

FEATURES

Very small inherent latency variation: <2 DAC clock cycles
Proprietary low spurious and distortion design
6-carrier GSM ACLR = 79 dBc at 200 MHz IF
SFDR > 85 dBc (bandwidth = 300 MHz) at ZIF
Flexible 16-bit LVDS interface
Supports word and byte load
Multiple chip synchronization
Fixed latency and data generator latency compensation
Selectable 2x, 4x, 8x interpolation filter
Low power architecture
fs/4 power saving coarse mixer
Input signal power detection
Emergency stop for downstream analog circuitry protection
FIFO error detection
On-chip numeric control oscillator allows carrier placement anywhere in the DAC Nyquist bandwidth
Transmit enable function for extra power saving
High performance, low noise PLL clock multiplier
Digital gain and phase adjustment for sideband suppression
Digital inverse sinc filter
Supports single DAC mode
Low power: 2.0 W at 1.6 GSPS, 1.7 W at 1.25 GSPS, full operating conditions
72-lead LFCSP

APPLICATIONS

Wireless communications: 3G/4G and MC-GSM base stations, wideband repeaters, software defined radios
Wideband communications: point-to-point, LMDS/MMDS
Transmit diversity/MIMO
Instrumentation
Automated test equipment

GENERAL DESCRIPTION

The AD9142 is a dual, 16-bit, high dynamic range digital-to-analog converter (DAC) that provides a sample rate of 1600 MSPS, permitting a multicarrier generation up to the Nyquist frequency. The AD9142 TxDAC+® includes features optimized for direct conversion transmit applications, including complex digital modulation, input signal power detection, and gain, phase, and offset compensation. The DAC outputs are optimized to interface seamlessly with analog quadrature modulators, such as the ADL537x F-MOD series and the ADRF670x series from Analog Devices, Inc. A 3-wire serial port interface provides for the programming/readback of many internal parameters. Full-scale output current can be programmed over a range of 9 mA to 33 mA. The AD9142 is available in a 72-lead LFCSP.

PRODUCT HIGHLIGHTS

1. Advanced low spurious and distortion design techniques provide high quality synthesis of wideband signals from baseband to high intermediate frequencies.
2. Very small inherent latency variation simplifies both software and hardware design in the system. It allows easy multichip synchronization for most applications.
3. New low power architecture improves power efficiency (mW/MHz/channel) by 30%.
4. Input signal power and FIFO error detection simplify designs for downstream analog circuitry protection.
5. Programmable transmit enable function allows easy design balance between power consumption and wakeup time.

AD9142* PRODUCT PAGE QUICK LINKS

Last Content Update: 09/27/2017

COMPARABLE PARTS

View a parametric search of comparable parts.

EVALUATION KITS

- AD9142 Evaluation Board

DOCUMENTATION

Application Notes

- AN-1342: AD9142 to AD9142A Migration

Data Sheet

- AD9142: Dual, 16-Bit, 1600 MSPS, TxDAC+ Digital-to-Analog Converter Data Sheet

TOOLS AND SIMULATIONS

- AD9142 IBIS Model

REFERENCE MATERIALS

Informational

- Advantiv™ Advanced TV Solutions

Technical Articles

- MS-2739: High Dynamic IF Receiver Simplifies Design of Next Generation μ W Point-to-Point Modems

DESIGN RESOURCES

- AD9142 Material Declaration
- PCN-PDN Information
- Quality And Reliability
- Symbols and Footprints

DISCUSSIONS

View all AD9142 EngineerZone Discussions.

SAMPLE AND BUY

Visit the product page to see pricing options.

TECHNICAL SUPPORT

Submit a technical question or find your regional support number.

DOCUMENT FEEDBACK

Submit feedback for this data sheet.

TABLE OF CONTENTS

| | | | |
|--|----|---|----|
| Features | 1 | Input Signal Power Detection and Protection | 30 |
| Applications | 1 | Transmit Enable Function | 31 |
| General Description | 1 | Digital Function Configuration | 31 |
| Product Highlights | 1 | Multidevice Synchronization and Fixed Latency | 32 |
| Revision History | 3 | Very Small Inherent Latency Variation | 32 |
| Functional Block Diagram | 4 | Further Reducing the Latency Variation | 32 |
| Specifications | 5 | Synchronization Implementation | 33 |
| DC Specifications | 5 | Synchronization Procedures | 33 |
| Digital Specifications | 6 | Interrupt Request Operation | 34 |
| DAC Latency Specifications | 7 | Interrupt Working Mechanism | 34 |
| Latency Variation Specifications | 7 | Interrupt Service Routine | 34 |
| AC Specifications | 7 | Temperature Sensor | 35 |
| Operating Speed Specifications | 8 | DAC Input Clock Configurations | 36 |
| Absolute Maximum Ratings | 9 | Driving the DACCLK and REFCLK Inputs | 36 |
| Thermal Resistance | 9 | Direct Clocking | 36 |
| ESD Caution | 9 | Clock Multiplication | 36 |
| Pin Configuration and Function Descriptions | 10 | PLL Settings | 37 |
| Typical Performance Characteristics | 12 | Configuring the VCO Tuning Band | 37 |
| Terminology | 17 | Automatic VCO Band Select | 37 |
| Serial Port Operation | 18 | Manual VCO Band Select | 37 |
| Data Format | 18 | Analog Outputs | 38 |
| Serial Port Pin Descriptions | 18 | Transmit DAC Operation | 38 |
| Serial Port Options | 18 | Interfacing to Modulators | 39 |
| Data Interface | 20 | Reducing LO Leakage and Unwanted Sidebands | 40 |
| LVDS Input Data Ports | 20 | Example Start-Up Routine | 41 |
| Word Interface Mode | 20 | Device Configuration Register Map and Description | 42 |
| Byte Interface Mode | 20 | SPI Configure Register | 44 |
| Data Interface Configuration Options | 20 | Power-Down Control Register | 44 |
| Interface Delay Line | 22 | Interrupt Enable0 Register | 44 |
| FIFO Operation | 23 | Interrupt Enable1 Register | 44 |
| Resetting the FIFO | 24 | Interrupt Flag0 Register | 45 |
| Serial Port Initiated FIFO Reset | 24 | Interrupt Flag1 Register | 45 |
| Frame Initiated FIFO Reset | 24 | Interrupt Select0 Register | 45 |
| Digital Datapath | 26 | Interrupt Select1 Register | 46 |
| Interpolation Filters | 26 | DAC Clock Receiver Control Register | 46 |
| Digital Modulation | 28 | Ref Clock Receiver Control Register | 46 |
| Datapath Configuration | 29 | PLL Control Register | 47 |
| Digital Quadrature Gain and Phase Adjustment | 29 | PLL Control Register | 47 |
| DC Offset Adjustment | 29 | PLL Control Register | 47 |
| Inverse Sinc Filter | 30 | PLL Status Register | 48 |

| | | | |
|--|----|--|----|
| PLL Status Register | 48 | NCO_PHASE_OFFSET1 Register..... | 55 |
| IDAC FS Adjust LSB Register..... | 48 | IQ_PHASE_ADJ0 Register..... | 55 |
| IDAC FS Adjust MSB Register | 48 | IQ_PHASE_ADJ1 Register..... | 55 |
| QDAC FS Adjust LSB Register | 48 | IDAC_DC_OFFSET0 Register..... | 55 |
| QDAC FS Adjust MSB Register | 49 | IDAC_DC_OFFSET1 Register..... | 55 |
| Die Temperature Sensor Control Register..... | 49 | QDAC_DC_OFFSET0 Register..... | 55 |
| Die Temperature LSB Register | 49 | QDAC_DC_OFFSET1 Register..... | 56 |
| Die Temperature MSB Register..... | 49 | IDAC_GAIN_ADJ Register..... | 56 |
| Chip ID Register..... | 49 | QDAC_GAIN_ADJ Register..... | 56 |
| Interrupt Configuration Register | 50 | Gain Step Control0 Register..... | 56 |
| Sync CTRL Register | 50 | Gain Step Control1 Register..... | 56 |
| Frame Reset CTRL Register..... | 50 | TX Enable Control Register | 57 |
| FIFO Level Configuration Register | 51 | DAC Output Control Register | 57 |
| FIFO Level Readback Register | 51 | Data Receiver Test Control Register..... | 57 |
| FIFO CTRL Register..... | 51 | Data Receiver Test Control Register..... | 57 |
| Data Format Select Register..... | 52 | Device Configuration0 Register..... | 58 |
| Datapath Control Register | 52 | Version Register | 58 |
| Interpolation Control Register | 52 | Device Configuration1 Register..... | 58 |
| Over Threshold CTRL0 Register | 53 | Device Configuration2 Register..... | 58 |
| Over Threshold CTRL1 Register | 53 | DAC Latency and System Skews..... | 59 |
| Over Threshold CTRL2 Register | 53 | DAC Latency Variations..... | 59 |
| Input Power Readback LSB Register | 53 | FIFO Latency Variation..... | 59 |
| Input Power Readback MSB Register..... | 53 | Clock Generation Latency Variation..... | 60 |
| NCO Control Register..... | 54 | Correcting System Skews..... | 60 |
| NCO_FREQ_TUNING_WORD0 Register..... | 54 | Packaging and Ordering Information | 61 |
| NCO_FREQ_TUNING_WORD1 Register..... | 54 | Outline Dimensions..... | 61 |
| NCO_FREQ_TUNING_WORD2 Register..... | 54 | Ordering Guide | 61 |
| NCO_FREQ_TUNING_WORD3 Register..... | 54 | | |
| NCO_PHASE_OFFSET0 Register..... | 54 | | |

REVISION HISTORY

11/12—Revision 0: Initial Version

FUNCTIONAL BLOCK DIAGRAM

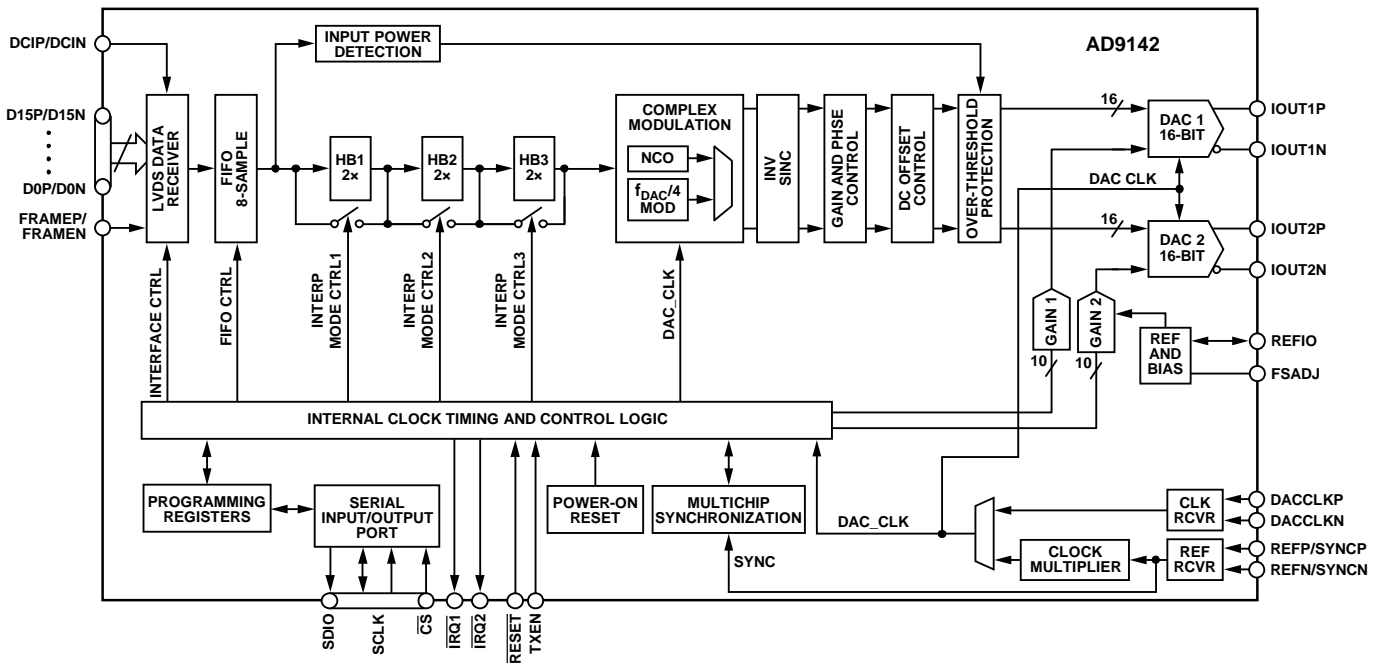


Figure 1.

109301-001

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 | | 1.71 | 1.8 | 1.89 | V |
| DIGITAL SUPPLY VOLTAGES | | | | | |
| DVDD18 | | 1.71 | 1.8 | 1.89 | V |
| POWER CONSUMPTION | | | | | |
| 2× Mode | $f_{DAC} = 491.52$ MSPS | | | | |
| NCO OFF | | | 700 | | mW |
| NCO ON | | | 870 | | mW |
| 4× Mode | $f_{DAC} = 737.28$ MSPS | | | | |
| NCO OFF | | | 836 | | mW |
| NCO ON | | | 1085 | | mW |
| 4× Mode | $f_{DAC} = 983.04$ MSPS | | | | |
| NCO OFF | | | 1030 | | mW |
| NCO ON | | | 1365 | | mW |
| 8× Mode | $f_{DAC} = 1600$ MSPS | | | | |
| NCO OFF | | | 1315 | | mW |
| NCO ON | | | 1815 | | mW |
| Phase-Lock Loop | | | 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 |

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 | | | | | | |
| Input Voltage Range | V_{IA} or V_{IB} | | 825 | | 1675 | mV |
| Input Differential Threshold | V_{IDTH} | Data and FRAME inputs DCI input | -100 | | +100 | mV |
| | | | -225 | | +225 | mV |
| Input Differential Hysteresis | V_{IDTHH} to V_{IDTHL} | | | 20 | | mV |
| Receiver Differential Input Impedance | R_{IN} | | | 120 | | Ω |
| DAC UPDATE RATE | | | | | 1600 | MSPS |
| DAC Adjusted Update Rate | | 2x interpolation | | | 250 | 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/SYCN) | | | | | | |
| Differential Peak-to-Peak Voltage | | | 100 | 500 | 2000 | mV |
| Common-Mode Voltage | | | | 1.25 | | V |
| Input Clock Frequency | | $1 \text{ GHz} \leq f_{VCO} \leq 2.1 \text{ GHz}$ | | | 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 |
| Setup Time | t_{DS} | SDIO to SCLK | 1.5 | | | ns |
| Hold Time | t_{DH} | SDIO to SCLK | 0.68 | | | ns |
| Setup Time | t_{DCSB} | \overline{CS} to SCLK | 2.38 | 1.4 | | ns |

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 SPECIFICATIONS¹**Table 4.**

| Parameter | Min | Typ | Max | Unit |
|-----------------------|-----|-----|-----|---------------|
| DAC LATENCY VARIATION | | | | |
| SYNC Off | | | 2 | DACCLK cycles |
| SYNC On | | | 1 | DACCLK cycles |

¹ DAC latency is defined as the elapsed time from a data sample clocked at the input to the AD9142 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 = 360MHz | | | 85 | | dBc |
| $f_{DAC} = 1228.8$ MSPS | $f_{OUT} = 280$ MHz | | | | |
| BW = 200MHz | | | 85 | | dBc |
| BW = 500MHz | | | 75 | | dBc |
| $f_{DAC} = 1474.56$ MSPS | $f_{OUT} = 10$ MHz | | | | |
| BW = 737MHz | $f_{OUT} = 280$ MHz | | 85 | | dBc |
| BW = 400MHz | | | 80 | | dBc |
| TWO-TONE INTERMODULATION DISTORTION (IMD) | −6 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 |

| Parameter | Test Conditions/Comments | Min | Typ | Max | Unit |
|---|--|-----|----------------|-----|------------------|
| NOISE SPECTRAL DENSITY (NSD) f _{DAC} = 737.28 MSPS f _{DAC} = 983.04 MSPS f _{DAC} = 1228.8 MSPS f _{DAC} = 1474.56 MSPS | Eight-tone, 500 kHz tone spacing f _{OUT} = 200 MHz | | -160 | | dBm/Hz |
| | f _{OUT} = 200 MHz | | -161.5 | | dBm/Hz |
| | f _{OUT} = 280 MHz | | -164.5 | | dBm/Hz |
| | f _{OUT} = 10 MHz f _{OUT} = 280 MHz | | -166 -162.5 | | dBm/Hz dBm/Hz |
| W-CDMA ADJACENT CHANNEL LEAKAGE RATIO (ACLR) f _{DAC} = 983.04 MSPS f _{DAC} = 1228.8 MSPS f _{DAC} = 1474.56 MSPS | Single carrier f _{OUT} = 200 MHz | | 81 | | dBc |
| | f _{OUT} = 20 MHz | | 83 | | dBc |
| | f _{OUT} = 280 MHz | | 80 | | dBc |
| | f _{OUT} = 20 MHz f _{OUT} = 280 MHz | | 81 80 | | dBc dBc |
| W-CDMA SECOND (ACLR) f _{DAC} = 983.04 MSPS f _{DAC} = 1228.8 MSPS f _{DAC} = 1474.56 MSPS | Single carrier f _{OUT} = 200 MHz | | 85 | | dBc |
| | f _{OUT} = 20 MHz | | 86 | | dBc |
| | f _{OUT} = 280 MHz | | 86 | | dBc |
| | f _{OUT} = 20 MHz f _{OUT} = 280 MHz | | 86 85 | | dBc dBc |

OPERATING SPEED SPECIFICATIONS

Table 6.

| Interpolation Factor | DVDD18, CVDD18 = 1.8 V ± 5% | | DVDD18, CVDD18 = 1.8 V ± 2% or 1.9 V ± 5% | |
|----------------------|-----------------------------------|-----------------------------|---|-----------------------------|
| | f _{INTERFACE} (Mbps) Max | f _{DAC} (Mbps) Max | f _{INTERFACE} (Mbps) Max | f _{DAC} (Mbps) Max |
| 2× | 250 | 500 | 250 | 500 |
| 4× | 250 | 1000 | 250 | 1000 |
| 8× | 187.5 | 1500 | 200 | 1600 |

ABSOLUTE MAXIMUM RATINGS

Table 7.

| Parameter | Rating |
|---|--------------------------|
| AVDD33 to AVSS, EPAD, CVSS, DVSS | −0.3 V to +3.6 V |
| DVDD18, CVDD18 to AVSS, EPAD, CVSS, DVSS | −0.3 V to +2.1 V |
| AVSS to EPAD, CVSS, DVSS | −0.3 V to +0.3 V |
| EPAD to AVSS, CVSS, DVSS | −0.3 V to +0.3 V |
| CVSS to AVSS, EPAD, DVSS | −0.3 V to +0.3 V |
| DVSS to AVSS, EPAD, CVSS | −0.3 V to +0.3 V |
| FSADJ, REFIO, IOUT1P/IOUT1N, IOUT2P/IOUT2N to AVSS | −0.3 V to AVDD33 + 0.3 V |
| D[15:0]P/D[15:0]N, FRAMEP/FRAMEN, DCIP/DCIN to EPAD, DVSS | −0.3 V to DVDD18 + 0.3 V |
| DACCLKP/DACCLKN, REFP/SYNCP/REFN/SYCN to CVSS | −0.3 V to CVDD18 + 0.3 V |
| RESET, IRQ1, IRQ2, CS, SCLK, SDIO to EPAD, DVSS | −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} .

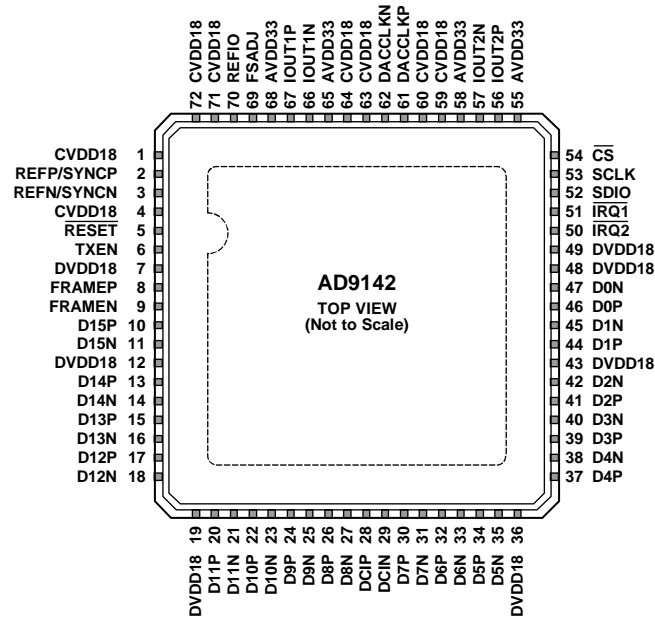
Table 8. Thermal Resistance

| 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



- NOTES
1. EXPOSED PAD (EPAD) MUST BE SOLDERED TO THE GROUND PLANE (AVSS). THE EPAD PROVIDES AN ELECTRICAL, THERMAL, AND MECHANICAL CONNECTION TO THE BOARD.
 2. EPAD IS THE GROUND CONNECTION FOR CVSS AND DVSS.

106930-002

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 Input, Positive. |
| 3 | REFN/SYNCN | PLL Reference 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 Register 0x43 in Table 77 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 | Frame Input, Positive. |
| 9 | FRAMEN | Frame 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. |
| 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. |
| 24 | D9P | Data Bit 9, Positive. |

| Pin No. | Mnemonic | Description |
|---------|--------------------------|--|
| 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 | <u>1.8 V Digital Supply.</u> 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 | <u>1.8 V Digital Supply.</u> 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 (LSB), Positive. |
| 47 | D0N | Data Bit 0 (LSB), Negative. |
| 48 | DVDD18 | <u>1.8 V Digital Supply.</u> Pin 48 supplies power to the digital core, digital data ports, serial port input/output pins, RESET, IRQ1, and IRQ2. |
| 49 | DVDD18 | <u>1.8 V Digital Supply.</u> 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 k Ω resistor. |
| 51 | $\overline{\text{IRQ1}}$ | First Interrupt Request. Open-drain, active low output. Connect an external pull-up to DVDD18 through a 10 k Ω 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 | IOOUT2P | QDAC Positive Current Output. |
| 57 | IOOUT2N | 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 | IOOUT1N | IDAC Negative Current Output. |
| 67 | IOOUT1P | IDAC Positive Current Output. |
| 68 | AVDD33 | 3.3 V Analog Supply. |
| 69 | FSADJ | Full-Scale Current Output Adjust. Place a 10 k Ω resistor from this pin to AVSS. |
| 70 | REFIO | Voltage Reference. Nominally 1.2 V output. Decouple REFIO to AVSS. |
| 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). The EPAD provides an electrical, thermal, and mechanical connection to the board. |

TYPICAL PERFORMANCE CHARACTERISTICS

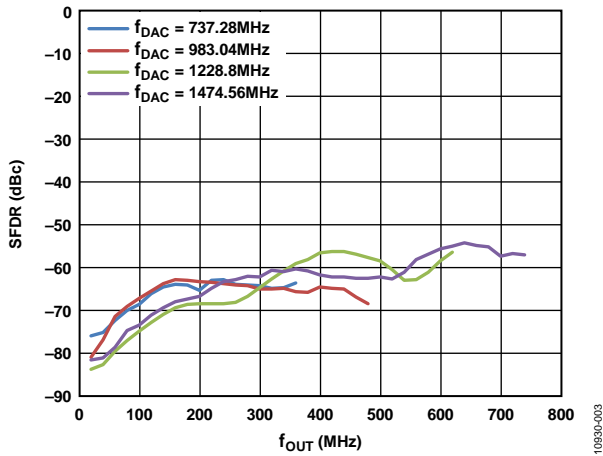


Figure 3. Single-Tone (0 dBFS) SFDR vs. f_{OUT} in the First Nyquist Zone over f_{DAC}

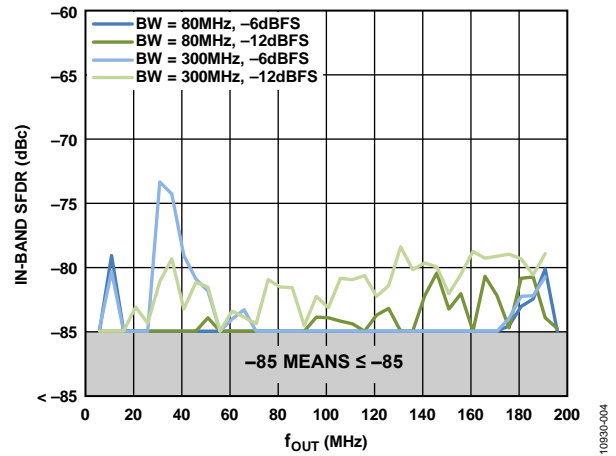


Figure 6. Single-Tone SFDR (Excluding 2nd Harmonic) vs. f_{OUT} in 80 MHz and 300 MHz Bandwidths, $f_{DAC} = 737.28$ MHz

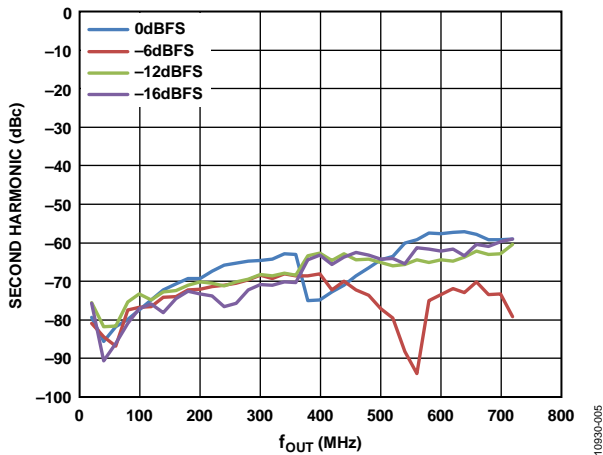


Figure 4. Single-Tone Second Harmonic vs. f_{OUT} in the First Nyquist Zone over Digital Back Off, $f_{DAC} = 1474.56$ MHz

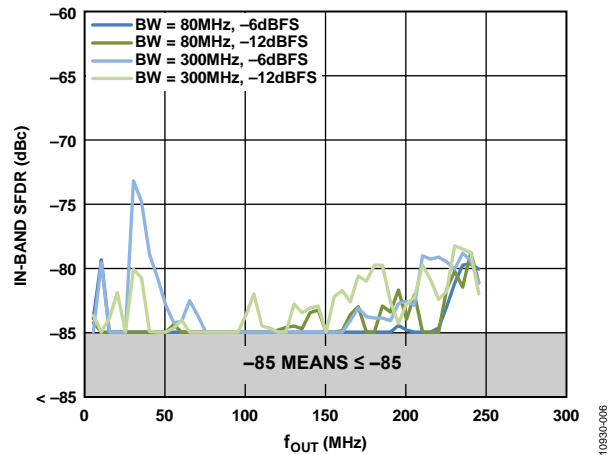


Figure 7. Single-Tone SFDR (Excluding 2nd Harmonic) vs. f_{OUT} in 80 MHz and 300 MHz BW, $f_{DAC} = 983.04$ MHz

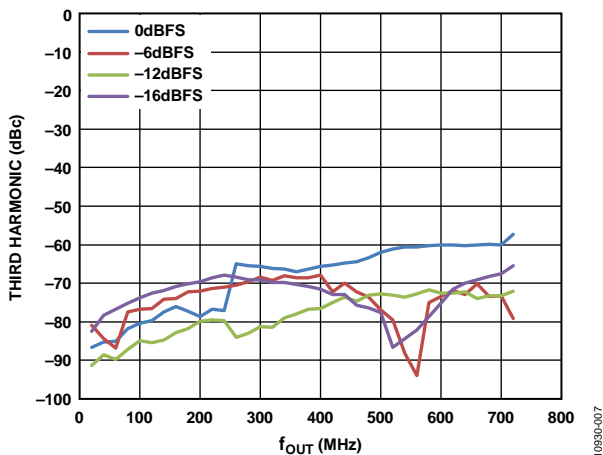


Figure 5. Single-Tone Third Harmonic vs. f_{OUT} in the First Nyquist Zone over Digital Back Off, $f_{DAC} = 1474.56$ MHz

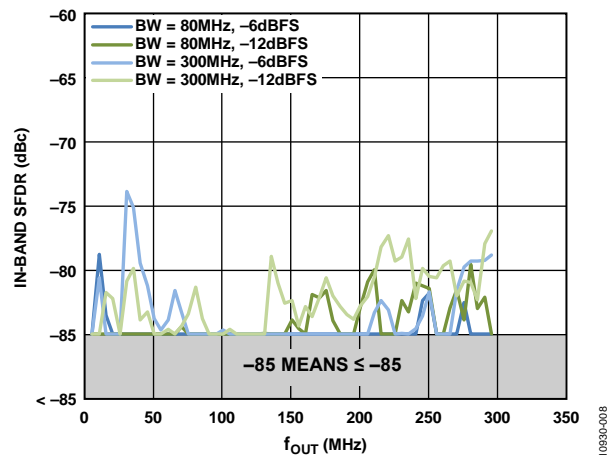


Figure 8. Single-Tone SFDR (Excluding 2nd Harmonic) vs. f_{OUT} in 80 MHz and 300 MHz Bandwidths, $f_{DAC} = 1228.8$ MHz

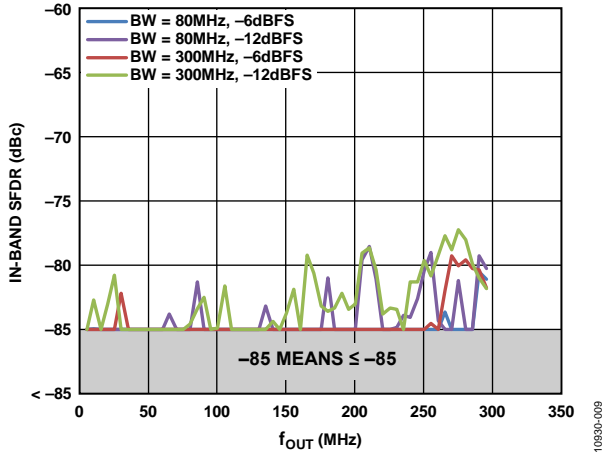


Figure 9. Single-Tone SFDR (Excluding 2nd Harmonic) vs. f_{OUT} in 80 MHz and 300 MHz Bandwidths, $f_{DAC} = 1474.56$ MHz

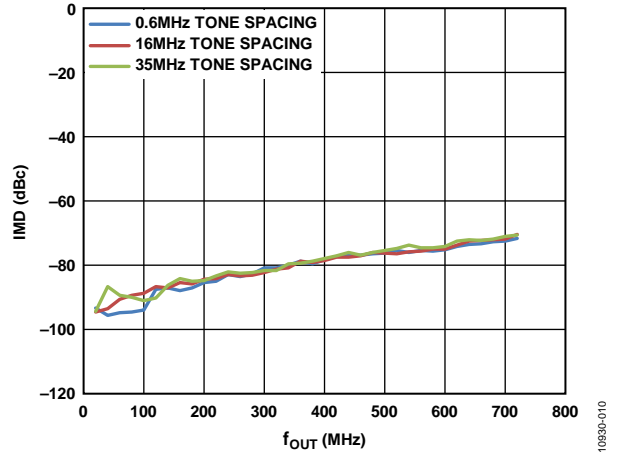


Figure 12. Two-Tone Third IMD vs. f_{OUT} over Tone Spacing, $f_{DAC} = 1474.56$ MHz

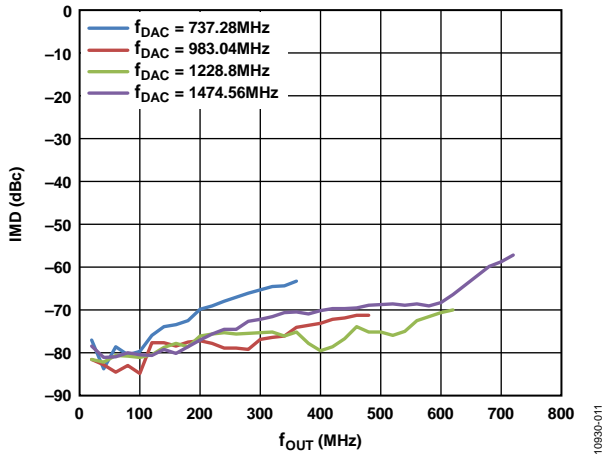


Figure 10. Two-Tone Third IMD vs. f_{OUT} over f_{DAC}

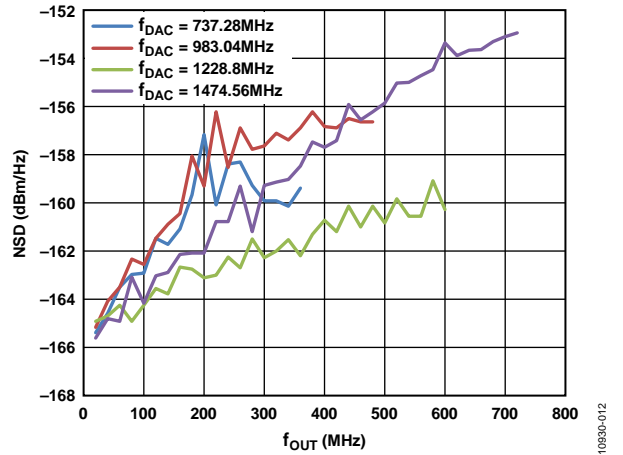


Figure 13. Single-Tone (0 dBFS) NSD vs. f_{OUT} over f_{DAC}

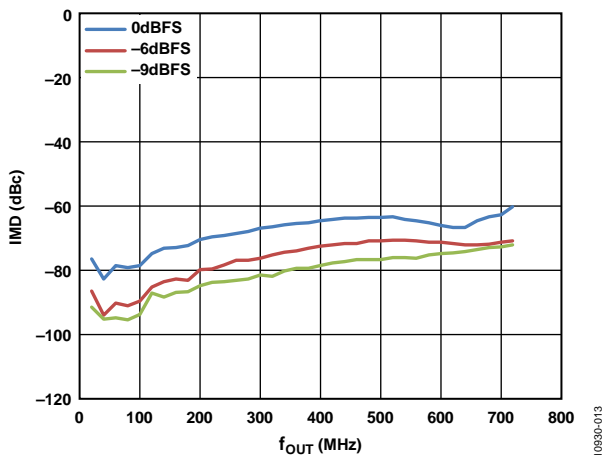


Figure 11. Two-Tone Third IMD vs. f_{OUT} over Digital Back Off, $f_{DAC} = 1474.56$ MHz

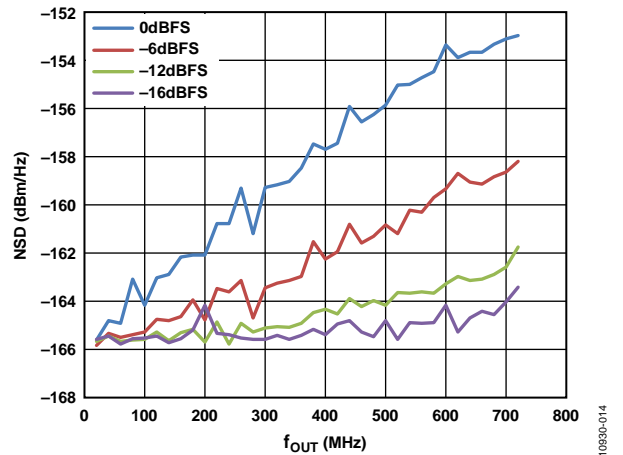


Figure 14. Single-Tone NSD vs. f_{OUT} over Digital Back Off, $f_{DAC} = 1474.56$ MHz

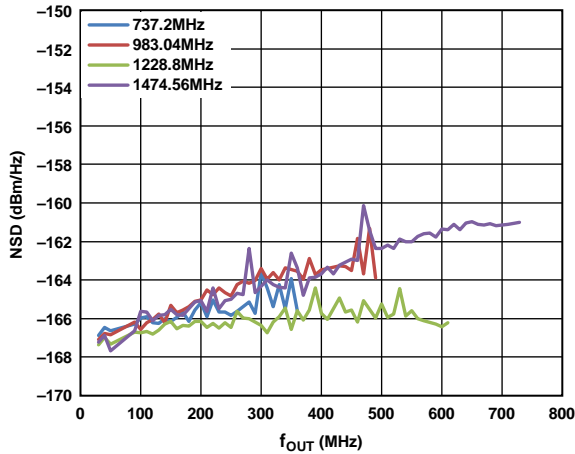


Figure 15. 1C WCDMA NSD vs. f_{OUT} , over f_{DAC}

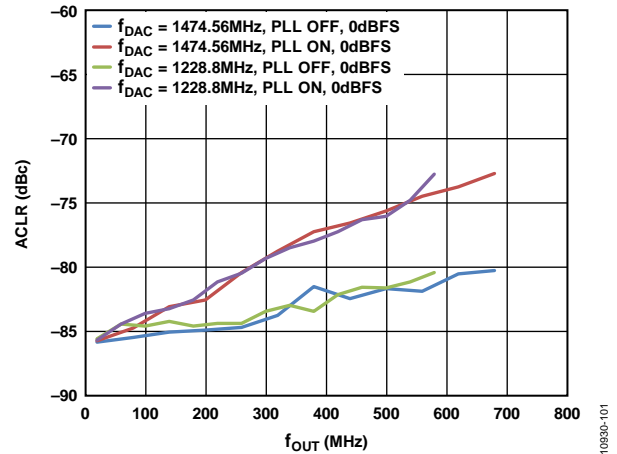


Figure 18. 1C WCDMA 2nd Adjacent ACLR vs. f_{OUT} , PLL On and Off

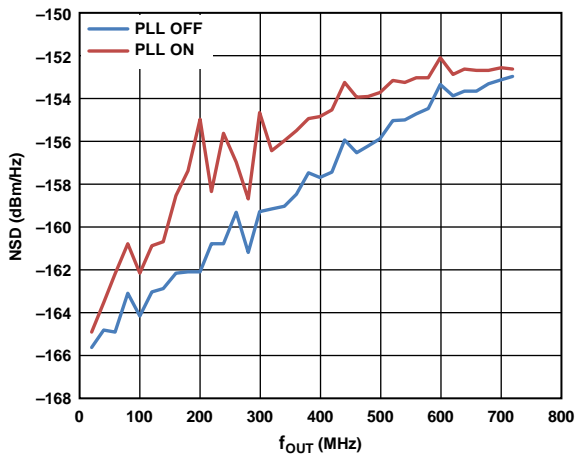


Figure 16. Single-Tone NSD vs. f_{OUT} , $f_{DAC} = 1474.28$ MHz, PLL On and Off

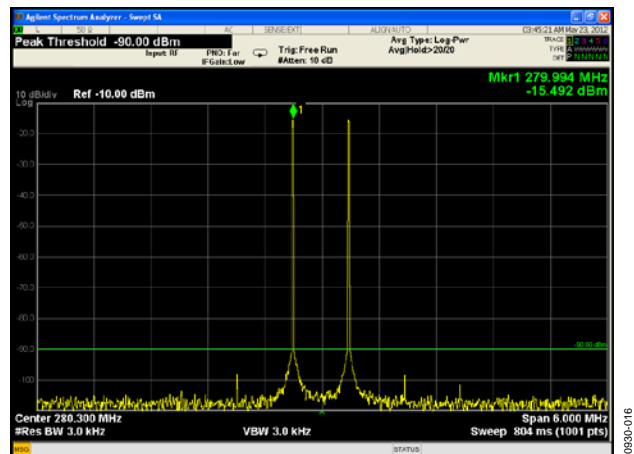


Figure 19. Two-Tone Third IMD Performance, $IF = 280$ MHz, $f_{DAC} = 1474.28$ MHz

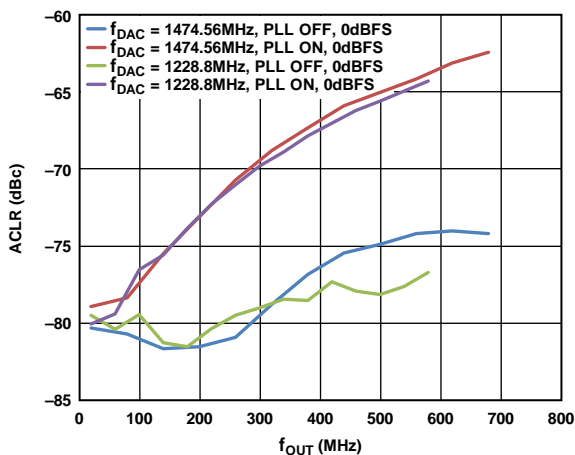


Figure 17. 1C WCDMA 1st Adjacent ACLR vs. f_{OUT} , PLL On and Off

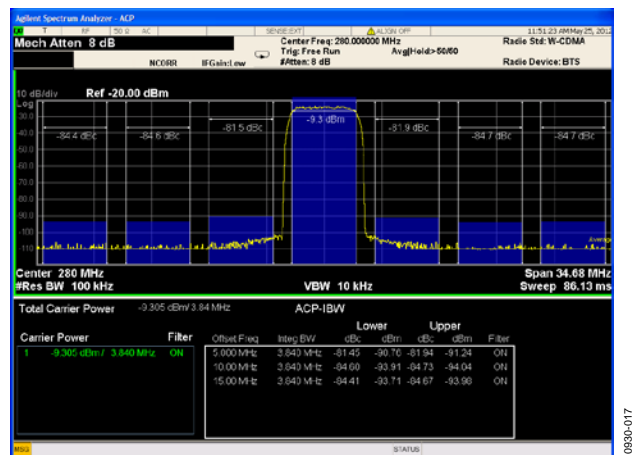


Figure 20. 1C WCDMA ACLR Performance, $IF = 280$ MHz, $f_{DAC} = 1474.28$ MHz

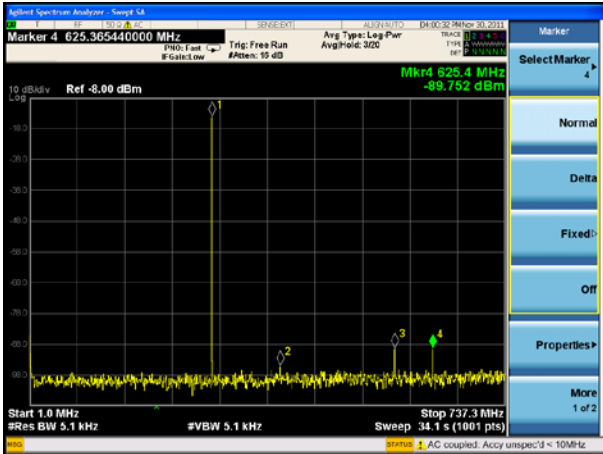


Figure 21. Single-Tone $f_{DAC} = 1474.56$ MHz, $f_{OUT} = 280$ MHz, -14 dBFS

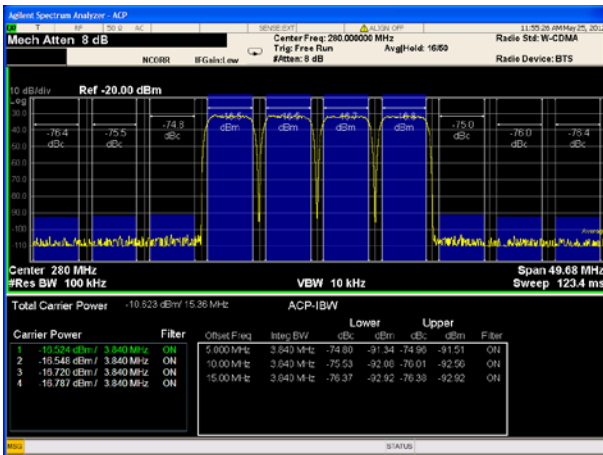


Figure 22. 4C WCDMA ACLR Performance, $IF = 280$ MHz, $f_{DAC} = 1474.28$ MHz

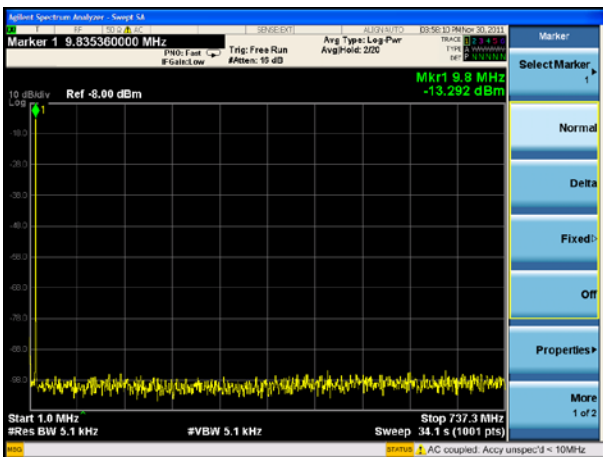


Figure 23. Single-Tone SFDR $f_{DAC} = 1474.56$ MHz, 4x Interpolation, $f_{OUT} = 10$ MHz, -14 dBFS

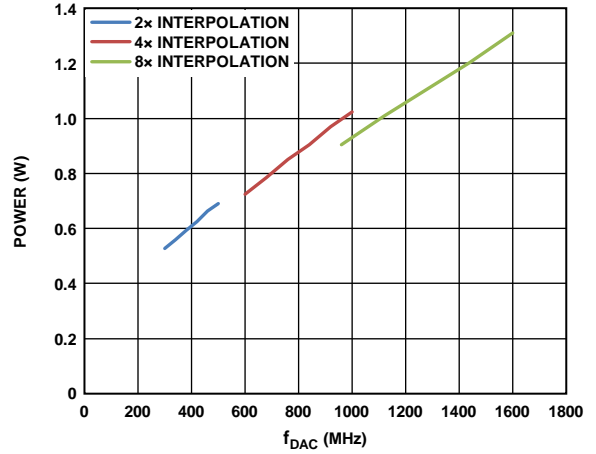


Figure 24. Total Power Consumption vs. f_{DAC} over Interpolation

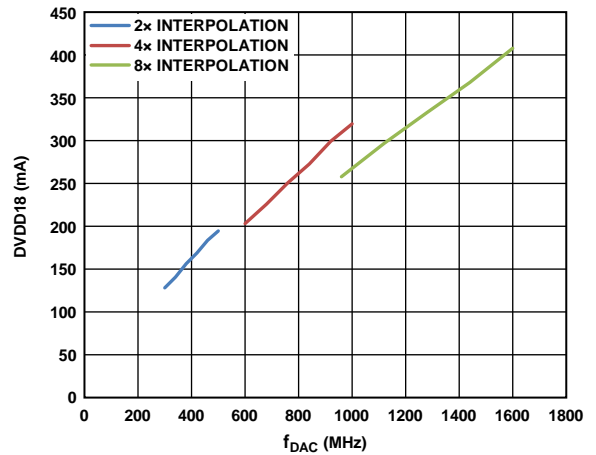


Figure 25. DVDD18 Current vs. f_{DAC} over Interpolation

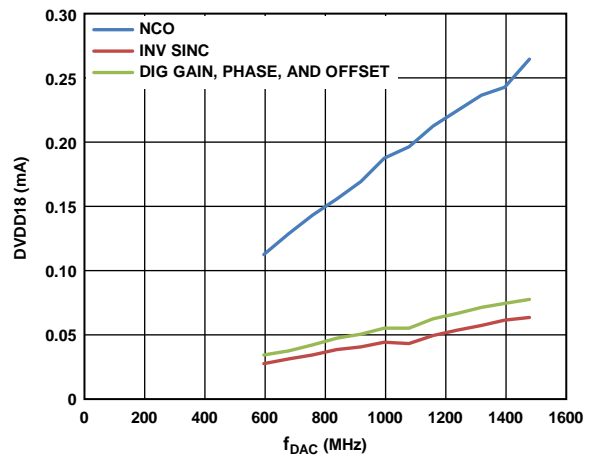


Figure 26. DVDD18 Current vs. f_{DAC} over Digital Functions

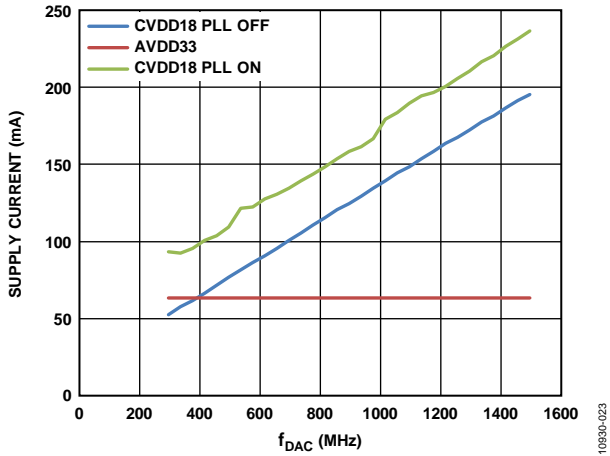


Figure 27. CVDD18, AVDD33 Current vs. f_{DAC}

106930-023

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 micro-controllers 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 AD9142. 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).

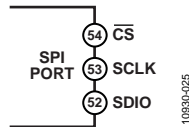


Figure 28. Serial Port Interface Pins

There are two phases to a communication cycle with the AD9142. 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 following data transfer.

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.

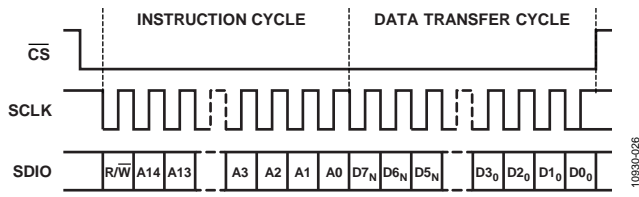


Figure 29. Serial Register Interface Timing, MSB First

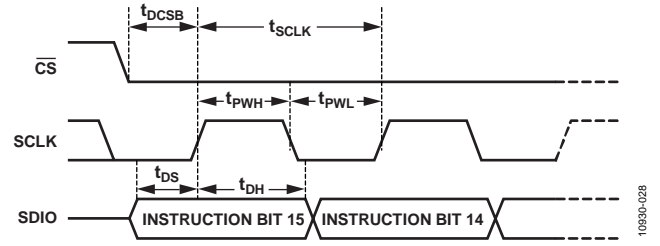


Figure 31. Timing Diagram for Serial Port Register Write

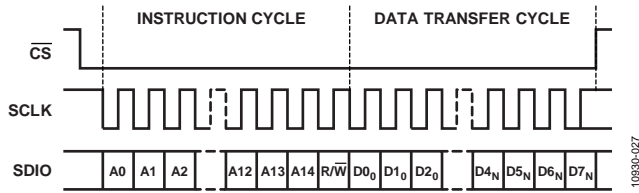


Figure 30. Serial Register Interface Timing, LSB First

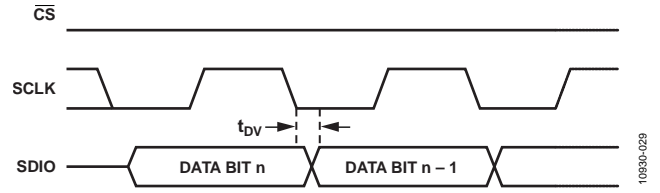


Figure 32. Timing Diagram for Serial Port Register Read

DATA INTERFACE

LVDS INPUT DATA PORTS

The AD9142 has a 16-bit LVDS bus that accepts 16-bit I and Q data either in word wide (16-bit) or byte wide (8-bit) formats. In the word wide interface mode, the data is sent over the entire 16-bit data bus. In the byte wide 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 Data Input 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 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.

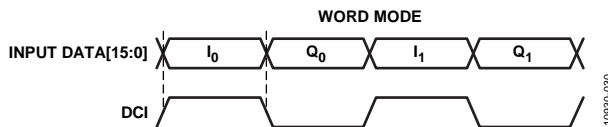


Figure 33. Timing Diagram for Word Mode

BYTE INTERFACE MODE

In byte 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 needs to hold 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.

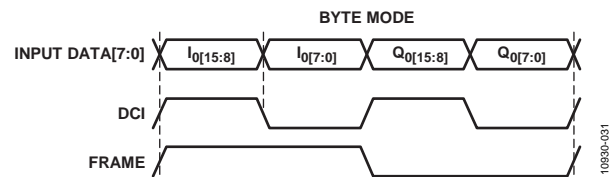


Figure 34. Timing Diagram for Byte Mode

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 | Function |
|-------------------------|--|
| 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]. |

LVDS Input Level Requirements

There are two types of LVDS receivers in the AD9142. The 16-bit data bus and the frame input share the same LVDS receiver design. The DCI uses a different LVDS design. The main difference between the two LVDS receivers is the required input differential swing level. The data bus and frame receiver require a minimum of ± 100 mV swing at the input. The DCI

receiver requires a minimum of ± 225 mV swing at its input. Figure 35 shows the LVDS input configuration and the required swing levels. Because the DCI is typically generated from the same bank as the data in the data source, it is recommended that the output swing of the LVDS driver be larger than the required DCI input level, thus meeting both input data and DCI requirements.

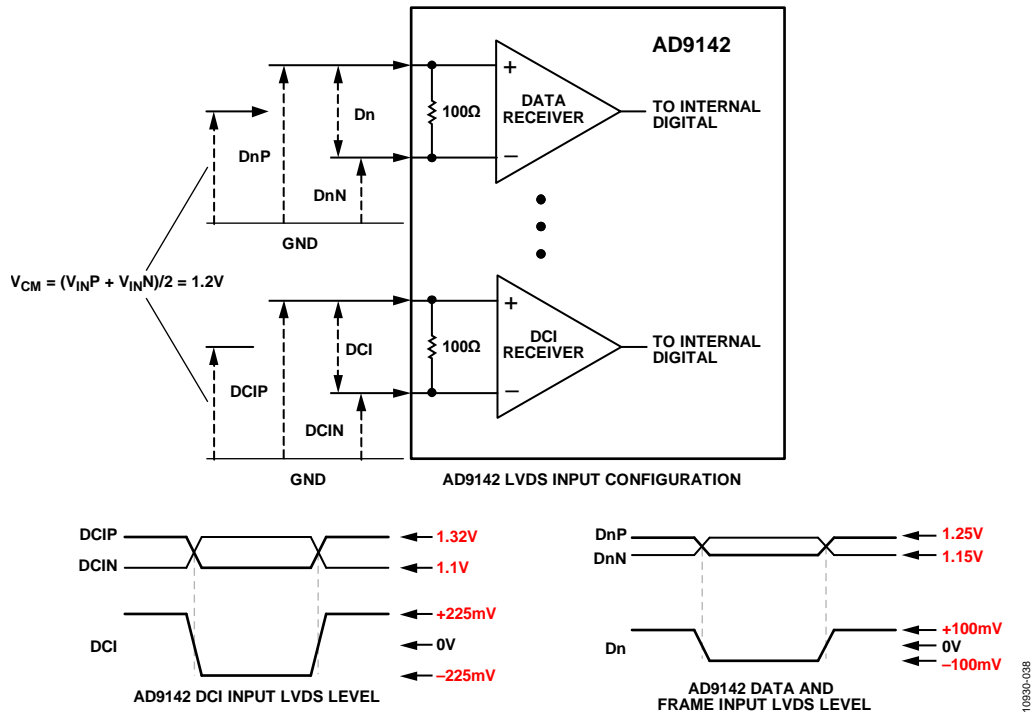


Figure 35. Data Interface Voltage Swing Requirements

INTERFACE DELAY LINE

A four-tap delay line is provided for the user to adjust the timing between the data bus and the DCI. Table 13 specifies the setup and hold times for each delay tap.

There is a fixed 1.9 ns delay on the DCI when the delay line is enabled. Each tap adds a nominal delay of 300 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 36 is an example of calculating the optimal external delay.

Table 13. Setup and Hold Times

| Delay Setting | 0 | 1 | 2 | 3 |
|-------------------------|-------|-------|-------|-------|
| Register 0x5E[7:0] | 0x00 | 0x07 | 0x7F | 0xFF |
| Register 0x5F[2:0] | 0x0 | 0x0 | 0x0 | 0x5 |
| t_s (ns) ¹ | -1.25 | -1.50 | -1.70 | -1.93 |
| t_H (ns) | 2.51 | 2.82 | 3.23 | 3.64 |
| $ t_s + t_H $ (ns) | 1.26 | 1.32 | 1.53 | 1.71 |

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.

Interface Timing Requirements

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

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

The shadow area in Figure 36 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

$$t_{DELAY} = \frac{(|t_s| + |t_H|)}{2} - \frac{t_{DATA PERIOD}}{2} = 1.88 - 1.25 = 0.63 \text{ ns}$$

SPI Sequence to Enable Delay Line-Based Mode

It is recommended that the following SPI sequence be used to enable the delay line-based mode:

1. 0x79 → 0x18 /* Configure Data Interface */
2. 0x5E → 0x00 /* Delay setting 0 */
0x5F → 0x00
3. 0x5F[3] → 1b /* Enable the delay line */

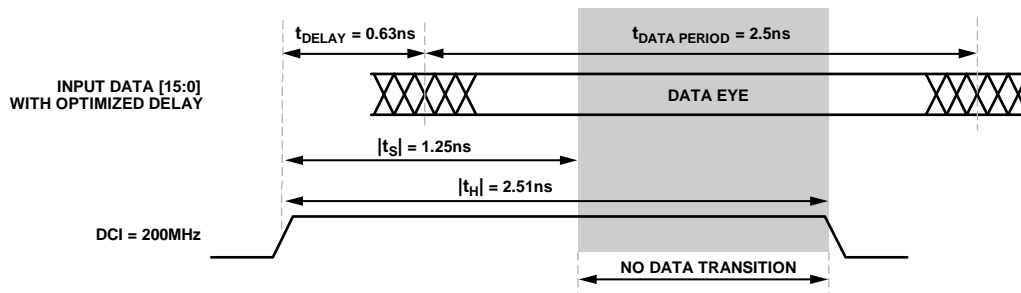


Figure 36. Example of Interfacing Timing in the Delay Line-Based Mode

10890-038

FIFO OPERATION

As is shown in the Data Interface section, the AD9142 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 AD9142, 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 AD9142 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 37 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 AD9142.

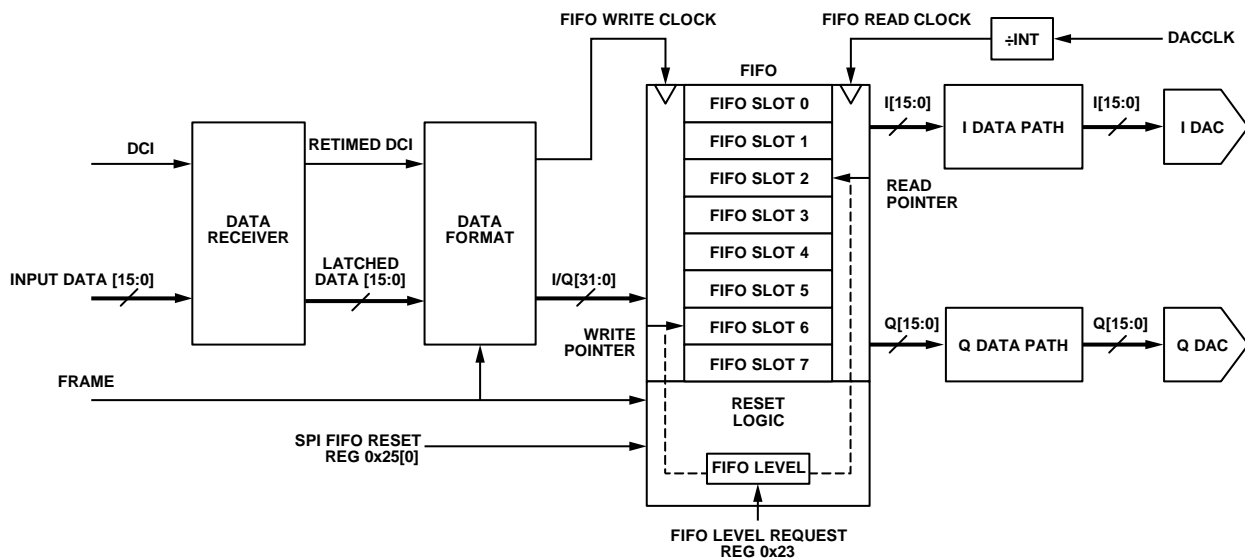


Figure 37. Block Diagram of FIFO

10830-040

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 document), which consists of two parts: the integral FIFO level and the fractional FIFO level.

The integer FIFO level represents the difference of the states between the read and write point in the unit of input data period ($1/f_{DATA}$). The fractional FIFO level represents the difference of the FIFO pointers 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

$$FIFO\ Latency = Integral\ Level + Fractional\ Level$$

Because the FIFO has eight data slots, there are eight possible FIFO integral levels. The maximum supported interpolation rate in the AD9142 is $8\times$ interpolation. Therefore, there are eight possible FIFO fractional levels. Two 3-bit registers in Register 0x23 are assigned to represent each level separately; Bits[6:4] represent the FIFO integral level and Bits[2:0] represent the FIFO fractional level. For example, if the interpolation rate is $4\times$ and the desired 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 $4\times$ interpolation. Table 14 shows additional examples of configuring the desired FIFO level in various interpolation rate modes.

Table 14. Examples of FIFO Level Configuration

| Interpolation Rate | Example FIFO Level ($1/f_{DATA}$) | Integer Level (Register 0x23[6:4]) | Fractional Level (Register 0x23[2:0]) |
|--------------------|-------------------------------------|------------------------------------|---------------------------------------|
| $2\times$ | $3 + 1/2$ | 3 | 1 |
| $4\times$ | $4 + 1/4$ | 4 | 1 |
| $8\times$ | $4 + 3/8$ | 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 $8\times$ interpolation, the maximum 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 $4\times$ 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 part 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 part 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 should be 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 mode and two DCI periods in byte 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. Program Register 0x23 to the customized value, if the desired value is not 0x40.
4. Configure the FRAME_RESET_MODE bits (Register 0x22[1:0]) to 00b.
5. Choose whether continuous or one shot mode is desired by writing 0 or 1 to EN_CON_FRAME_RESET (Register 0x22[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 FRAME_RESET_ACK, Register 0x22[3], to verify that the reset is complete.
8. 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 should be 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 should 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

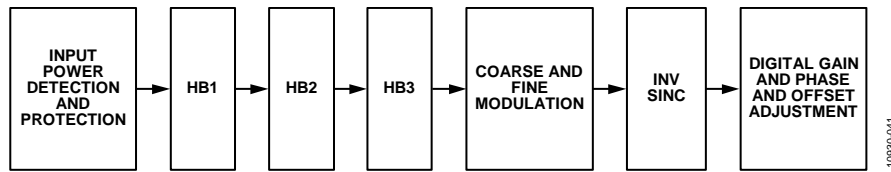


Figure 38. Block Diagram of Digital Datapath

The block diagram in Figure 38 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 AD9142 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 39 and Figure 40 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.

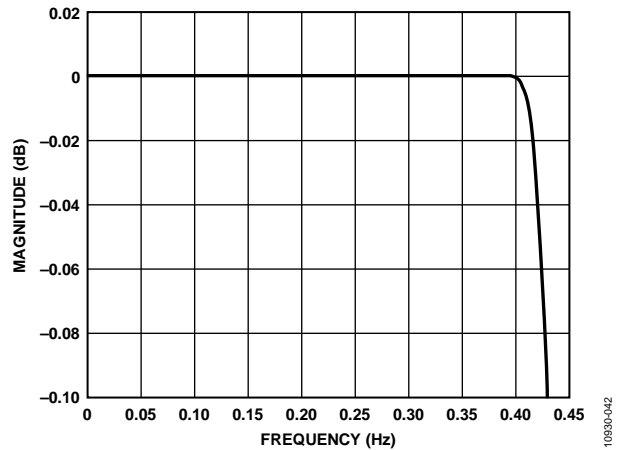


Figure 39. Pass-Band Detail of 2× Mode

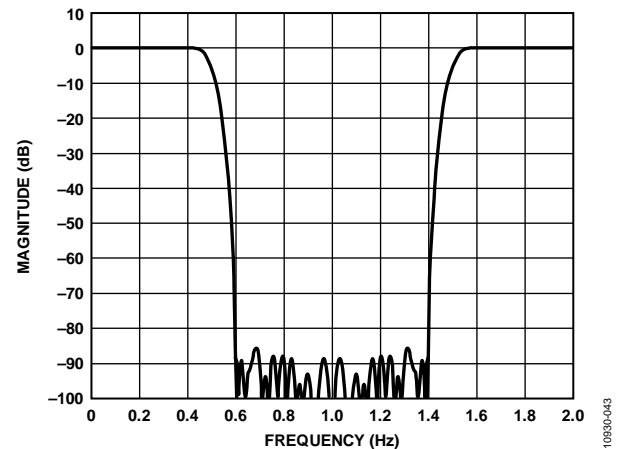


Figure 40. All-Band Response of 2× Mode

4× Interpolation Mode

Figure 41 and Figure 42 show the pass-band and all-band filter responses for 4× mode.

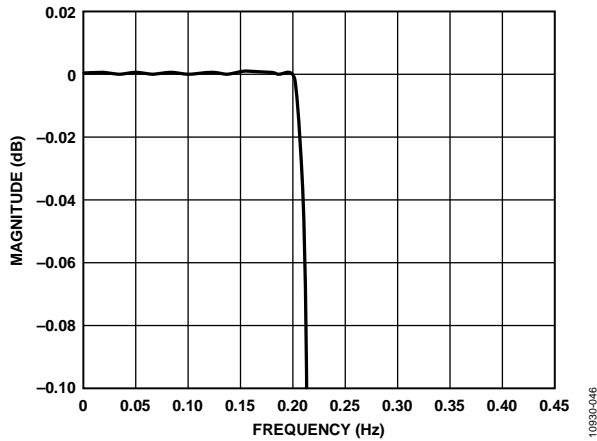


Figure 41. Pass-Band Detail of 4× Mode

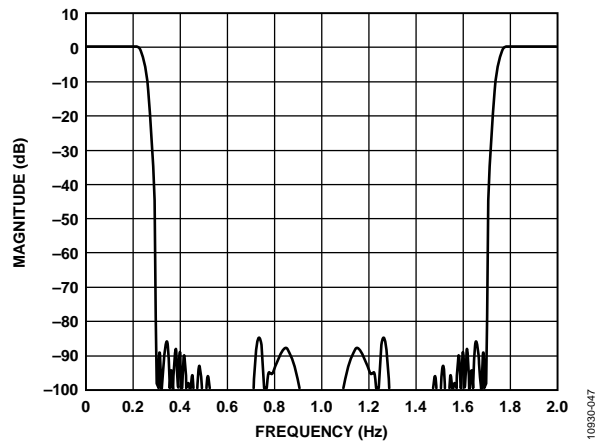


Figure 42. All-Band Response of 4× Mode

8× Interpolation Mode

Figure 43 and Figure 44 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).

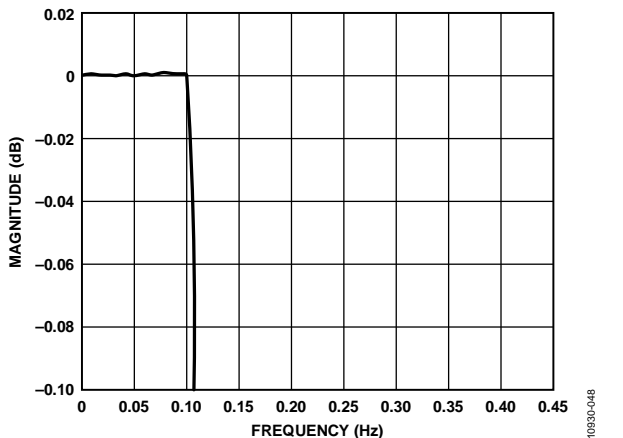


Figure 43. Pass-Band Detail of 8× Mode

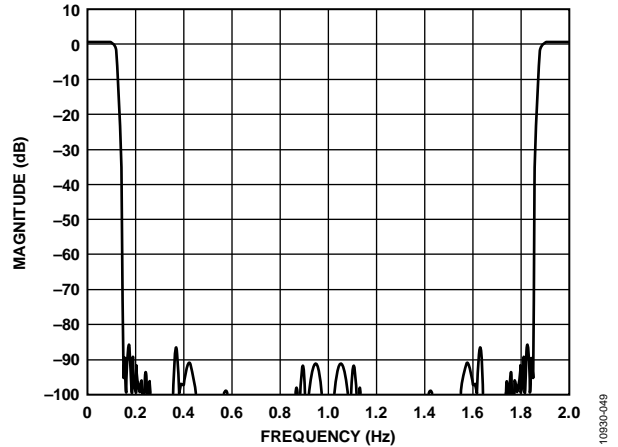


Figure 44. All-Band Response of 8× Mode

Table 15. 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 16. 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 17. 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 AD9142 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 45. The NCO modulation allows the DAC output signal to be placed anywhere in the output spectrum with very fine frequency resolution.

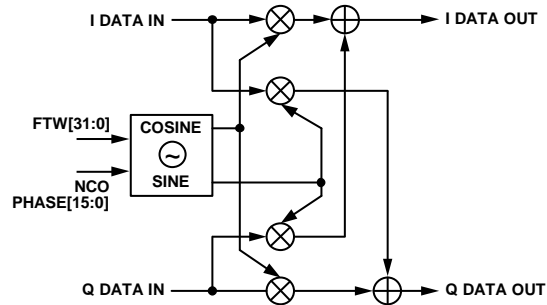


Figure 45. NCO Modulator Block Diagram

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 degrees from each other. The frequency of the complex carrier signal is set via NCO_FREQ_TUNING_WORD[31:0] 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

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

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

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

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 45.

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 needs to 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]) needs to be set in NCO only or FIFO and NCO, depending on whether 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 second step 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 AD9142 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 in configuring 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 $[-\infty, 6 \text{ dB}]$ 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 degrees between them. The quadrature phase adjustment is used to change the angle between the I and Q channels. IQ_PHASE_ADJ[12:0] (Register 0x37 and Register 0x38) provide an adjustment range of ± 14 degrees with a resolution of 0.0035 degrees. If the original angle is precisely 90 degrees, setting IQ_PHASE_ADJ[12:0] to 0x0FFF adds approximately 14 degrees between I and QDAC outputs, creating an angle of 104 degrees between the channels. Likewise, if the original angle is precisely 90 degrees, setting IQ_PHASE_ADJ[12:0] to 0x1000 adds approximately –14 degrees between the I and QDAC outputs, creating an angle of 76 degrees 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 16-bit registers, IDAC_DC_OFFSET, Bits[15:0] and QDAC_DC_OFFSET, Bits[15:0] (Register 0x3B through Register 0x3E). These values are added directly to the datapath values. Care should be taken not to overrange the transmitted values.

As shown in Figure 46, the DAC offset current varies as a function of the I/QDAC_DC_OFFSET values. Figure 46 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.

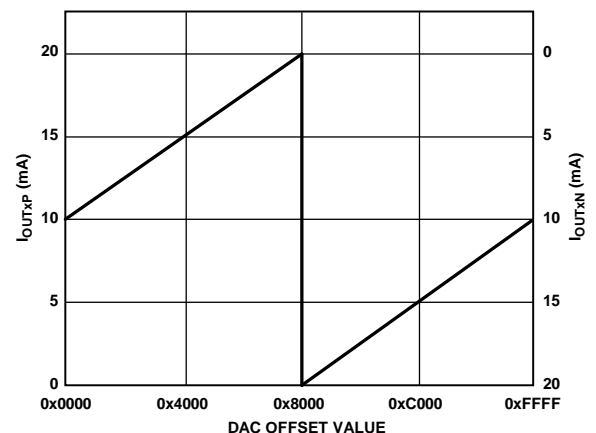


Figure 46. DAC Output Currents vs. DAC Offset Value

INVERSE SINC FILTER

The AD9142 provides a digital inverse sinc filter to compensate for the DAC rolloff over frequency. The inverse sinc (sinc^{-1}) filter is a seven tap FIR filter. Figure 47 shows the frequency response of $\sin(x)/x$ rolloff, 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 $0.4 \times f_{\text{DAC}}$.

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].

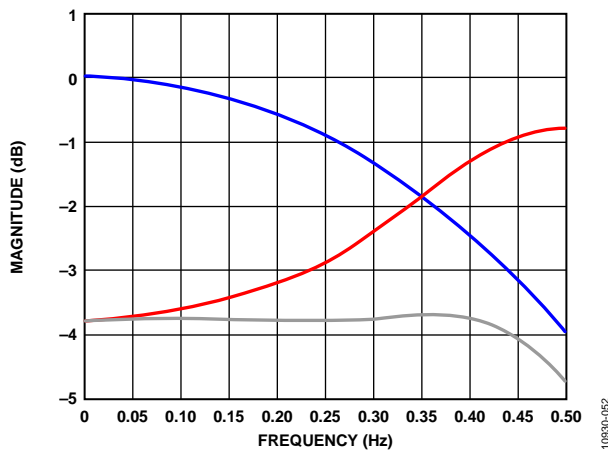


Figure 47. Responses of $\sin(x)/x$ Roll Off (Blue), the Sinc^{-1} Filter (Red), and the Composite of Both (Black)

Table 18. 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 AD9142 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 48 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, OVER_THRESHOLD_LEVEL[11:0] (Register 0x29 and Register 0x2A). 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.

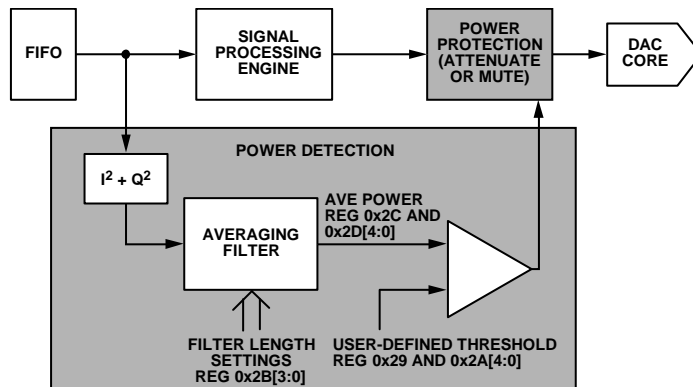


Figure 48. Block Diagram of Input Signal Power Detection and Protection Function

TRANSMIT ENABLE FUNCTION

The transmit enable (TXEN) function provides the user 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.

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 parts 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 re-transmit 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]. Although all of these actions can be taken through SPI writes, TXEN provides a much faster way to turn on and off the DAC output. The response time of a SPI write command is dominated by the SPI port communication time. This feature is useful when the user must turn off the DAC very quickly.

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 pre-distortion, where deterministic latency is desired, the variation of the pipeline latency must be minimized. Deterministic latency in this document 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 AD9142 minimizes the inherent latency variation. The worst-case variation in the AD9142 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 under any scenario. Therefore, without turning on the synchronization engine, the DAC outputs from multiple AD9142 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 start-up. Therefore, the AD9142 can decrease the complexity of system design in multi transmit channel applications.

Note the alignment of the DCI signals in the design. The DCI is used as a reference in the AD9142 design to align the FIFO and the phase of internal clocks in multiple parts. The achieved DAC output alignment depends on how well the DCIs are aligned at the input of each device. The equation below is the expression of the worst-case DAC output alignment accuracy in the case of DCI 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 AD9142 devices.

$t_{SK(DCI)}$ is the skew between two DCIs at the DCI input of the two AD9142 devices.

f_{DAC} is the DACCLK frequency.

The better the alignment of the DCIs, 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 AD9142 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 AD9142 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 19.

For users who want to synchronize the device in a one-shot manner and continue to monitor the synchronization status, the AD9142 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 19. Sync Clock and DAC Clock Setup and Hold Times

| Falling Edge Sync Timing (default) | Max | Unit |
|------------------------------------|-----|------|
| t_s (ns) | 246 | ps |
| t_h (ns) | -11 | ps |
| $ t_s + t_h $ (ns) | 235 | ps |

SYNCHRONIZATION IMPLEMENTATION

The AD9142 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. The sync clock, f_{SYNC} , should be $1/8 \times f_{\text{DATA}}$ 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 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 AD9142 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 mode and two DCI cycles in the byte mode. When the frame is a continuous clock, f_{FRAME} should be at $1/8 \times f_{\text{DATA}}$ or slower by a factor of $2n$, n being an integer (1, 2, 3...). Table 20 lists the requirements of the frame clock in various conditions.

Table 20. 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 mode = one DCI cycle and byte mode = two DCI cycles. |
| Continuous | $f_{\text{DATA}}/8$ | |

¹ N/A means not applicable.

SYNCHRONIZATION PROCEDURES

When the sync accuracy of an application is looser 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 finer than two-DAC clock cycle sync accuracy, 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 AD9142 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 FRAME_RESET_MODE.
 - 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 AD9142 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 AD9142 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 $f_{\text{DATA}}/8$, the same limit also applies to the reference clock. Therefore, only $2\times$ 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 FRAME_RESET_MODE.
 - 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 AD9142 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 AD9142 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 AD9142 provides an interrupt request output signal on Pin 50 and Pin 51 ($\overline{\text{IRQ2}}$ and $\overline{\text{IRQ1}}$, 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 $\overline{\text{IRQ1}}$ pin is an open-drain, active low output. Pull the $\overline{\text{IRQ1}}$ 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 $\overline{\text{IRQ1}}$ and $\overline{\text{IRQ2}}$ pin remain inactive.

INTERRUPT WORKING MECHANISM

Figure 49 shows the interrupt related circuitry and how the event flag signals propagate to the $\overline{\text{IRQx}}$ 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 $\overline{\text{IRQ}}$ 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 $\overline{\text{IRQ}}$ 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 $\overline{\text{IRQ2}}$ circuitry works in the same way as the $\overline{\text{IRQ1}}$ circuitry. Any one or multiple event flags can be enabled to trigger the $\overline{\text{IRQ1}}$ and $\overline{\text{IRQ2}}$ 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 $\overline{\text{IRQ1}}$ and set these registers to 1 for $\overline{\text{IRQ2}}$.

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 $\overline{\text{IRQx}}$ 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 can be monitored directly.
3. Perform any actions that may be required to clear the EVENT_FLAG_SOURCE . 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 .
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 and Description section.

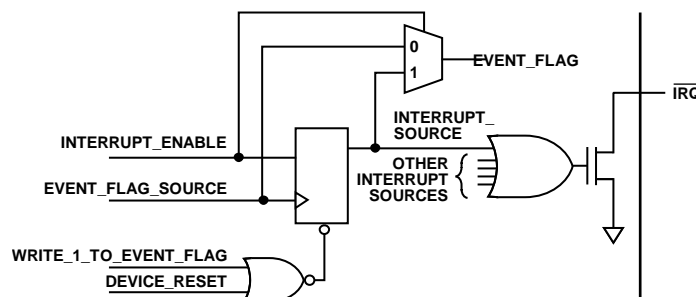


Figure 49. Simplified Schematic of $\overline{\text{IRQ}}$ Circuitry

TEMPERATURE SENSOR

The AD9142 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 $\pm 7^{\circ}\text{C}$ typical over the $+85^{\circ}\text{C}$ to -40°C range with one point temperature calibration against a known temperature. A typical plot of the die temperature vs. die temperature code readback is shown in Figure 50.

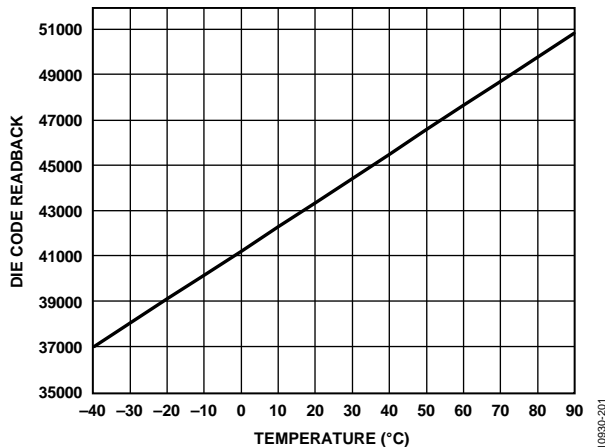


Figure 50. Die Temperature vs. Die Temperature Code Readback

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

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

where:

T_A is the ambient temperature in degrees Celsius.

θ_{JA} is the thermal resistance from junction to ambient of the AD9142 as shown in Table 8.

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

DAC INPUT CLOCK CONFIGURATIONS

The AD9142 DAC sample clock (DACCLK) can be sourced directly or by clock multiplying. Clock multiplying employs the on-chip phase-locked loop (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 DACCLK and REFCLK differential inputs share similar clock receiver input circuitry. Figure 51 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.

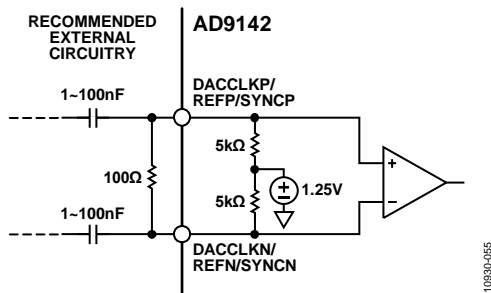


Figure 51. Clock Receiver Input Simplified Equivalent Circuit

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 REFCLK 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 REFCLK input and the DACCLK input is left floating. The functional diagram of the clock multiplier is shown in Figure 52.

The clock multiplication circuit operates such that the VCO outputs a frequency, f_{VCO} , equal to the REFCLK input signal frequency multiplied by $N1 \times N0$. $N1$ 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 1.0 GHz to 2.1 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.

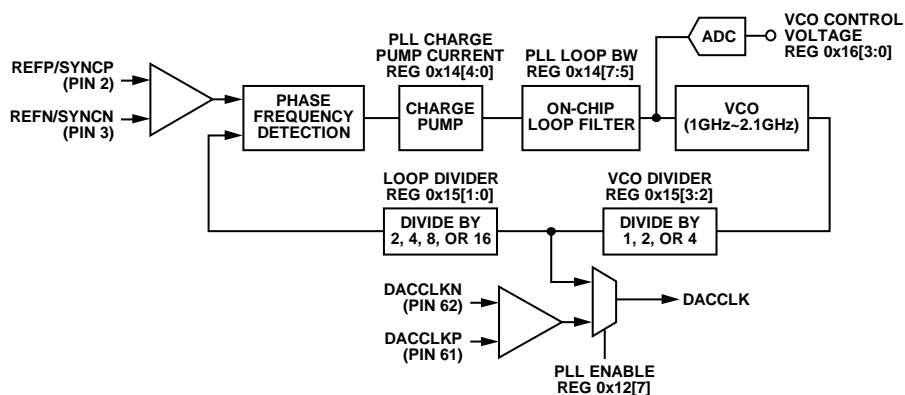


Figure 52. PLL Clock Multiplication Circuit

PLL SETTINGS

The PLL circuitry requires three settings to be programmed to their nominal values. The PLL values shown in Table 21 are the recommended settings for these parameters.

Table 21. 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 Control Enable | 0x15[4] | 0 |

CONFIGURING THE VCO TUNING BAND

The PLL VCO has a valid operating range from approximately 1.0 GHz to 2.1 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 53. 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 $+85^{\circ}\text{C}$ 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.

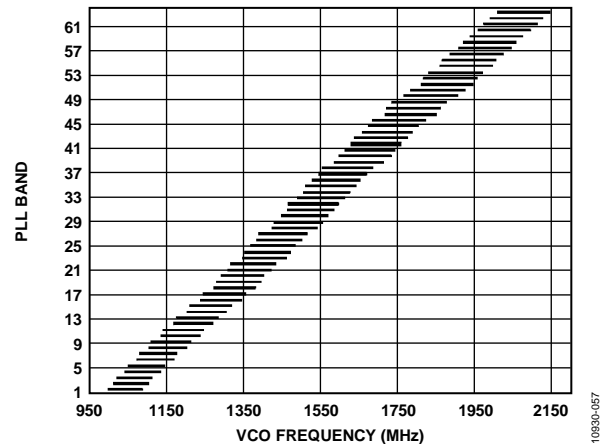


Figure 53. PLL Lock Range for a Typical Device

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

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

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 111b to PLL loop bandwidth for the best performance.
3. Select the desired band.
4. Set the PLL mode to manual using Register 0x12[6] = 1b.
5. Enable the PLL using Register 0x12[7] = 1b.
6. Enable the PLL one more time using Register 0x12[7] = 1b.

ANALOG OUTPUTS

TRANSMIT DAC OPERATION

Figure 54 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.

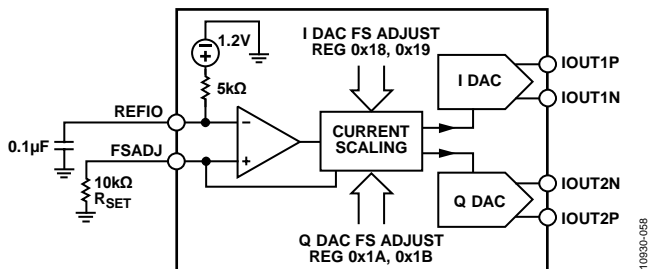


Figure 54. Simplified Block Diagram of DAC Core

The DAC has a 1.2 V band gap reference with an output impedance of 5 kΩ. The reference output voltage appears on the REFIO pin. When using the internal reference, decouple the REFIO 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 REFIO pin. If desired, the internal reference can be overdriven by applying an external reference (from 1.10 V to 1.30 V) to the REFIO 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 set individually for the Q and IDACs in Register 0x40 and Register 0x44, respectively, is as follows:

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

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 20.16 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 55.

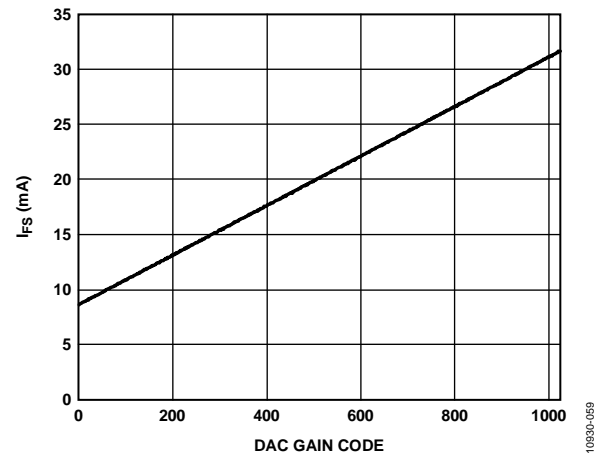


Figure 55. DAC Full-Scale Current vs. DAC Gain Code

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 AD9142 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 feed-through, and noise.

Figure 56 shows the most basic DAC output circuitry. A pair of resistors, R_O , convert 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 57 for the output voltage waveforms.

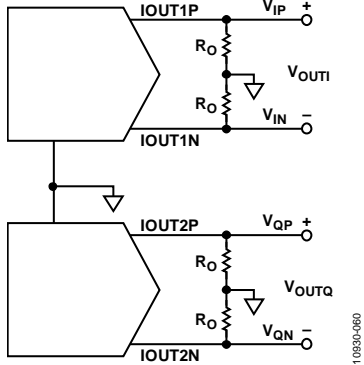


Figure 56. Basic Transmit DAC Output Circuit

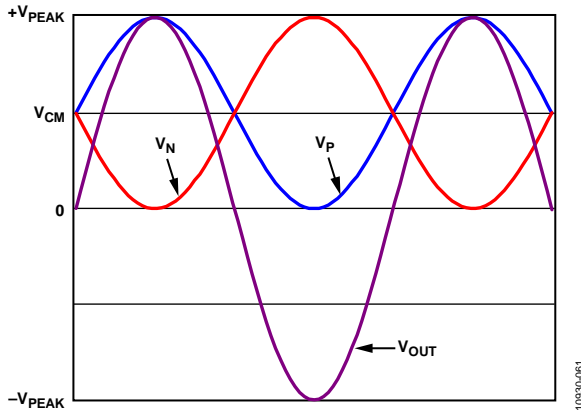


Figure 57. Output Voltage Waveforms

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 AD9142 interfaces to the ADL537x family of modulators with a minimal number of components. An example of the recommended interface circuitry is shown in Figure 58.

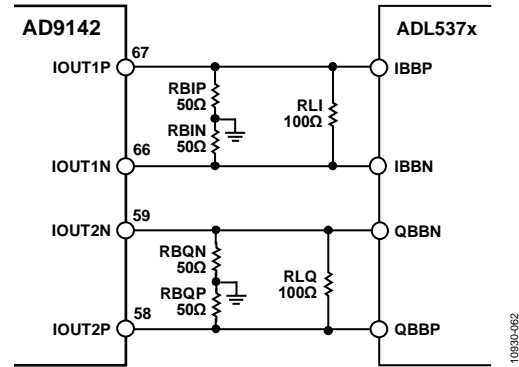


Figure 58. Typical Interface Circuitry Between the AD9142 and the ADL537x Family of Modulators

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 59 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 AD9142 DAC to an IQ modulator, refer to the [Circuits from the Lab CN-0205, Interfacing the ADL5375 I/Q Modulator to the AD9122 Dual Channel, 1.2 GSPS High Speed DAC](#) on the Analog Devices website.

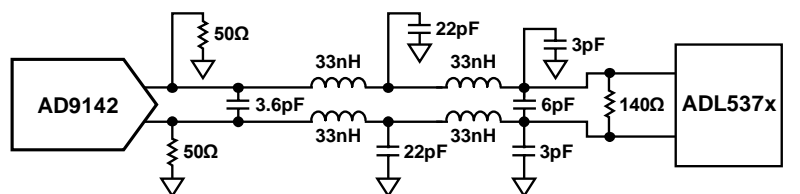


Figure 59. DAC Modulator Interface with Fifth-Order, Low-Pass Filter

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 [Application Note AN-1039, Correcting Imperfections in IQ Modulators to Improve RF Signal Fidelity](#) and [Application Note AN-1100, Wireless Transmitter IQ Balance and Sideband Suppression](#) from the Analog Devices website.

EXAMPLE START-UP ROUTINE

To ensure reliable start-up of the AD9142, certain sequences must be followed. This section shows an example start-up routine.

Device Configuration and Start-Up Sequence

- $f_{\text{DATA}} = 200$ MHz, interpolation is 8 \times .
- Input data is baseband data.
- $f_{\text{OUT}} = 350$ MHz.
- PLL is enabled, $f_{\text{REF}} = 200$ MHz.
- Fine NCO is enabled, inverse sinc filter is enabled.
- A delay line-based mode is used with an interface delay setting of 0.

Derived PLL Settings

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

- $f_{\text{DAC}} = 200 \times 8 = 1600$ MHz.
- $f_{\text{VCO}} = f_{\text{DAC}} = 1600$ MHz ($1 \text{ GHz} < f_{\text{VCO}} < 2 \text{ GHz}$).
- VCO divider = $f_{\text{VCO}}/f_{\text{DAC}} = 1$.
- Loop divider = $f_{\text{DAC}}/f_{\text{REF}} = 8$.

Derived NCO Settings

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

- $f_{\text{DAC}} = 200 \times 8 = 1600$ MHz.
- $f_{\text{CARRIER}} = f_{\text{OUT}} = 350$ MHz.
- $\text{FTW} = f_{\text{CARRIER}}/f_{\text{DAC}} \times 2^{32} = 0x38000000$.

Start-Up Sequence

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 H/W reset (optional).

/* Device configuration register write sequence. Must be written in sequence for every device after reset*/

```
0x00 → 0x20 /* Issue software reset */
0x20 → 0x01 /* Device Startup Configuration */
0x79 → 0x18 /* Device Startup Configuration */
0x80 → 0xAD /* Device Startup Configuration */
0xE1 → 0x1A /* Device Startup Configuration */
```

```
/* Configure PLL */
0x14 → 0xE3 /* Configure PLL loop BW and charge
pump current */
0x15 → 0xC2 /* Configure VCO divider and Loop
divider */
0x12 → 0xC0 /*Enable the PLL */
0x12 → 0x80
```

```
/* Configure Data Interface */
0x5E → 0x00 /* Delay setting 0 */
0x5F → 0x08 /* Enable the delay line */
```

```
/* Configure Interpolation filter */
0x28 → 0x03 /* 8 $\times$  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 */
```

```
/* Configure NCO */
0x27 → 0x40 /* Enable NCO */
0x31 → 0x00
0x32 → 0x00
0x33 → 0x00
0x34 → 0x38
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 REGISTER MAP AND DESCRIPTION

Table 22. 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] | 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] | Reserved | | | | | FIFO_UNDERFLOW | FIFO_OVERFLOW | FIFO_WARNING | 0x00 | R |
| 0x07 | IRQ_SELO | [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] | Reserved | | | | | SEL_FIFO_UNDERFLOW | SEL_FIFO_OVERFLOW | SEL_FIFO_WARNING | 0x00 | RW |
| 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 | | | | | 0xBF | 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 | | | | FRAME_RESET_ACK | EN_CON_FRAME_RESET | FRAME_RESET_MODE | | 0x12 | RW |
| 0x23 | FIFO_LEVEL_CONFIG | [7:0] | Reserved | INTEGRAL_FIFO_LEVEL_REQUEST | | | Reserved | FRACTIONAL_FIFO_LEVEL_REQUEST | | | 0x40 | RW |
| 0x24 | FIFO_LEVEL_READBACK | [7:0] | Reserved | INTEGRAL_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_SEL | [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_SIDE_BAND_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 | | | 0x000 | 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 | [7:0] | Reserved | | | RAMP_UP_STEP | | | 0x01 | RW | | |
| 0x42 | GAIN_STEP_CTRL1 | [7:0] | DAC_OUTPUT_STATUS | DAC_OUTPUT_ON | RAMP_DOWN_STEP | | | | 0x01 | 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 | OVER_THRESHOLD_SHUTDOWN_EN | Reserved | FIFO_ERROR_SHUTDOWN_EN | 0x8F | RW | |
| 0x5E | DATA_RX_CTRL0 | [7:0] | DLY_TAP_LSB | | | | | | 0xFF | RW | | |
| 0x5F | DATA_RX_CTRL1 | [7:0] | Reserved | | | DLYLINE_EN | DLY_TAP_MSB | | | 0x07 | RW | |
| 0x79 | DEVICE_CONFIG0 | [7:0] | DEVICE_CONFIGURATION0 | | | | | | 0x00 | RW | | |
| 0x7F | Version | [7:0] | Version | | | | | | 0x05 | R | | |
| 0x80 | DEVICE_CONFIG1 | [7:0] | DEVICE_CONFIGURATION1 | | | | | | 0x00 | RW | | |
| 0xE1 | DEVICE_CONFIG2 | [7:0] | DEVICE_CONFIGURATION2 | | | | | | 0x00 | RW | | |

SPI CONFIGURE REGISTER

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

Table 23. Bit Descriptions for Common

| Bits | Bit Name | Settings | Description | Reset | Access |
|------|---------------|----------|---|-------|--------|
| 6 | SPI_LSB_FIRST | 0 1 | Serial port communication, MSB-first or LSB-first selection. MSB first. LSB first. | 0x0 | RW |
| 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. | 0x0 | RW |

POWER-DOWN CONTROL REGISTER

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

Table 24. 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. | 0x1 | 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 data path is not affected. | 0x1 | 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. | 0x0 | RW |
| 2 | PD_DEVICE | | The bandgap circuitry is powered down when set to 1. This bit powers down the entire chip. | 0x0 | RW |
| 1 | PD_DACCLK | | The DAC clocking 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. | 0x0 | 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 frame is not used. | 0x0 | RW |

INTERRUPT ENABLE0 REGISTER

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

Table 25. Bit Descriptions for INTERRUPT_ENABLE0

| Bits | Bit Name | Settings | Description | Reset | Access |
|------|-----------------------|----------|-------------------------------------|-------|--------|
| 6 | ENABLE_SYNC_LOST | | Enable interrupt for sync lost. | 0x0 | RW |
| 5 | ENABLE_SYNC_LOCKED | | Enable interrupt for sync lock. | 0x0 | RW |
| 4 | ENABLE_SYNC_DONE | | Enable interrupt for sync done. | 0x0 | RW |
| 3 | ENABLE_PLL_LOST | | Enable interrupt for PLL lost. | 0x0 | RW |
| 2 | ENABLE_PLL_LOCKED | | Enable interrupt for PLL locked. | 0x0 | RW |
| 1 | ENABLE_OVER_THRESHOLD | | Enable interrupt for overthreshold. | 0x0 | RW |
| 0 | ENABLE_DACOUT_MUTED | | Enable interrupt for DACOUT muted. | 0x0 | RW |

INTERRUPT ENABLE1 REGISTER

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

Table 26. Bit Descriptions for INTERRUPT_ENABLE1

| Bits | Bit Name | Settings | Description | Reset | Access |
|------|-----------------------|----------|--------------------------------------|-------|--------|
| 2 | ENABLE_FIFO_UNDERFLOW | | Enable interrupt for FIFO underflow. | 0x0 | RW |
| 1 | ENABLE_FIFO_OVERFLOW | | Enable interrupt for FIFO overflow. | 0x0 | RW |
| 0 | ENABLE_FIFO_WARNING | | Enable interrupt for FIFO warning. | 0x0 | RW |

INTERRUPT FLAG0 REGISTER

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

Table 27. 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. | 0x0 | R |
| 5 | SYNC_LOCKED | | SYNC_LOCKED is set to 1 when sync is locked. | 0x0 | R |
| 4 | SYNC_DONE | | SYNC_DONE is set to 1 when sync is done. | 0x0 | R |
| 3 | PLL_LOST | | PLL_LOST is set to 1 when PLL loses lock. | 0x0 | R |
| 2 | PLL_LOCKED | | PLL_LOCKED is set to 1 when PLL is locked. | 0x0 | R |
| 1 | OVER_THRESHOLD | | OVER_THRESHOLD is set to 1 when input power is overthreshold. | 0x0 | R |
| 0 | DACOUT_MUTED | | DACOUT_MUTED is set to 1 when the DAC output is muted (midscale dc). | 0x0 | R |

INTERRUPT FLAG1 REGISTER

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

Table 28. Bit Descriptions for INTERRUPT_FLAG1

| Bits | Bit Name | Settings | Description | Reset | Access |
|------|----------------|----------|--|-------|--------|
| 2 | FIFO_UNDERFLOW | | FIFO_UNDERFLOW is set to 1 when the FIFO read pointer catches the FIFO write pointer. | 0x0 | R |
| 1 | FIFO_OVERFLOW | | FIFO_OVERFLOW is set to 1 when the FIFO write pointer catches the FIFO read pointer. | 0x0 | R |
| 0 | FIFO_WARNING | | FIFO_WARNING is set to 1 when the FIFO is one slot from empty (≤ 1) or full (≥ 6). | 0x0 | R |

INTERRUPT SELECT0 REGISTER

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

Table 29. Bit Descriptions for IRQ_SEL0

| Bits | Bit Name | Settings | Description | Reset | Access |
|------|--------------------|----------|---|-------|--------|
| 6 | SEL_SYNC_LOST | 0 | Selects the $\overline{\text{IRQ1}}$ pin. | 0x0 | RW |
| | | 1 | Selects the $\overline{\text{IRQ2}}$ pin. | | |
| 5 | SEL_SYNC_LOCKED | 0 | Selects the $\overline{\text{IRQ1}}$ pin. | 0x0 | RW |
| | | 1 | Selects the $\overline{\text{IRQ2}}$ pin. | | |
| 4 | SEL_SYNC_DONE | 0 | Selects the $\overline{\text{IRQ1}}$ pin. | 0x0 | RW |
| | | 1 | Selects the $\overline{\text{IRQ2}}$ pin. | | |
| 3 | SEL_PLL_LOST | 0 | Selects the $\overline{\text{IRQ1}}$ pin. | 0x0 | RW |
| | | 1 | Selects the $\overline{\text{IRQ2}}$ pin. | | |
| 2 | SEL_PLL_LOCKED | 0 | Selects the $\overline{\text{IRQ1}}$ pin. | 0x0 | RW |
| | | 1 | Selects the $\overline{\text{IRQ2}}$ pin. | | |
| 1 | SEL_OVER_THRESHOLD | 0 | Selects the $\overline{\text{IRQ1}}$ pin. | 0x0 | RW |
| | | 1 | Selects the $\overline{\text{IRQ2}}$ pin. | | |
| 0 | SEL_DACOUT_MUTED | 0 | Selects the $\overline{\text{IRQ1}}$ pin. | 0x0 | RW |
| | | 1 | Selects the $\overline{\text{IRQ2}}$ pin. | | |

INTERRUPT SELECT1 REGISTER

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

Table 30. Bit Descriptions for IRQ_SEL1

| Bits | Bit Name | Settings | Description | Reset | Access |
|------|--------------------|----------|---|-------|--------|
| 2 | SEL_FIFO_UNDERFLOW | 0 | Selects the $\overline{\text{IRQ1}}$ pin. | 0x0 | RW |
| | | 1 | Selects the $\overline{\text{IRQ2}}$ pin. | | |
| 1 | SEL_FIFO_OVERFLOW | 0 | Selects the $\overline{\text{IRQ1}}$ pin. | 0x0 | RW |
| | | 1 | Selects the $\overline{\text{IRQ2}}$ pin. | | |
| 0 | SEL_FIFO_WARNING | 0 | Selects the $\overline{\text{IRQ1}}$ pin. | 0x0 | RW |
| | | 1 | Selects the $\overline{\text{IRQ2}}$ pin. | | |

DAC CLOCK RECEIVER CONTROL REGISTER

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

Table 31. 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. | 0x1 | RW |
| 5 | DACCLK_CROSSPOINT_CTRL_ENABLE | | Enables crosspoint control at the DACCLK input. For best performance, the default and recommended status is turned on. | 0x1 | RW |
| [4:0] | DACCLK_CROSSPOINT_LEVEL | 01111 11111 | A twos complement value. For best performance, it is recommended to set DACCLK_CROSSPOINT_LEVEL to the default value. Highest crosspoint. Lowest crosspoint. | 0x1F | RW |

REF CLOCK RECEIVER CONTROL REGISTER

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

Table 32. Bit Descriptions for REFCLK_RECEIVER_CTRL

| Bits | Bit Name | Settings | Description | Reset | Access |
|-------|-------------------------------|----------------|--|-------|--------|
| 7 | DUTYCYCLE_CORRECTION | | Enables duty cycle correction at the REFCLK input. For best performance, the default and recommended status is turned off. | 0x0 | RW |
| 5 | REFCLK_CROSSPOINT_CTRL_ENABLE | | Enables crosspoint control at the REFCLK input. For best performance, the default and recommended status is turned off. | 0x0 | RW |
| [4:0] | REFCLK_CROSSPOINT_LEVEL | 01111 11111 | A twos complement value. For best performance, it is recommended to set REFCLK_CROSSPOINT_LEVEL to the default value. Highest crosspoint. Lowest crosspoint. | 0x1F | RW |

PLL CONTROL REGISTER

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

Table 33. Bit Descriptions for PLL_CTRL0

| Bits | Bit Name | Settings | Description | Reset | Access |
|-------|-----------------|------------------|---|-------|--------|
| 7 | PLL_ENABLE | | Enables PLL clock multiplier. | 0x0 | RW |
| 6 | AUTO_MANUAL_SEL | 0 1 | PLL band selection mode. Automatic mode. Manual mode. | 0x0 | RW |
| [5:0] | PLL_MANUAL_BAND | 000000 111111 | PLL band setting in manual mode. 64 bands in total, covering a 1 GHz to 2.1 GHz VCO range. Lowest band (1 GHz). Highest band (2.1 GHz). | 0x00 | RW |

PLL CONTROL REGISTER

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

Table 34. Bit Descriptions for PLL_CTRL2

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

PLL CONTROL REGISTER

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

Table 35. Bit Descriptions for PLL_CTRL3

| Bits | Bit Name | Settings | Description | Reset | Access |
|-------|--------------------|----------------------|---|-------|--------|
| [7:6] | DIGLOGIC_DIVIDER | 00 01 10 11 | 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 below 75 MHz. $f_{REFCLK}/f_{DIG} = 2.$ $f_{REFCLK}/f_{DIG} = 4.$ $f_{REFCLK}/f_{DIG} = 8.$ $f_{REFCLK}/f_{DIG} = 16.$ | 0x3 | RW |
| 4 | CROSSPOINT_CTRL_EN | | Enable loop divider crosspoint control. The default and recommended setting is turned off (0) for optimal PLL performance. | 0x0 | RW |
| [3:2] | VCO_DIVIDER | 00 01 10 11 | PLL VCO divider. This divider determines the ratio of the VCO frequency to the DACCLK frequency. $f_{VCO}/f_{DACCLK} = 1.$ $f_{VCO}/f_{DACCLK} = 2.$ $f_{VCO}/f_{DACCLK} = 4.$ $f_{VCO}/f_{DACCLK} = 4.$ | 0x2 | RW |
| [1:0] | LOOP_DIVIDER | 00 01 10 11 | PLL loop divider. This divider determines the ratio of the DACCLK frequency to the REFCLK frequency. $f_{DACCLK}/f_{REFCLK} = 2.$ $f_{DACCLK}/f_{REFCLK} = 4.$ $f_{DACCLK}/f_{REFCLK} = 8.$ $f_{DACCLK}/f_{REFCLK} = 16.$ | 0x1 | RW |

PLL STATUS REGISTER

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

Table 36. Bit Descriptions for PLL_STATUS0

| Bits | Bit Name | Settings | Description | Reset | Access |
|-------|---------------------------|----------------------|---|-------|--------|
| 7 | PLL_LOCK | | PLL clock multiplier output is stable. | 0x0 | R |
| [3:0] | VCO_CTRL_VOLTAGE_READBACK | 1111 0111 0000 | VCO control voltage readback. A binary value. The highest VCO control voltage. The mid value 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. The lowest VCO control voltage. | 0x0 | R |

PLL STATUS REGISTER

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

Table 37. 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 38. Bit Descriptions for IDAC_FS_ADJ0

| Bits | Bit Name | Settings | Description | Reset | Access |
|-------|---------------------------|----------|--------------------|-------|--------|
| [7:0] | IDAC_FULLSCALE_ADJUST_LSB | | See Register 0x19. | 0xF9 | RW |

IDAC FS ADJUST MSB REGISTER

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

Table 39. Bit Descriptions for IDAC_FS_ADJ1

| Bits | Bit Name | Settings | Description | Reset | Access |
|-------|---------------------------|----------|---|-------|--------|
| [1:0] | IDAC_FULLSCALE_ADJUST_MSB | | IDAC full-scale adjust, Bits[9:0] sets 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 40. Bit Descriptions for QDAC_FS_ADJ0

| Bits | Bit Name | Settings | Description | Reset | Access |
|-------|---------------------------|----------|--------------------|-------|--------|
| [7:0] | QDAC_FULLSCALE_ADJUST_LSB | | See Register 0x1B. | 0xF9 | RW |

QDAC FS ADJUST MSB REGISTER

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

Table 41. Bit Descriptions for QDAC_FS_ADJ1

| Bits | Bit Name | Settings | Description | Reset | Access |
|-------|---------------------------|----------|---|-------|--------|
| [1:0] | QDAC_FULLSCALE_ADJUST_MSB | | QDAC full-scale adjust, Bits[9:0] sets 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 42. Bit Descriptions for DIE_TEMP_SENSOR_CTRL

| Bits | Bit Name | Settings | Description | Reset | Access |
|-------|--------------------|---------------------------------|--|-------|--------|
| [6:4] | FS_CURRENT | 000 001 ... 110 111 | Temperature sensor ADC full-scale current. Using the default setting is recommended. 50 μ A. 62.5 μ A. 125 μ A. 137.5 μ A. | 0x0 | RW |
| [3:1] | REF_CURRENT | 000 001 ... 110 111 | Temperature sensor ADC reference current. Using the default setting is recommended. 12.5 μ A. 19 μ A. 50 μ A. 56.5 μ A. | 0x1 | RW |
| 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 43. Bit Descriptions for DIE_TEMP_LSB

| Bits | Bit Name | Settings | Description | Reset | Access |
|-------|--------------|----------|--------------------|-------|--------|
| [7:0] | DIE_TEMP_LSB | | See Register 0x1E. | 0x00 | R |

DIE TEMPERATURE MSB REGISTER

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

Table 44. Bit Descriptions for DIE_TEMP_MSB

| Bits | Bit Name | Settings | Description | Reset | Access |
|-------|--------------|----------|---|-------|--------|
| [7:0] | DIE_TEMP_MSB | | Die temperature, Bits[15:0] 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 45. Bit Descriptions for CHIP_ID

| Bits | Bit Name | Settings | Description | Reset | Access |
|-------|----------|----------|---|-------|--------|
| [7:0] | CHIP_ID | | The AD9142 chip ID is 0x0A. | 0x0A | R |

INTERRUPT CONFIGURATION REGISTER

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

Table 46. 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 Interrupt Request Operation section). | | |

SYNC CTRL REGISTER

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

Table 47. Bit Descriptions for SYNC_CTRL

| Bits | Bit Name | Settings | Description | Reset | Access |
|------|-------------------|----------|---|-------|--------|
| 1 | SYNC_CLK_EDGE_SEL | 0 1 | Selects the sampling edge of the DACCLK on the SYNC CLK. SYNC CLK is sampled by rising edges of DACCLK. SYNC CLK is sampled by falling edges of DACCLK. | 0x0 | RW |
| 0 | SYNC_ENABLE | | Enables multichip synchronization. | 0x0 | RW |

FRAME RESET CTRL REGISTER

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

Table 48. Bit Descriptions for FRAME_RST_CTRL

| Bits | Bit Name | Settings | Description | Reset | Access |
|-------|--------------------|----------------------|--|-------|--------|
| 3 | FRAME_RESET_ACK | | Frame reset acknowledge. This bit is set to 1 when a valid frame pulse is received. | 0x0 | R |
| 2 | EN_CON_FRAME_RESET | 0 1 | Reset mode selection. Responds to only the first valid frame pulse and resets the FIFO and/or NCO one time only. This is the default and recommended mode. Responds to every valid frame pulse and resets the FIFO and/or NCO accordingly. | 0x0 | RW |
| [1:0] | FRAME_RESET_MODE | 00 01 10 11 | These bits determine what is to be reset when the device receives a valid frame signal. FIFO only. NCO only. FIFO and NCO. None. | 0x2 | RW |

FIFO LEVEL CONFIGURATION REGISTER

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

Table 49. Bit Descriptions for FIFO_LEVEL_CONFIG

| Bits | Bit Name | Settings | Description | Reset | Access |
|-------|-------------------------------|---|--|-------|--------|
| [6:4] | INTEGRAL_FIFO_LEVEL_REQUEST | 000 001 ... 111 | Sets the integral 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 integral level = 4 and fractional level = 0. See the FIFO Operation section for details. 0. 1. 7. | 0x4 | RW |
| [2:0] | FRACTIONAL_FIFO_LEVEL_REQUEST | 000 001 ... Max allowed setting. | Sets 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. 0. 1. 001 in 2x. 003 in 4x. 007 in 8x. | 0x0 | RW |

FIFO LEVEL READBACK REGISTER

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

Table 50. Bit Descriptions for FIFO_LEVEL_READBACK

| Bits | Bit Name | Settings | Description | Reset | Access |
|-------|--------------------------------|----------|---|-------|--------|
| [6:4] | INTEGRAL_FIFO_LEVEL_READBACK | | The integral 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 Bit[6:4]. | 0x0 | R |

FIFO CTRL REGISTER

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

Table 51. 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 52. Bit Descriptions for DATA_FORMAT_SEL

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

DATAPATH CONTROL REGISTER

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

Table 53. 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_SIDEHAND_SEL | 0 1 | Selects the single-side NCO modulation image. The NCO outputs the high-side image. The NCO outputs the low-side image. | 0x0 | RW |
| 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 54. Bit Descriptions for INTERPOLATION_CTRL

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

OVER THRESHOLD CTRL0 REGISTER

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

Table 55. Bit Descriptions for OVER_THRESHOLD_CTRL0

| Bits | Bit Name | Settings | Description | Reset | Access |
|-------|-----------------------------|----------|--------------------|-------|--------|
| [7:0] | THRESHOLD_LEVEL_REQUEST_LSB | | See Register 0x2A. | 0x0 | RW |

OVER THRESHOLD CTRL1 REGISTER

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

Table 56. Bit Descriptions for OVER_THRESHOLD_CTRL1

| Bits | Bit Name | Settings | Description | Reset | Access |
|-------|-----------------------------|----------|---|-------|--------|
| [4:0] | THRESHOLD_LEVEL_REQUEST_MSB | | Minimum average input power ($I^2 + Q^2$) to trigger the input power protection function. | 0x00 | RW |

OVER THRESHOLD CTRL2 REGISTER

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

Table 57. 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 | 0000 0001 ... 1010 1011 to 1111 | Number of data input samples for power averaging. 512 IQ data sample pairs. 1024 IQ data sample pairs. ... 2 ¹⁹ IQ data sample pairs. invalid. | 0x0 | RW |

INPUT POWER READBACK LSB REGISTER

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

Table 58. Bit Descriptions for INPUT_POWER_READBACK_LSB

| Bits | Bit Name | Settings | Description | Reset | Access |
|-------|--------------------------|----------|--------------------|-------|--------|
| [7:0] | INPUT_POWER_READBACK_LSB | | See Register 0x2D. | 0x0 | R |

INPUT POWER READBACK MSB REGISTER

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

Table 59. Bit Descriptions for INPUT_POWER_READBACK_MSB

| Bits | Bit Name | Settings | Description | Reset | Access |
|-------|--------------------------|----------|--------------------------------------|-------|--------|
| [4:0] | INPUT_POWER_READBACK_MSB | | Input signal average power readback. | 0x00 | R |

NCO CONTROL REGISTER

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

Table 60. 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_FREQ_TUNING_WORD0 REGISTER

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

Table 61. Bit Descriptions for NCO_FREQ_TUNING_WORD0

| Bits | Bit Name | Settings | Description | Reset | Access |
|-------|----------|----------|--------------------|-------|--------|
| [7:0] | NCO_FTW0 | | See Register 0x34. | 0x00 | RW |

NCO_FREQ_TUNING_WORD1 REGISTER

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

Table 62. Bit Descriptions for NCO_FREQ_TUNING_WORD1

| Bits | Bit Name | Settings | Description | Reset | Access |
|-------|----------|----------|--------------------|-------|--------|
| [7:0] | NCO_FTW1 | | See Register 0x34. | 0x00 | RW |

NCO_FREQ_TUNING_WORD2 REGISTER

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

Table 63. Bit Descriptions for NCO_FREQ_TUNING_WORD2

| Bits | Bit Name | Settings | Description | Reset | Access |
|-------|----------|----------|--------------------|-------|--------|
| [7:0] | NCO_FTW2 | | See Register 0x34. | 0x00 | RW |

NCO_FREQ_TUNING_WORD3 REGISTER

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

Table 64. Bit Descriptions for NCO_FREQ_TUNING_WORD3

| Bits | Bit Name | Settings | Description | Reset | Access |
|-------|----------|----------|---|-------|--------|
| [7:0] | NCO_FTW3 | | FTW[31:0] is 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_OFFSET0 REGISTER

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

Table 65. Bit Descriptions for NCO_PHASE_OFFSET0

| Bits | Bit Name | Settings | Description | Reset | Access |
|-------|----------------------|----------|--------------------|-------|--------|
| [7:0] | NCO_PHASE_OFFSET_LSB | | See Register 0x36. | 0x00 | RW |

NCO_PHASE_OFFSET1 REGISTER

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

Table 66. Bit Descriptions for NCO_PHASE_OFFSET1

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

IQ_PHASE_ADJ0 REGISTER

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

Table 67. Bit Descriptions for IQ_PHASE_ADJ0

| Bits | Bit Name | Settings | Description | Reset | Access |
|-------|------------------|----------|--------------------|-------|--------|
| [7:0] | IQ_PHASE_ADJ_LSB | | See Register 0x38. | 0x00 | RW |

IQ_PHASE_ADJ1 REGISTER

Address: 0x38, Reset: 0x000, Name: IQ_PHASE_ADJ1

Table 68. Bit Descriptions for IQ_PHASE_ADJ1

| Bits | Bit Name | Settings | Description | Reset | Access |
|-------|------------------|----------|---|-------|--------|
| [4:0] | IQ_PHASE_ADJ_MSB | | IQ phase adjust, Bits[12:0], is used to insert a phase offset between the I and Q datapaths. It provides an adjustment range of ± 14 degrees with a step of 0.0035 degrees. This value is in twos complement. See the Quadrature Phase Adjustment section for more information. | 0x0 | RW |

IDAC_DC_OFFSET0 REGISTER

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

Table 69. Bit Descriptions for IDAC_DC_OFFSET0

| Bits | Bit Name | Settings | Description | Reset | Access |
|-------|--------------------|----------|--------------------|-------|--------|
| [7:0] | IDAC_DC_OFFSET_LSB | | See Register 0x3C. | 0x00 | RW |

IDAC_DC_OFFSET1 REGISTER

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

Table 70. Bit Descriptions for IDAC_DC_OFFSET1

| Bits | Bit Name | Settings | Description | Reset | Access |
|-------|--------------------|----------|--|-------|--------|
| [7:0] | IDAC_DC_OFFSET_MSB | | IDAC DC offset, Bits[15:0], is a dc value that is added directly to the sample values written to the IDAC. | 0x00 | RW |

QDAC_DC_OFFSET0 REGISTER

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

Table 71. Bit Descriptions for QDAC_DC_OFFSET0

| Bits | Bit Name | Settings | Description | Reset | Access |
|-------|--------------------|----------|--------------------|-------|--------|
| [7:0] | QDAC_DC_OFFSET_LSB | | See Register 0x3E. | 0x00 | RW |

QDAC_DC_OFFSET1 REGISTER

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

Table 72. Bit Descriptions for QDAC_DC_OFFSET1

| Bits | Bit Name | Settings | Description | Reset | Access |
|-------|--------------------|----------|--|-------|--------|
| [7:0] | QDAC_DC_OFFSET_MSB | | QDAC DC offset, Bits[15:0], is a dc value that is added directly to the sample values written to the QDAC. | 0x00 | RW |

IDAC_GAIN_ADJ REGISTER

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

Table 73. 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_ADJ REGISTER

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

Table 74. 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 $-\infty$ to 6 dB. The default gain setting is 0x20, which maps to unity gain (0 dB). | 0x20 | RW |

GAIN_STEP_CTRL0 REGISTER

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

Table 75. 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 I/QDAC_GAIN_ADJ (Register 0x3F and Register 0x40). The bit weighting is MSB = 2^1 , LSB = 2^{-4} . Note that the value in this register should not be greater than the values in the I/QDAC_GAIN_ADJ (Register 0x3F and Register 0x40). | 0x01 | RW |

GAIN_STEP_CTRL1 REGISTER

Address: 0x42, Reset: 0x01, Name: GAIN_STEP_CTRL1

Table 76. Bit Descriptions for GAIN_STEP_CTRL1

| Bits | Bit Name | Settings | Description | Reset | Access |
|-------|-------------------|----------|---|-------|--------|
| 7 | DAC_OUTPUT_STATUS | | This bit indicates the DAC output on/off status. When the DAC output is automatically turned off, this bit is 1. | 0x0 | RW |
| 6 | DAC_OUTPUT_ON | | In the case where the DAC output is automatically turned off in the input power protection mode or TX enable mode, this register allows for turning on the DAC output manually. It is a self clear bit. | 0x0 | 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 should not be greater than the values in the I/QDAC_GAIN_ADJ (Register 0x3F and Register 0x40). | 0x01 | RW |

TX ENABLE CONTROL REGISTER

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

Table 77. 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. | 0x1 | 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. | 0x1 | RW |
| 0 | TXENABLE_POWER_DOWN_EN | | When set to 1, the device is put in power down mode when TXENABLE signal from the TXEN pin is low. | 0x1 | RW |

DAC OUTPUT CONTROL REGISTER

Address: 0x44, Reset: 0x8F, Name: DAC_OUTPUT_CTRL

Table 78. Bit Descriptions for DAC_OUTPUT_CTRL

| Bits | Bit Name | Settings | Description | Reset | Access |
|------|---------------------------|----------|--|-------|--------|
| 7 | DAC_OUTPUT_CTRL_EN | | Enable the DAC output control. This bit needs to be set to 1 to enable the rest of the 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 |

DATA RECEIVER TEST CONTROL REGISTER

Address: 0x5E, Reset: 0xFF, Name: DATA_RX_CTRL0

Table 79. Bit Descriptions for DATA_RX_CTRL0

| Bits | Bit Name | Settings | Description | Reset | Access |
|-------|-------------|----------|-------------------------|-------|--------|
| [7:0] | DLY_TAP_LSB | | See Register 0x5F[2:0]. | 0xFF | RW |

DATA RECEIVER TEST CONTROL REGISTER

Address: 0x5F, Reset: 0x07, Name: DATA_RX_CTRL1

Table 80. Bit Descriptions for DATA_RX_CTRL1

| Bits | Bit Name | Settings | Description | Reset | Access |
|-------|-------------|----------|---|-------|--------|
| 3 | DLYLINE_EN | | 1 = Enable the data interface. | 0x0 | RW |
| [2:0] | DLY_TAP_MSB | | Four available delay settings. See the Interface Delay Line section for more information. | 0x7 | RW |
| | | 00 | 0x000 | | |
| | | 01 | 0x007 | | |
| | | 10 | 0x07F | | |
| | | 11 | 0x5FF | | |

DEVICE CONFIGURATION0 REGISTER

Address: 0x79, Reset: 0x00, Name: DEVICE_CONFIG0

Table 81. Bit Descriptions for DEVICE_CONFIG0

| Bits | Bit Name | Settings | Description | Reset | Access |
|-------|-----------------------|----------|---|-------|--------|
| [7:0] | DEVICE_CONFIGURATION0 | 0x18 | Recommended setting for device start-up configuration | 0x00 | RW |

VERSION REGISTER

Address: 0x7E, Reset: 0x05, Name: Version

Table 82. Bit Descriptions for Version

| Bits | Bit Name | Settings | Description | Reset | Access |
|-------|----------|----------|--------------|-------|--------|
| [7:0] | Version | | Chip version | 0x05 | R |

DEVICE CONFIGURATION1 REGISTER

Address: 0x80, Reset: 0x00, Name: DEVICE_CONFIG1

Table 83. Bit Descriptions for DEVICE_CONFIG1

| Bits | Bit Name | Settings | Description | Reset | Access |
|-------|-----------------------|----------|---|-------|--------|
| [7:0] | DEVICE_CONFIGURATION1 | 0xAD | Recommended setting for device start-up configuration | 0x00 | RW |

DEVICE CONFIGURATION2 REGISTER

Address: 0xE1, Reset: 0x00, Name: DEVICE_CONFIG2

Table 84. Bit Descriptions for DEVICE_CONFIG2

| Bits | Bit Name | Settings | Description | Reset | Access |
|-------|-----------------------|----------|---|-------|--------|
| [7:0] | DEVICE_CONFIGURATION2 | 0x1A | Recommended setting for device start-up configuration | 0x00 | RW |

DAC LATENCY AND SYSTEM SKEWS

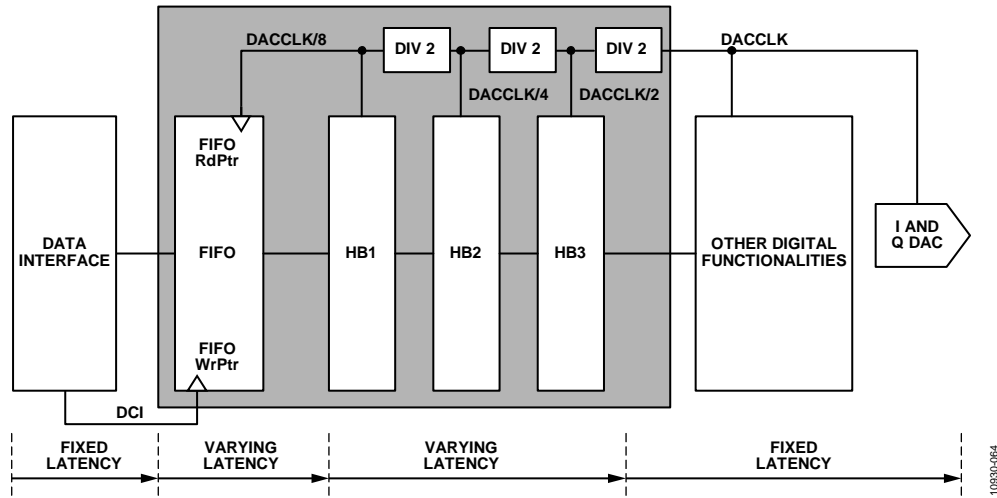


Figure 60. Breakdown of Pipeline Latencies

DAC LATENCY VARIATIONS

DACs, like any other devices with internal multiphase clocks, have an inherent pipeline latency variation. Figure 60 shows the breakdown of pipeline latencies in the AD9142. 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 61 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 predefined 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 side.

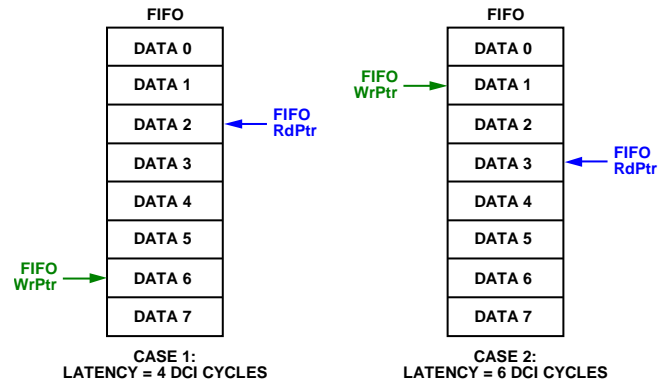


Figure 61. Example of FIFO Latency Difference

Figure 62 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.

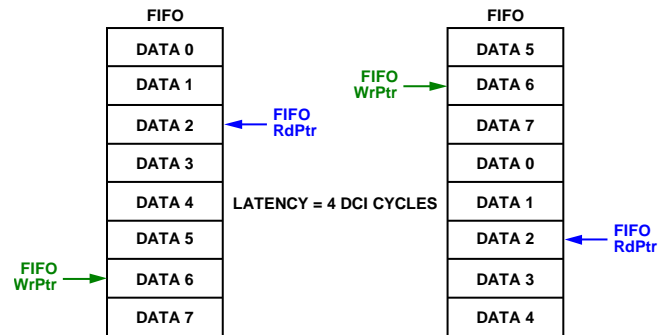


Figure 62. Example of Equal FIFO Latencies

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 need to be internally generated by dividing down the DACCLK), there is an inherent latency variation in the DAC. Figure 63 is an example of this latency variation in 2x 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 62). 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.

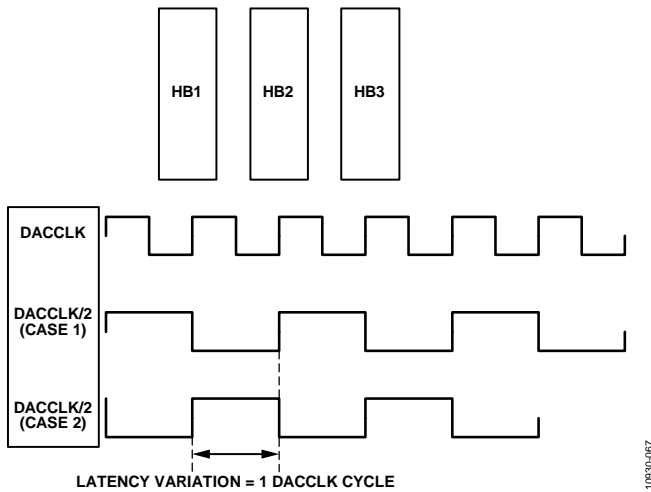


Figure 63. Latency Variation in 2x Interpolation from Clock Generation

10830-067

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 64 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 AD9142 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.

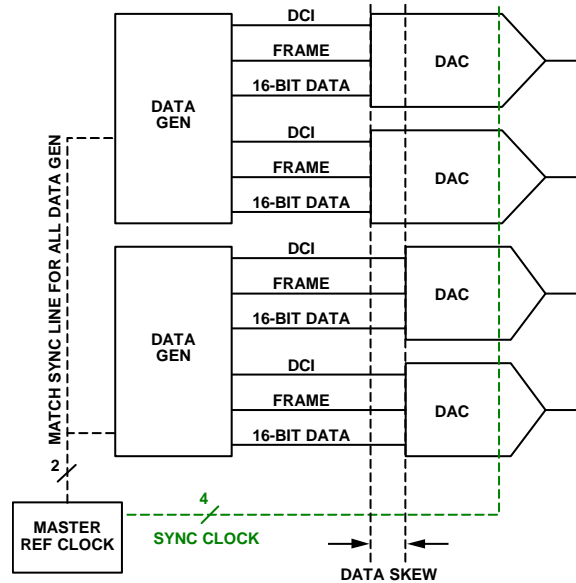
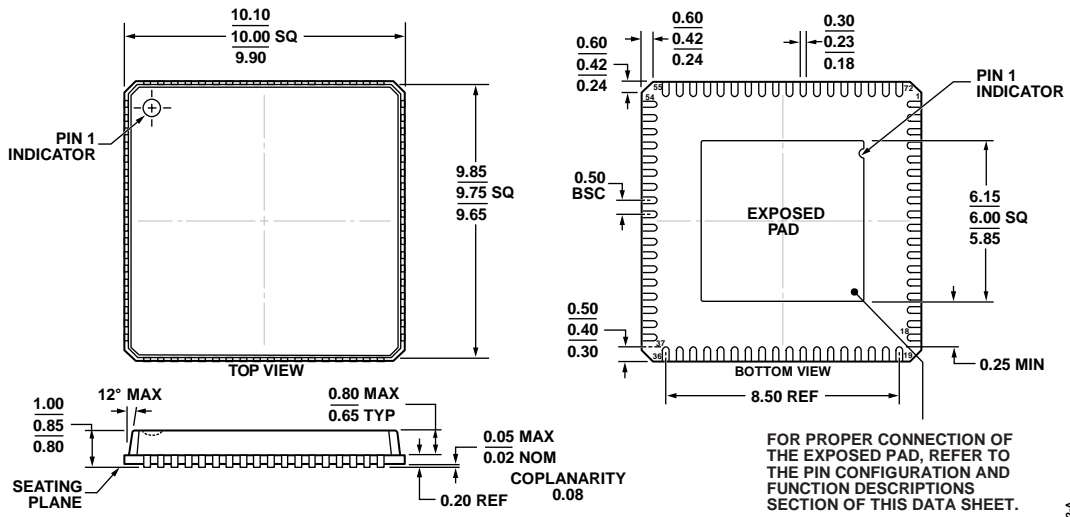


Figure 64. DAC Output Skew from Skewed Input Data and DCI

10830-068

PACKAGING AND ORDERING INFORMATION

OUTLINE DIMENSIONS



COMPLIANT TO JEDEC STANDARDS MO-220-VNND-4

Figure 65. 72-Lead Lead Frame Chip Scale Package [LFCSP_VQ]
 10 mm × 10 mm Body, Very Thin Quad
 (CP-72-7)
 Dimensions shown in millimeters

06-25-2012-A

ORDERING GUIDE

| Model ¹ | Temperature Range | Package Description | Package Option |
|--------------------|-------------------|---|----------------|
| AD9142BCPZ | -40°C to +85°C | 72-lead LFCSP_VQ | CP-72-8 |
| AD9142BCPZRL | -40°C to +85°C | 72-lead LFCSP_VQ | CP-72-7 |
| AD9142-M5372-EBZ | | Evaluation Board Connected to ADL5372 Modulator | |
| AD9142-M5375-EBZ | | Evaluation Board Connected to ADL5375 Modulator | |

¹ Z = RoHS Compliant Part.

NOTES

NOTES

NOTES