[7:0] in Register 0x3E, is a dc value that is added directly to the sample values written to the QDAC.	0x00	RW

QDAC DC OFFSET 1 REGISTER

Address: 0x3E, Reset: 0x00, Name: QDAC_DC_OFFSET1

Table 82. Bit Descriptions for QDAC_DC_OFFSET1

Bits	Bit Name	Settings	Description	Reset	Access
[7:0]	QDAC_DC_OFFSET_MSB		QDAC dc offset, Bits[7:0] along with Bits[7:0] in Register 0x3D, is a dc value that is added directly to the sample values written to the QDAC.	0x00	RW

IDAC GAIN ADJUST REGISTER

Address: 0x3F, Reset: 0x20, Name: IDAC_GAIN_ADJ

Table 83. Bit Descriptions for IDAC_GAIN_ADJ

Bits	Bit Name	Settings	Description	Reset	Access
[5:0]	IDAC_GAIN_ADJ		This register is the 6-bit digital gain adjust on the I channel. The bit weighting is MSB = 2^0, LSB = 2^{-5}, which yields a multiplier range of 0 to 2 or $-\infty$ to 6 dB. The default gain setting is 0x20, which maps to unity gain (0 dB).	0x20	RW

QDAC GAIN ADJUST REGISTER

Address: 0x40, Reset: 0x20, Name: QDAC_GAIN_ADJ

Table 84. Bit Descriptions for QDAC_GAIN_ADJ

Bits	Bit Name	Settings	Description	Reset	Access
[5:0]	QDAC_GAIN_ADJ		This register is the 6-bit digital gain adjust on the Q channel. The bit weighting is MSB = 2^0, LSB = 2^{-5}, which yields a multiplier range of 0 to 2 or −∞ to 6 dB. The default gain setting is 0x20, which maps to unity gain (0 dB).	0x20	RW

GAIN STEP CONTROL 0 REGISTER

Address: 0x41, Reset: 0x01, Name: GAIN_STEP_CTRL0

Table 85. Bit Descriptions for GAIN_STEP_CTRL0

Bits	Bit Name	Settings	Description	Reset	Access
[5:0]	RAMP_UP_STEP		This register sets the step size of the increasing gain. The digital gain increases by the configured amount in every four DAC cycles until the gain reaches the setting in IDAC_GAIN_ADJ (Register 0x3F). The bit weighting is MSB = 2^1, LSB = 2^{-4}. Note that the value in this register must not be greater than the values in the IDAC_GAIN_ADJ.	0x01	RW

GAIN STEP CONTROL 1 REGISTER

Address: 0x42, Reset: 0x41, Name: GAIN_STEP_CTRL1

Table 86. Bit Descriptions for GAIN_STEP_CTRL1

Bits	Bit Name	Settings	Description	Reset	Access
7	DAC_OUTPUT_OFF		This bit allows for turning the DAC output on and off manually. The digital IQ gain function (Register 0x27, Bit 5) must be turned on for this bit to function.	0x0	RW
6	DAC_OUTPUT_STATUS		This bit indicates the DAC output on/off status. When the DAC output is turned off, this bit is 1. Upon power-up, this bit is 1. The digital IQ gain function (Register 0x27, Bit 5) must be turned on for this bit to track the on/off status	0x1	R
[5:0]	RAMP_DOWN_STEP		This register sets the step size of the decreasing gain. The digital gain decreases by the configured amount in every four DAC cycles until the gain reaches zero. The bit weighting is MSB = 2^1, LSB = 2^{-4}. Note that the value in this register must not be greater than the values in the IDAC_GAIN_ADJ (Register 0x3F).	0x01	RW

TX ENABLE CONTROL REGISTER

Address: 0x43, Reset: 0x07, Name: TX_ENABLE_CTRL

Table 87. Bit Descriptions for TX_ENABLE_CTRL

Bits	Bit Name	Settings	Description	Reset	Access
2	TXENABLE_GAINSTEP_EN		DAC output gradually turns on/off under the control of the TXENABLE signal from the TXEN pin according to the settings in Register 0x41 and Register 0x42.	1	RW
1	TXENABLE_SLEEP_EN		When set to 1, the device is put in sleep mode when the TXENABLE signal from the TXEN pin is low.	1	RW
0	TXENABLE_POWER_DOWN_EN		When set to 1, the device is put in power down mode when the TXENABLE signal from the TXEN pin is low.	1	RW

DAC OUTPUT CONTROL REGISTER

Address: 0x44, Reset: 0x8D, Name: DAC_OUTPUT_CTRL

Table 88. Bit Descriptions for DAC_OUTPUT_CTRL

Bits	Bit Name	Settings	Description	Reset	Access
7	DAC_OUTPUT_CTRL_EN		Enables the DAC output control. This bit needs to be set to 1 to enable the remaining bits in this register.	0x1	RW
3	FIFO_WARNING_SHUTDOWN_EN		When this bit and Bit 7 are both high, if a FIFO warning occurs, the DAC output shuts down automatically. By default, this function is on.	0x1	RW
2	OVERTHRESHOLD_SHUTDOWN_EN		The DAC output is turned off when the input average power is greater than the predefined threshold.	0x1	RW
0	FIFO_ERROR_SHUTDOWN_EN		The DAC output is turned off when the FIFO reports warnings.	0x1	RW

DLL CELL ENABLE 0 REGISTER

Address: 0x5E, Reset: 0xFF, Name: ENABLE_DLL_DELAY_CELL0

Table 89. Bit Descriptions for ENABLE_DLL_DELAY_CELL0

Bits	Bit Name	Description	Reset	Access
[7:0]	DELAY_CELL_ENABLE [7:0]	Set each bit to enable or disable the delay cell. Delay cell number corresponds to bit number. 1 = enable delay cell (default). 0 = disable delay cell. Different recommended values should be used in DLL mode and delay line mode. See the Data Interface section.	0xFF	RW

DLL CELL ENABLE 1 REGISTER

Address: 0x5F, Reset: 0x67, Name: ENABLE_DLL_DELAY_CELL1

Table 90. Bit Descriptions for ENABLE_DLL_DELAY_CELL1

Bits	Bit Name	Description	Reset	Access
[7:3]	Reserved	Must write the default value for optimal performance.	0x0C	RW
[2:0]	DELAY_CELL_ENABLE [10:8]	Set each bit to enable or disable the delay cell. Delay cell numbers are 10, 9, 8 corresponding to bits Bit, Bit 2, and Bit 0, respectively. 1 = enable delay cell (default). 0 = disable delay cell.	0x7	RW

SED CONTROL REGISTER

Address: 0x60, Reset: 0x00, Name: SED_CTRL

Table 91. Bit Descriptions for SED_CTRL

Bits	Bit Name	Description	Reset	Access
7	SED_ENABLE	Set to 1 to Enable the SED compare logic.	0	RW
6	SED_ERR_CLEAR	When set to 1, clears all SED reported error bits, Bit 2, Bit 1, and Bit 0.	0	RW
5	AED_ENABLE	When set to 1, enables the AED function (SED with auto clear after eight passing sets).	0	RW
4	SED_DEPTH	0 = SED depth of two words, 1 = SED depth of four words.	0	RW
3	Reserved	Reserved.	0	R
2	AED_PASS	When AED = 1, it signals eight true compare cycles.	0	RW
1	AED_FAIL	When AED = 1, it signals a mismatch in comparison.	0	R
0	SED_FAIL	Signals that an SED mismatch in comparison occurred (with SED or AED enabled).	0	R

SED PATTERN I0 LOW BITS REGISTER

Address: 0x61, Reset: 0x00, Name: SED_PATT_L_I0

Table 92. Bit Descriptions for SED_PATT_L_I0

Bits	Bit Name	Description	Reset	Access
[7:0]	SED_PATTERN_RISE_I0[7:0]	SED I0 rising edge low bits.	0x00	RW

SED PATTERN I0 HIGH BITS REGISTER

Address: 0x62, Reset: 0x00, Name: SED_PATT_H_I0

Table 93. Bit Descriptions for SED_PATT_H_I0

Bits	Bit Name	Description	Reset	Access
[7:0]	SED_PATTERN_RISE_I0[15:8]	SED I0 rising edge high bits.	0x00	RW

SED PATTERN Q0 LOW BITS REGISTER

Address: 0x63, Reset: 0x00, Name: SED_PATT_L_Q0

Table 94. Bit Descriptions for SED_PATT_L_Q0

Bits	Bit Name	Description	Reset	Access
[7:0]	SED_PATTERN_FALL_Q0[7:0]	SED Q0 falling edge low bits.	0x00	RW

SED PATTERN Q0 HIGH BITS REGISTER

Address: 0x64, Reset: 0x00, Name: SED_PATT_H_Q0

Table 95. Bit Descriptions for SED_PATT_H_Q0

Bits	Bit Name	Description	Reset	Access
[7:0]	SED_PATTERN_FALL_Q0[15:8]	SED Q0 falling edge high bits.	0x00	RW

SED PATTERN I1 LOW BITS REGISTER

Address: 0x65, Reset: 0x00, Name: SED_PATT_L_I1

Table 96. Bit Descriptions for SED_PATT_L_I1

Bits	Bit Name	Description	Reset	Access
[7:0]	SED_PATTERN_RISE_I1[7:0]	SED I1 rising edge low bits.	0x00	RW

SED PATTERN I1 HIGH BITS REGISTER

Address: 0x66, Reset: 0x00, Name: SED_PATT_H_I1

Table 97. Bit Descriptions for SED_PATT_H_I1

Bits	Bit Name	Description	Reset	Access
[2:0]	SED_PATTERN_RISE_I1[15:8]	SED I1 rising edge high bits.	0x00	RW

SED PATTERN Q1 LOW BITS REGISTER

Address: 0x67, Reset: 0x00, Name: SED_PATT_L_Q1

Table 98. Bit Descriptions for SED_PATT_L_Q1

Bits	Bit Name	Description	Reset	Access
[7:0]	SED_PATTERN_FALL_Q1[7:0]	SED Q1 falling edge low bits.	0x00	RW

SED PATTERN Q1 HIGH BITS REGISTER

Address: 0x68, Reset: 0x00, Name: SED_PATT_H_Q1

Table 99. Bit Descriptions for SED_PATT_H_Q1

Bits	Bit Name	Description	Reset	Access
[2:0]	SED_PATTERN_FALL_Q1[15:8]	SED Q1 falling edge high bits.	0x00	RW

PARITY CONTROL REGISTER

Address: 0x6A, Reset: 0x00, Name: PARITY_CTRL

Table 100. Bit Descriptions for PARITY_CTRL

Bits	Bit Name	Settings	Description	Reset	Access
7	PARITY_ENABLE	1	Enable parity.	0	RW
6	PARITY_EVEN	0 1	Odd parity. Even parity.	0	RW
5	PARITY_ERR_CLEAR		Set to 1 to clear parity error counters.	0	RW
[4:2]	Reserved		Reserved.	0x0	R
1	PARERRFAL		When 1, signals a falling edge parity error was detected.	0	R
0	PARERRRISE		When 1, signals a rising edge parity error was detected.	0	R

PARITY ERROR RISING EDGE REGISTER

Address: 0x6B, Reset: 0x00, Name: PARITY_ERR_RISING

Table 101. Bit Descriptions for PARITY_ERR_RISING

Bits	Bit Name	Description	Reset	Access
[7:0]	Parity Rising Edge Error Count	Number of rising edge-based errors detected (S0 and S2). Clipped to 256.	0x00	R

PARITY ERROR FALLING EDGE REGISTER

Address: 0x6C, Reset: 0x00, Name: PARITY_ERR_FALLING

Table 102. Bit Descriptions for PARITY_ERR_FALLING

Bits	Bit Name	Description	Reset	Access
[7:0]	Parity Falling Edge Error Count	Number of falling edge-based errors detected (S1 and S3). Clipped to 256.	0x00	R

VERSION REGISTER

Address: 0x7F, Reset: 0x0B, Name: Version

Table 103. Bit Descriptions for Version

Bits	Bit Name	Settings	Description	Reset	Access
[7:0]	Version		Chip version.	0x0B	R

DAC LATENCY AND SYSTEM SKEWS

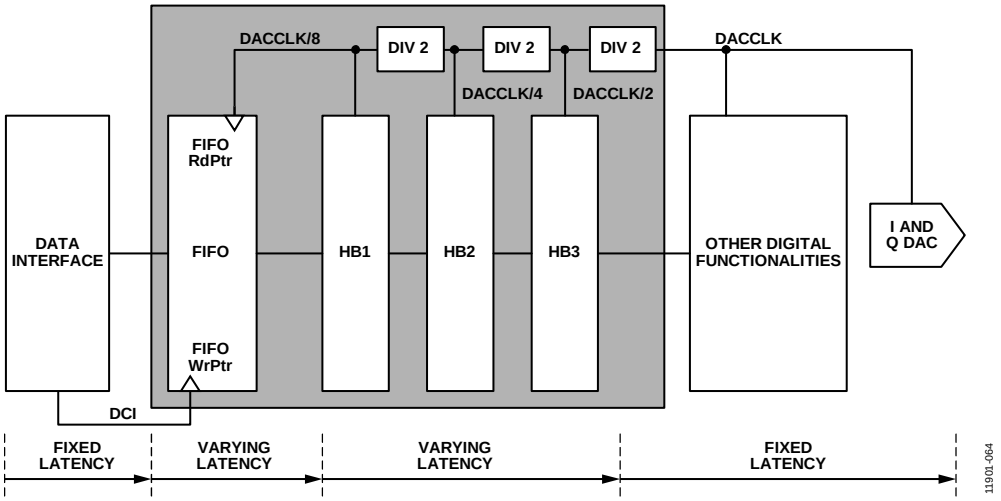


Figure 62. Breakdown of Pipeline Latencies

DAC LATENCY VARIATIONS

DACs, like any other devices with internal multiphase clocks, have an inherent pipeline latency variation. Figure 62 shows the delineation of pipeline latencies in the AD9142A. The highlighted section, including the FIFO and the clock generation circuitry, is where the pipeline latencies vary. Upon each power-on, the status of both the FIFO and the clock generation state machine is arbitrary. This leads to varying latency in these two blocks.

FIFO LATENCY VARIATION

There are eight data slots in the FIFO. The FIFO read and write pointers circulate the FIFO from Slot 0 to Slot 7 and back to Slot 0. The FIFO depth is defined as the number of FIFO slots that are required for the read pointer to catch the write pointer. It is also the time a particular piece of data stays in the FIFO from the point that it is written into the FIFO to the point where it is read out from the FIFO. Therefore, the latency of the FIFO is equivalent to its depth.

Figure 63 is an example of FIFO latency variation. The latency in Case 2 is two data cycles longer than that in Case 1. If other latencies are the same, the skew between the DAC outputs in these two cases is, likewise, two data cycles. Therefore, to keep a constant FIFO latency, the FIFO depth needs to be reset to a pre-defined value. Theoretically, any value other than 0 is valid but typically it is set to 4 to maximize the capacity of absorbing the rate fluctuation between the read and write sides.

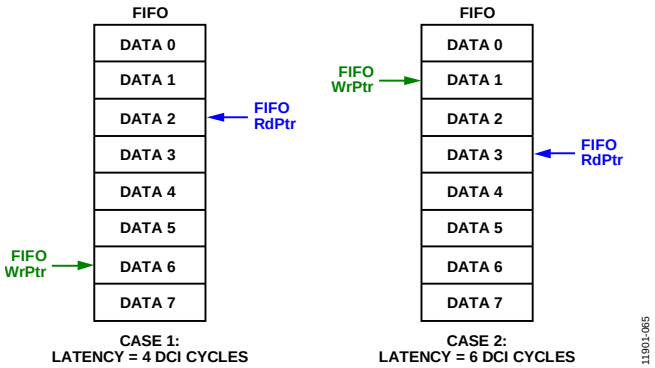


Figure 63. Example of FIFO Latency Difference

Figure 64 shows two equivalent cases of FIFO latency of four data cycles. Although neither the read nor the write pointer match each other in these two cases, the FIFO depth is the same in both cases. Also, note that the beginning slots of the data stream in the two cases are not the same, but the read and write pointers point to the same piece of data in both cases. This does not affect the alignment accuracy of the DAC outputs as long as the data and the DCIs are well aligned at multiple devices.

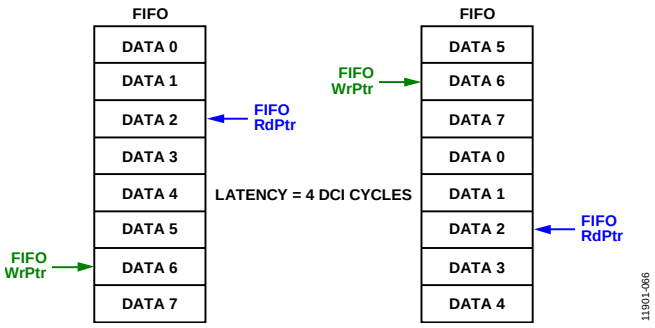


Figure 64. Example of Equal FIFO Latencies

CLOCK GENERATION LATENCY VARIATION

The state machine of the clock generation circuitry is another source of latency variations; this type of latency variation results from inherent phase uncertainty of the static frequency dividers. The divided down clock can be high or low at the rising edge of the input clock, unless specifically forced to a known state. This means that whenever there is interpolation (when slower clocks must be internally generated by dividing down the DACCLK), there is an inherent latency variation in the DAC. Figure 65 is an example of this latency variation in 2× interpolation.

There are two phase possibilities in the DACCLK/2 clock. The DACCLK/2 clock is used to read data from the FIFO and to drive the interpolation filter. Regardless of which clock edge is used to drive the digital circuit, there is a latency of one DAC clock cycle between Case 1 and Case 2 (see Figure 65). Because the power-on state arbitrarily falls in one of the two cases, the phase uncertainty of the divider appears as a varying skew between two DAC outputs.

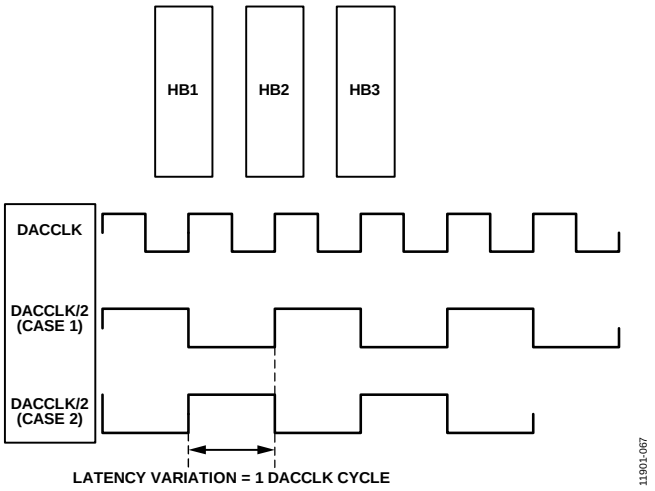


Figure 65. Latency Variation in 2× Interpolation from Clock Generation

CORRECTING SYSTEM SKEWS

Generally, it is assumed that the input data and the DCI among multiple devices are well aligned to each other. Depending on the system design, the data and DCI being input into each DAC can originate from various FPGAs or ASICs. Without synchronizing the data sources, the output of one data source can be skewed from that of another. The alignment between multiple data sources can also drift over temperature.

Figure 66 shows an example of a 2-channel transmitter with two data sources and two dual DACs. A constant but unknown phase offset appears between the outputs of the DAC devices, even if the DAC does not introduce any latency variations. The multidevice synchronization in the AD9142A can be used to compensate the skew due to misalignment of the data sources by resetting the two sides of the FIFO independently through two external reference clocks: the frame and the sync clock. The offset between the two data sources is then absorbed by the FIFO and clock generation block in the DAC. For more information about using the multidevice synchronization function, refer to the Synchronization Implementation section.

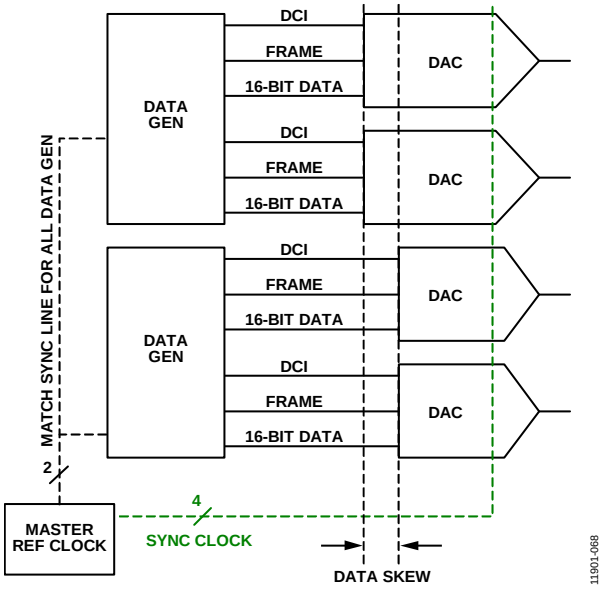
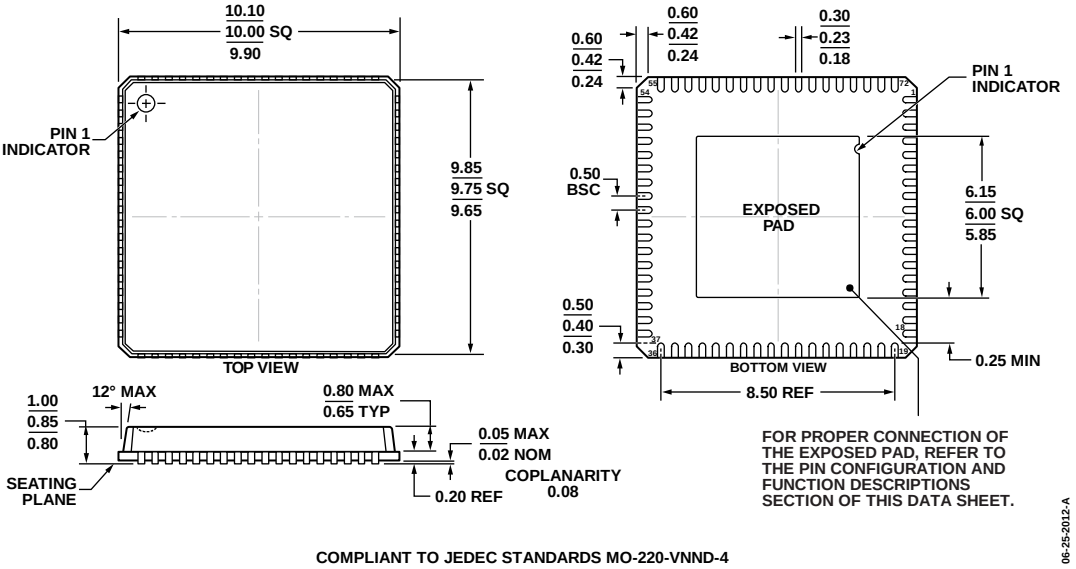


Figure 66. DAC Output Skew from Skewed Input Data and DCI

PACKAGING AND ORDERING INFORMATION

OUTLINE DIMENSIONS



COMPLIANT TO JEDEC STANDARDS MO-220-VNND-4

Figure 67. 72-Lead Lead Frame Chip Scale Package [LFCSP_VQ]
10 mm × 10 mm Body, Very Thin Quad
(CP-72-7)
Dimensions shown in millimeters

ORDERING GUIDE

Model[1]	Temperature Range	Package Description	Package Option
AD9142ABCPZ	−40°C to +85°C	72-Lead LFCSP_VQ	CP-72-7
AD9142ABCPZRL	−40°C to +85°C	72-Lead LFCSP_VQ	CP-72-7
AD9142A-M5372-EBZ		Evaluation Board Connected to ADL5372 Modulator	
AD9142A-M5375-EBZ		Evaluation Board Connected to ADL5375 Modulator	

[1] Z = RoHS Compliant Part.

D11901-0-5/14(A)