# 16-Bit, 12 GSPS, RF DAC and Digital Upconverter

Data Sheet **AD9163**

## FEATURES

- **DAC update rate up to 12 GSPS (minimum)**
- **Direct RF synthesis at 6 GSPS (minimum)**
  - **DC to 3 GHz in nonreturn-to-zero (NRZ) mode**
  - **DC to 6 GHz in 2× NRZ mode**
  - **1.5 GHz to 7.5 GHz in Mix-Mode**
- **Selectable interpolation**
  - **6×, 8×, 12×, 16×, 24×**
- **Excellent dynamic performance**

## APPLICATIONS

- **Broadband communications systems**
  - **DOCSIS 3.1 cable modem termination system (CMTS)/ video on demand (VOD)/edge quadrature amplitude modulation (EQAM)**
- **Wireless communications infrastructure**
  - **MC-GSM, W-CDMA, LTE, LTE-A, point to point**

## GENERAL DESCRIPTION

The AD9163[1] is a high performance, 16-bit digital-to-analog converter (DAC) that supports data rates to 6 GSPS. The DAC core is based on a quad-switch architecture coupled with a 2× interpolator filter that enables an effective DAC update rate of up to 12 GSPS in some modes. The high dynamic range and bandwidth makes this DAC ideally suited for the most demanding high speed radio frequency (RF) DAC applications.

Superior RF performance and deep interpolation rates enable use of the AD9163 in many wireless infrastructure applications, including MC-GSM, W-CDMA, LTE, and LTE-A.

The wide bandwidth of up to 1 GHz and the complex NCO and digital upconverter enable dual band and triple band direct RF synthesis of wireless infrastructure signals, eliminating costly analog upconverters.

Wide analog bandwidth capability combines with high dynamic range to support DOCSIS 3.1 cable infrastructure compliance from the minimum of one carrier up to 1 GHz of signal bandwidth, making it ideal for cable multiple dwelling unit (MDU) applications. A 2× interpolator filter (FIR85) enables the AD9163 to be configured for lower data rates and converter clocking to reduce the overall system power and ease the filtering requirements. In Mix-Mode™ operation, the AD9163 can reconstruct RF carriers in the second and third Nyquist zones up to 7.5 GHz while still maintaining exceptional dynamic range. The output current can be programmed from 8 mA to 38.76 mA. The AD9163 data interface consists of up to eight JESD204B serializer/deserializer (SERDES) lanes that are programmable in terms of lane speed and number of lanes to enable application flexibility.

A serial peripheral interface (SPI) configures the AD9163 and monitors the status of all the registers. The AD9163 is offered in a 169-ball, 11 mm × 11 mm, 0.8 mm pitch CSP_BGA package.

## PRODUCT HIGHLIGHTS

1. High dynamic range and signal reconstruction bandwidth supports RF signal synthesis of up to 7.5 GHz.
2. Up to eight lanes JESD204B SERDES interface, flexible in terms of number of lanes and lane speed.
3. Bandwidth and dynamic range to meet multiband wireless communications standards with margin.

## FUNCTIONAL BLOCK DIAGRAM

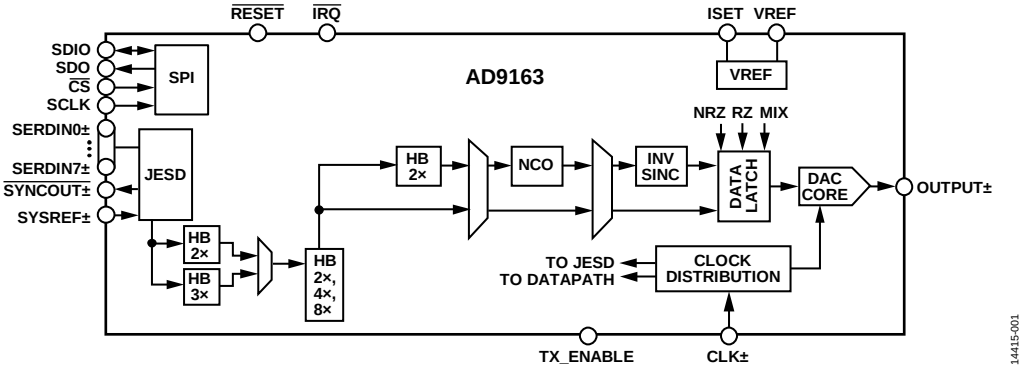


*Figure 1.*

[1] Protected by U.S. Patents 6,842,132 and 7,796,971.



**Information furnished by Analog Devices is believed to be accurate and reliable. However, no responsibility is assumed by Analog Devices for its use, nor for any infringements of patents or other rights of third parties that may result from its use. Specifications subject to change without notice. No license is granted by implication or otherwise under any patent or patent rights of Analog Devices. Trademarks and registered trademarks are the property of their respective owners.**

**One Technology Way, P.O. Box 9106, Norwood, MA 02062-9106, U.S.A.**
**Tel: 781.329.4700** **©2016–2017 Analog Devices, Inc. All rights reserved.**
**Technical Support** **www.analog.com**

# TABLE OF CONTENTS


Features ... 1
Applications ... 1
General Description ... 1
Product Highlights ... 1
Functional Block Diagram ... 1
Revision History ... 3
Specifications ... 4
DC Specifications ... 4
DAC Input Clock Overclocking Specifications ... 5
Power Supply DC Specifications ... 5
Serial Port and CMOS Pin Specifications ... 6
JESD204B Serial Interface Speed Specifications ... 7
SYSREF± to DAC Clock Timing Specifications ... 7
Digital Input Data Timing Specifications ... 8
JESD204B Interface Electrical Specifications ... 8
AC Specifications ... 9
Absolute Maximum Ratings ... 10
Reflow Profile ... 10
Thermal Management ... 10
Thermal Resistance ... 10
ESD Caution ... 10
Pin Configuration and Function Descriptions ... 11
Typical Performance Characteristics ... 13
Static Linearity ... 13
AC Performance (NRZ Mode) ... 14
AC (Mix-Mode) ... 19
DOCSIS Performance (NRZ Mode) ... 22
Terminology ... 27
Theory of Operation ... 28
Serial Port Operation ... 29
Data Format ... 29
Serial Port Pin Descriptions ... 29
Serial Port Options ... 29
JESD204B Serial Data Interface ... 31
JESD204B Overview ... 31
Physical Layer ... 32
Data Link Layer ... 35
Transport Layer ... 43
JESD204B Test Modes ... 45
JESD204B Error Monitoring ... 47
Hardware Considerations ... 49
Main Digital Datapath ... 50
Data Format ... 50
Interpolation Filters ... 50
Digital Modulation ... 53
Inverse Sinc ... 55
Downstream Protection ... 55
Datapath PRBS ... 56
Datapath PRBS IRQ ... 56
Interrupt Request Operation ... 58
Interrupt Service Routine ... 58
Applications Information ... 59
Hardware Considerations ... 59
Analog Interface Considerations ... 62
Analog Modes of Operation ... 62
Clock Input ... 63
Shuffle Mode ... 64
DLL ... 64
Voltage Reference ... 64
Temperature Sensor ... 65
Analog Outputs ... 65
Start-Up Sequence ... 67
Register Summary ... 69
Register Details ... 76
Outline Dimensions ... 126
Ordering Guide ... 126

## REVISION HISTORY

**7/2017—Rev. B to Rev. C**

Changes to Table 44 ....74
Changes to Table 45 ....118

**5/2017—Rev. A to Rev. B**

Changes to Start-Up Sequence Section ....67

**2/2017—Rev. 0 to Rev. A**

Change to Endnote 1, Table 1 ....4
Added Temperature Sensor Parameter, Table 1 ....4
Change to OUTPUT± to VNEG_N1P2 Parameter, Table 10 ....10
Changes to Figure 99 ....38
Changes to Link Delay Setup Example, With Known Delays Section ....40
Changes to Link Delay Setup Example, Without Known Delays Section ....41
Changes to Table 23 ....43
Changed 4× Six-Lane (L = 3, M = 2, F = 4, S = 3) to 4× Three-Lane (L = 3, M = 2, F = 4, S = 3), Table 29 ....46
Added Datapath PRBS Section and Datapath PRBS IRQ Section ....56
Moved Figure 133 ....64
Added Temperature Sensor Section ....65
Changes to Equivalent DAC Output and Transfer Function Section ....65
Changes to Output Stage Configuration Section and Figure 140 Caption ....66
Added Register 0x132 Row to Register 0x135 Row, Table 44 ....70
Added Register 0x132 Row to Register 0x135 Row, Table 45 ....84
Change to Register 0x230 ....86

**7/2016—Revision 0: Initial Version**

# SPECIFICATIONS

## DC SPECIFICATIONS

VDD25_DAC = 2.5 V, VDD12A = VDD12_CLK = 1.2 V, VNEG_N1P2 = −1.2 V, DVDD = 1.2 V, IOVDD = 2.5 V, VDD_1P2 = DVDD_1P2 = PLL_LDO_VDD12 = 1.2 V, SYNC_VDD_3P3 = 3.3 V, DAC output full-scale current ($I_{OUTFS}$) = 40 mA, and $T_A$ = −40°C to +85°C, unless otherwise noted.

**Table 1.**

| Parameter | Test Conditions/Comments | Min | Typ | Max | Unit |
|---|---|---|---|---|---|
| RESOLUTION | | 16 | | | Bit |
| DAC Update Rate | | | | | |
| Minimum | | | | 1.5 | GSPS |
| Maximum | VDDx[1] = 1.3 V ± 2%[2] | 6 | 6.4 | | GSPS |
| | VDDx[1] = 1.3 V ± 2%[2], FIR85[3] 2× interpolator enabled | 12 | 12.8 | | GSPS |
| Adjusted[4] | VDDx[1] = 1.3 V ± 2%[2] | 1 | 1.0667 | | GSPS |
| ACCURACY | | | | | |
| Integral Nonlinearity (INL) | | | ±2.7 | | LSB |
| Differential Nonlinearity (DNL) | | | ±1.7 | | LSB |
| ANALOG OUTPUTS | | | | | |
| Gain Error (with Internal Reference) | | | −1.7 | | % |
| Full-Scale Output Current | | | | | |
| Minimum | $R_{SET}$ = 9.76 kΩ | 7.37 | 8 | 8.57 | mA |
| Maximum | $R_{SET}$ = 9.76 kΩ | 35.8 | 38.76 | 41.3 | mA |
| DAC CLOCK INPUT (CLK+, CLK−) | | | | | |
| Differential Input Power | $R_{LOAD}$ = 90 Ω differential on chip | −20 | 0 | +10 | dBm |
| Common-Mode Voltage | AC-coupled | | 0.6 | | V |
| Input Impedance[1] | 3 GSPS input clock | | 90 | | Ω |
| TEMPERATURE DRIFT | | | | | |
| Gain | | | 105 | | ppm/°C |
| Reference Voltage | | | 75 | | ppm/°C |
| TEMPERATURE SENSOR | | | | | |
| Accuracy | After single point calibration (see the Temperature Sensor section) | | ±5 | | % |
| REFERENCE | | | | | |
| Internal Reference Voltage | | | 1.19 | | V |
| ANALOG SUPPLY VOLTAGES | | | | | |
| VDD25_DAC | | 2.375 | 2.5 | 2.625 | V |
| VDD12A[2] | | 1.14 | 1.2 | 1.326 | V |
| VDD12_CLK[2] | | 1.14 | 1.2 | 1.326 | V |
| VNEG_N1P2 | | −1.26 | −1.2 | −1.14 | V |
| DIGITAL SUPPLY VOLTAGES | | | | | |
| DVDD | | 1.14 | 1.2 | 1.326 | V |
| IOVDD[3] | | 1.71 | 2.5 | 3.465 | V |
| SERDES SUPPLY VOLTAGES | | | | | |
| VDD_1P2 | | 1.14 | 1.2 | 1.326 | V |
| VTT_1P2 | Can connect to VDD_1P2 | 1.14 | 1.2 | 1.326 | V |
| DVDD_1P2 | | 1.14 | 1.2 | 1.326 | V |
| PLL_LDO_VDD12 | | 1.14 | 1.2 | 1.326 | V |
| PLL_CLK_VDD12 | Can connect to PLL_LDO_VDD12 | 1.14 | 1.2 | 1.326 | V |
| SYNC_VDD_3P3 | | 3.135 | 3.3 | 3.465 | V |
| BIAS_VDD_1P2 | Can connect to VDD_1P2 | 1.14 | 1.2 | 1.326 | V |

[1] See the Clock Input section for more details.
[2] For the lowest noise performance, use a separate power supply filter network for the VDD12_CLK and the VDD12A pins.
[3] IOVDD can range from 1.8 V to 3.3 V, with ±5% tolerance.
[4] The adjusted DAC update rate is calculated as $f_{DAC}$ divided by the minimum required interpolation factor. For the AD9163, the minimum interpolation factor is 6. Therefore, with $f_{DAC}$ = 6 GSPS, $f_{DAC}$ adjusted = 1 GSPS. When FIR85 is enabled, which puts the device into 2× NRZ mode, $f_{DAC}$ = 2 × (DAC clock input frequency), and the minimum interpolation increases to 12× (interpolation value). Thus, for the AD9163, with FIR85 enabled and DAC clock = 6 GSPS, $f_{DAC}$ = 12 GSPS, minimum interpolation = 12×, and the adjusted DAC update rate = 1 GSPS.

## DAC INPUT CLOCK OVERCLOCKING SPECIFICATIONS

VDD25_DAC = 2.5 V, VDD12A = VDD12_CLK = 1.2 V, VNEG_N1P2 = −1.2 V, DVDD = 1.2 V, IOVDD = 2.5 V, VDD_1P2 = DVDD_1P2 = PLL_LDO_VDD12 = 1.2 V, SYNC_VDD_3P3 = 3.3 V, $I_{OUTFS}$ = 40 mA, $T_A$ = −40°C to +85°C, unless otherwise noted.

Maximum guaranteed speed using the temperatures and voltages conditions as shown in Table 2, where VDDx is VDD12_CLK, DVDD, VDD_1P2, DVDD_1P2, and PLL_LDO_VDD12. Any DAC clock speed over 5.1 GSPS requires a maximum junction temperature of 105°C to avoid damage to the device. See Table 10 for details on maximum junction temperature permitted for certain clock speeds.

**Table 2.**

| Parameter[1] | Test Conditions/Comments | Min | Typ | Max | Unit |
|---|---|---|---|---|---|
| MAXIMUM DAC UPDATE RATE | | | | | |
| VDDx = 1.2 V ± 5% | $T_{JMAX}$ = 25°C | 6.0 | | | GSPS |
| | $T_{JMAX}$ = 85°C | 5.6 | | | GSPS |
| | $T_{JMAX}$ = 105°C | 5.4 | | | GSPS |
| VDDx = 1.2 V ± 2% | $T_{JMAX}$ = 25°C | 6.1 | | | GSPS |
| | $T_{JMAX}$ = 85°C | 5.8 | | | GSPS |
| | $T_{JMAX}$ = 105°C | 5.6 | | | GSPS |
| VDDx = 1.3 V ± 2% | $T_{JMAX}$ = 25°C | 6.4 | | | GSPS |
| | $T_{JMAX}$ = 85°C | 6.2 | | | GSPS |
| | $T_{JMAX}$ = 105°C | 6.0 | | | GSPS |

[1] $T_{JMAX}$ is the maximum junction temperature.

## POWER SUPPLY DC SPECIFICATIONS

$I_{OUTFS}$ = 40 mA, $T_A$ = −40°C to +85°C, unless otherwise noted. FIR85 is the finite impulse response with 85 dB digital attenuation.

**Table 3.**

| Parameter | Test Conditions/Comments | Min | Typ | Max | Unit |
|---|---|---|---|---|---|
| 8 LANES, 6× INTERPOLATION (80%), 3 GSPS | NCO on, FIR85 on | | | | |
| Analog Supply Currents | | | | | |
| VDD25_DAC = 2.5 V | | | 93.8 | | mA |
| VDD12A = 1.2 V | | | 3.7 | | µA |
| VDD12_CLK = 1.2 V | | | 228.7 | | mA |
| VNEG_N1P2 = −1.2 V | | | −120.7 | | mA |
| Digital Supply Currents | | | | | |
| DVDD = 1.2 V | | | 598.4 | | mA |
| IOVDD = 2.5 V | | | 2.5 | | mA |
| SERDES Supply Currents | | | | | |
| VDD_1P2 = 1.2 V | Includes VTT_1P2, BIAS_VDD_1P2 | | 443.4 | | mA |
| DVDD_1P2 = 1.2 V | | | 72.3 | | mA |
| PLL_LDO_VDD12 = 1.2 V | Connected to PLL_CLK_VDD12 | | 81.8 | | mA |
| SYNC_VDD_3P3 = 3.3 V | | | 9.4 | | mA |
| 8 LANES, 8× INTERPOLATION (80%), 5 GSPS | NCO on, FIR85 off (unless otherwise noted) | | | | |
| Analog Supply Currents | | | | | |
| VDD25_DAC = 2.5 V | | | 94 | 100 | mA |
| VDD12A = 1.2 V | | | 80 | 150 | µA |
| VDD12_CLK = 1.2 V | | | 341 | 435 | mA |
| VNEG_N1P2 = −1.2 V | | −119 | −112 | | mA |
| Digital Supply Currents | | | | | |
| DVDD = 1.2 V | NCO on, FIR85 off | | 495 | 878 | mA |
| IOVDD = 2.5 V | | | 2.5 | 2.7 | mA |
| SERDES Supply Currents | | | | | |
| VDD_1P2 = 1.2 V | Includes VTT_1P2, BIAS_VDD_1P2 | | 477 | 681 | mA |
| DVDD_1P2 = 1.2 V | | | 89 | 130 | mA |
| PLL_LDO_VDD12 = 1.2 V | Connected to PLL_CLK_VDD12 | | 81 | 112 | mA |
| SYNC_VDD_3P3 = 3.3 V | | | 9.3 | 11 | mA |

| Parameter | Test Conditions/Comments | Min | Typ | Max | Unit |
|---|---|---|---|---|---|
| POWER DISSIPATION | | | | | |
| 3 GSPS | | | | | |
| 2× NRZ Mode, 6×, FIR85 Enabled, NCO On | Using 80%, 3× filter, eight-lane JESD204B | | 2.1 | | W |
| NRZ Mode, 24×, FIR85 Disabled, NCO On | Using 80%, 2× filter, one-lane JESD204B | | 1.3 | | W |
| 5 GSPS | | | | | |
| NRZ Mode, 8×, FIR85 Disabled, NCO On | Using 80%, 2× filter, eight-lane JESD204B | | 2.18 | | W |
| NRZ Mode, 16×, FIR85 Disabled, NCO On | Using 80%, 2× filter, eight-lane JESD204B | | 2.09 | | W |
| 2× NRZ Mode, 6×, FIR85 Enabled, NCO On | Using 80%, 3× filter, eight-lane JESD204B | | 2.65 | | W |

## SERIAL PORT AND CMOS PIN SPECIFICATIONS

VDD25_DAC = 2.5 V, VDD12A = VDD12_CLK = 1.2 V, VNEG_N1P2 = −1.2 V, DVDD = 1.2 V, IOVDD = 2.5 V, VDD_1P2 = DVDD_1P2 = PLL_LDO_VDD12 = 1.2 V, SYNC_VDD_3P3 = 3.3 V, $I_{OUTFS}$ = 40 mA, $T_A$ = −40°C to +85°C, unless otherwise noted.

**Table 4.**

| Parameter | Symbol | Test Comments/Conditions | Min | Typ | Max | Unit |
|---|---|---|---|---|---|---|
| WRITE OPERATION | | See Figure 89 | | | | |
| Maximum SCLK Clock Rate | $f_{SCLK}$, $1/t_{SCLK}$ | | 100 | | | MHz |
| SCLK Clock High | $t_{PWH}$ | SCLK = 20 MHz | 3.5 | | | ns |
| SCLK Clock Low | $t_{PWL}$ | SCLK = 20 MHz | 4 | | | ns |
| SDIO to SCLK Setup Time | $t_{DS}$ | | 4 | 2 | | ns |
| SCLK to SDIO Hold Time | $t_{DH}$ | | 1 | 0.5 | | ns |
| $\overline{CS}$ to SCLK Setup Time | $t_S$ | | 9 | 1 | | ns |
| SCLK to $\overline{CS}$ Hold Time | $t_H$ | | 9 | 0.5 | | ns |
| READ OPERATION | | See Figure 88 | | | | |
| SCLK Clock Rate | $f_{SCLK}$, $1/t_{SCLK}$ | | | | 20 | MHz |
| SCLK Clock High | $t_{PWH}$ | | 20 | | | ns |
| SCLK Clock Low | $t_{PWL}$ | | 20 | | | ns |
| SDIO to SCLK Setup Time | $t_{DS}$ | | 10 | | | ns |
| SCLK to SDIO Hold Time | $t_{DH}$ | | 5 | | | ns |
| $\overline{CS}$ to SCLK Setup Time | $t_S$ | | 10 | | | ns |
| SCLK to SDIO (or SDO) Data Valid Time | $t_{DV}$ | | | | 17 | ns |
| $\overline{CS}$ to SDIO (or SDO) Output Valid to High-Z | | Not shown in Figure 88 or Figure 89 | | | 45 | ns |
| INPUTS (SDIO, SCLK, $\overline{CS}$, $\overline{RESET}$, TX_ENABLE) | | | | | | |
| Voltage Input | | | | | | |
| High | $V_{IH}$ | 1.8 V ≤ IOVDD ≤ 2.5 V | 0.7 × IOVDD | | | V |
| Low | $V_{IL}$ | 1.8 V ≤ IOVDD ≤ 2.5 V | | | 0.3 × IOVDD | V |
| Current Input | | | | | | |
| High | $I_{IH}$ | | | | 75 | µA |
| Low | $I_{IL}$ | | −150 | | | µA |
| OUTPUTS (SDIO, SDO) | | | | | | |
| Voltage Output | | | | | | |
| High | $V_{OH}$ | 1.8 V ≤ IOVDD ≤ 3.3 V | 0.8 × IOVDD | | | V |
| Low | $V_{OL}$ | 1.8 V ≤ IOVDD ≤ 3.3 V | | | 0.2 × IOVDD | V |
| Current Output | | | | | | |
| High | $I_{OH}$ | | | 4 | | mA |
| Low | $I_{OL}$ | | | 4 | | mA |

## JESD204B SERIAL INTERFACE SPEED SPECIFICATIONS

VDD25_DAC = 2.5 V, VDD12A = VDD12_CLK = 1.2 V, VNEG_N1P2 = −1.2 V, DVDD = 1.2 V, IOVDD = 2.5 V, VDD_1P2 = DVDD_1P2 = PLL_LDO_VDD12 = 1.2 V, SYNC_VDD_3P3 = 3.3 V, $I_{OUTFS}$ = 40 mA, $T_A$ = −40°C to +85°C, unless otherwise noted.

**Table 5.**

| Parameter | Test Conditions/Comments | Min | Typ | Max | Unit |
|---|---|---|---|---|---|
| SERIAL INTERFACE SPEED | Guaranteed operating range | | | | |
| Half Rate | | 6 | | 12.5 | Gbps |
| Full Rate | | 3 | | 6.25 | Gbps |
| Oversampling | | 1.5 | | 3.125 | Gbps |
| 2× Oversampling | | 0.750 | | 1.5625 | Gbps |

## SYSREF± TO DAC CLOCK TIMING SPECIFICATIONS

VDD25_DAC = 2.5 V, VDD12A = VDD12_CLK = 1.2 V, VNEG_N1P2 = −1.2 V, DVDD = 1.2 V, IOVDD = 2.5 V, VDD_1P2 = DVDD_1P2 = PLL_LDO_VDD12 = 1.2 V, SYNC_VDD_3P3 = 3.3 V, $I_{OUTFS}$ = 40 mA, $T_A$ = −40°C to +85°C, unless otherwise noted.

**Table 6.**

| Parameter[1] | Test Conditions/Comments | Min | Typ | Max | Unit |
|---|---|---|---|---|---|
| SYSREF± DIFFERENTIAL SWING = 1.0 V | | | | | |
| Minimum Setup Time, $t_{SYSS}$ | AC-coupled | | 65 | 117 | ps |
| | DC-coupled, common-mode voltage = 0 V | | 45 | 77 | ps |
| | DC-coupled, common-mode voltage = 1.25 V | | 68 | 129 | ps |
| Minimum Hold Time, $t_{SYSH}$ | AC-coupled | | 19 | 63 | ps |
| | DC-coupled, common-mode voltage = 0 V | | 5 | 37 | ps |
| | DC-coupled, common-mode voltage = 1.25 V | | 51 | 114 | ps |

[1] The SYSREF± pulse must be at least four DAC clock edges wide plus the setup and hold times in Table 6. For more information, see the Sync Processing Modes Overview section.

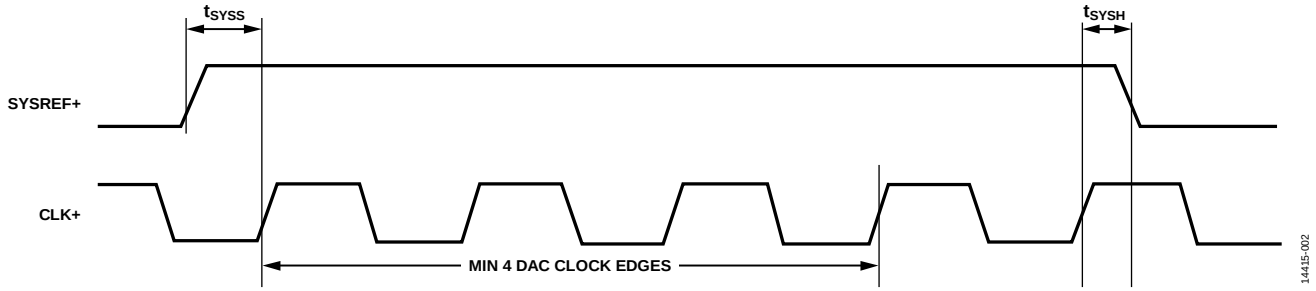


*Figure 2. SYSREF± to DAC Clock Timing Diagram (Only SYSREF+ and CLK+ Shown)*

## DIGITAL INPUT DATA TIMING SPECIFICATIONS

VDD25_DAC = 2.5 V, VDD12A = VDD12_CLK = 1.2 V, VNEG_N1P2 = −1.2 V, DVDD = 1.2 V, IOVDD = 2.5 V, VDD_1P2 = DVDD_1P2 = PLL_LDO_VDD12 = 1.2 V, SYNC_VDD_3P3 = 3.3 V, $I_{OUTFS}$ = 40 mA, $T_A$ = −40°C to +85°C, unless otherwise noted.

**Table 7.**

| Parameter | Test Conditions/Comments | Min | Typ | Max | Unit |
|---|---|---|---|---|---|
| LATENCY[1] | | | | | |
| Interface | | | 1 | | PCLK[2] cycle |
| Interpolation | | | See Table 32 | | |
| Power-Up Time | From DAC output off to enabled | | 10 | | ns |
| DETERMINISTIC LATENCY | | | | | |
| Fixed | | | | 12 | PCLK[2] cycles |
| Variable | | | | 2 | PCLK[2] cycles |
| SYSREF± to LOCAL MULTIFRAME CLOCKS (LMFC) DELAY | | | 4 | | DAC clock cycles |

[1] Total latency (or pipeline delay) through the device is calculated as follows:
*Total Latency = Interface Latency + Fixed Latency + Variable Latency + Pipeline Delay*
See Table 32 for examples of the pipeline delay per block.
[2] PCLK is the internal processing clock for the AD9163 and equals the lane rate ÷ 40.

## JESD204B INTERFACE ELECTRICAL SPECIFICATIONS

VDD25_DAC = 2.5 V, VDD12A = VDD12_CLK = 1.2 V, VNEG_N1P2 = −1.2 V, DVDD = 1.2 V, IOVDD = 2.5 V, VDD_1P2 = DVDD_1P2 = PLL_LDO_VDD12 = 1.2 V, SYNC_VDD_3P3 = 3.3 V, $I_{OUTFS}$ = 40 mA, $T_A$ = −40°C to +85°C, unless otherwise noted. $V_{TT}$ is the termination voltage.

**Table 8.**

| Parameter | Symbol | Test Conditions/Comments | Min | Typ | Max | Unit |
|---|---|---|---|---|---|---|
| JESD204B DATA INPUTS | | | | | | |
| Input Leakage Current | | $T_A$ = 25°C | | | | |
| Logic High | | Input level = 1.2 V ± 0.25 V, $V_{TT}$ = 1.2 V | | 10 | | µA |
| Logic Low | | Input level = 0 V | | −4 | | µA |
| Unit Interval | UI | | 80 | | 1333 | ps |
| Common-Mode Voltage | $V_{RCM}$ | AC-coupled, $V_{TT}$ = VDD_1P2[1] | −0.05 | | +1.85 | V |
| Differential Voltage | $R_V_{DIFF}$ | | 110 | | 1050 | mV |
| $V_{TT}$ Source Impedance | $Z_{TT}$ | At dc | | | 30 | Ω |
| Differential Impedance | $Z_{RDIFF}$ | At dc | 80 | 100 | 120 | Ω |
| Differential Return Loss | $RL_{RDIF}$ | | | 8 | | dB |
| Common-Mode Return Loss | $RL_{RCM}$ | | | 6 | | dB |
| SYSREF± INPUT | | | | | | |
| Differential Impedance | | 169-ball CSP_BGA | | 121 | | Ω |
| DIFFERENTIAL OUTPUTS ($\overline{SYNCOUT}$±)[2] | | Driving 100 Ω differential load | | | | |
| Output Differential Voltage | $V_{OD}$ | | 350 | 420 | 450 | mV |
| Output Offset Voltage | $V_{OS}$ | | 1.15 | 1.2 | 1.27 | V |

[1] As measured on the input side of the ac coupling capacitor.
[2] IEEE Standard 1596.3 LVDS compatible.

## AC SPECIFICATIONS

VDD25_DAC = 2.5 V, VDD12A = VDD12_CLK = 1.2 V, VNEG_N1P2 = −1.2 V, DVDD = 1.2 V, IOVDD = 2.5 V, VDD_1P2 = DVDD_1P2 = PLL_LDO_VDD12 = 1.2 V, SYNC_VDD_3P3 = 3.3 V, $I_{OUTFS}$ = 40 mA, $T_A$ = 25°C, unless otherwise noted.

**Table 9. AC Specifications**

| Parameter | Test Conditions/Comments | Min | Typ | Max | Unit |
|---|---|---|---|---|---|
| SPURIOUS-FREE DYNAMIC RANGE (SFDR)[1] | | | | | |
| Single Tone, $f_{DAC}$ = 5000 MSPS | | | | | |
| $f_{OUT}$ = 70 MHz | | | −82 | | dBc |
| $f_{OUT}$ = 500 MHz | | | −75 | | dBc |
| $f_{OUT}$ = 1000 MHz | | | −65 | | dBc |
| $f_{OUT}$ = 2000 MHz | | | −70 | | dBc |
| $f_{OUT}$ = 4000 MHz | FIR85 enabled | | −60 | | dBc |
| Single Tone, $f_{DAC}$ = 5000 MSPS | −6 dBFS, shuffle enabled | | | | |
| $f_{OUT}$ = 70 MHz | | | −75 | | dBc |
| $f_{OUT}$ = 500 MHz | | | −75 | | dBc |
| $f_{OUT}$ = 1000 MHz | | | −70 | | dBc |
| $f_{OUT}$ = 2000 MHz | | | −75 | | dBc |
| $f_{OUT}$ = 4000 MHz | FIR85 enabled | | −65 | | dBc |
| DOCSIS | $f_{DAC}$ = 3076 MSPS | | | | |
| $f_{OUT}$ = 70 MHz | Single carrier | | −70 | | dBc |
| $f_{OUT}$ = 70 MHz | Four carriers | | −70 | | dBc |
| $f_{OUT}$ = 70 MHz | Eight carriers | | −67 | | dBc |
| $f_{OUT}$ = 950 MHz | Single carrier | | −70 | | dBc |
| $f_{OUT}$ = 950 MHz | Four carriers | | −68 | | dBc |
| $f_{OUT}$ = 950 MHz | Eight carriers | | −64 | | dBc |
| Wireless Infrastructure | $f_{DAC}$ = 5000 MSPS | | | | |
| $f_{OUT}$ = 960 MHz | Two-carrier GSM signal at −9 dBFS; across 925 MHz to 960 MHz band | | −85 | | dBc |
| $f_{OUT}$ = 1990 MHz | Two-carrier GSM signal at −9 dBFS; across 1930 MHz to 1990 MHz band | | −81 | | dBc |
| ADJACENT CHANNEL POWER | $f_{DAC}$ = 5000 MSPS | | | | |
| $f_{OUT}$ = 877 MHz | One carrier, first adjacent channel | | −79 | | dBc |
| $f_{OUT}$ = 877 MHz | Two carriers, first adjacent channel | | −76 | | dBc |
| $f_{OUT}$ = 1887 MHz | One carrier, first adjacent channel | | −74 | | dBc |
| $f_{OUT}$ = 1980 MHz | Four carriers, first adjacent channel | | −70 | | dBc |
| INTERMODULATION DISTORTION | $f_{DAC}$ = 5000 MSPS, two-tone test | | | | |
| $f_{OUT}$ = 900 MHz | 0 dBFS | | −80 | | dBc |
| $f_{OUT}$ = 900 MHz | −6 dBFS, shuffle enabled | | −80 | | dBc |
| $f_{OUT}$ = 1800 MHz | 0 dBFS | | −68 | | dBc |
| $f_{OUT}$ = 1800 MHz | −6 dBFS, shuffle enabled | | −78 | | dBc |
| NOISE SPECTRAL DENSITY (NSD) | | | | | |
| Single Tone, $f_{DAC}$ = 5000 MSPS | | | | | |
| $f_{OUT}$ = 550 MHz | | | −168 | | dBm/Hz |
| $f_{OUT}$ = 960 MHz | | | −167 | | dBm/Hz |
| $f_{OUT}$ = 1990 MHz | | | −164 | | dBm/Hz |
| SINGLE SIDEBAND (SSB) PHASE NOISE AT OFFSET | $f_{OUT}$ = 3800 MHz, $f_{DAC}$ = 4000 MSPS | | | | |
| 1 kHz | | | −119 | | dBc/Hz |
| 10 kHz | | | −125 | | dBc/Hz |
| 100 kHz | | | −135 | | dBc/Hz |
| 1 MHz | | | −144 | | dBc/Hz |
| 10 MHz | | | −156 | | dBc/Hz |

[1] See the Clock Input section for more details on optimizing SFDR and reducing the image of the fundamental with clock input tuning.

# ABSOLUTE MAXIMUM RATINGS

**Table 10.**

| Parameter | Rating |
|---|---|
| ISET, VREF to VBG_NEG | −0.3 V to VDD25_DAC + 0.3 V |
| SERDINx±, VTT_1P2, SYNCOUT± | −0.3 V to SYNC_VDD_3P3 + 0.3 V |
| OUTPUT± to VNEG_N1P2 | −0.3 V to VDD25_DAC + 0.2 V |
| SYSREF± | GND − 0.5 V to +2.5 V |
| CLK± to Ground | −0.3 V to VDD12_CLK + 0.3 V |
| RESET, IRQ, CS, SCLK, SDIO, SDO to Ground | −0.3 V to IOVDD + 0.3 V |
| Junction Temperature[1] | |
| $f_{DAC}$ = 6 GSPS | 105°C |
| $f_{DAC}$ ≤ 5.1 GSPS | 110°C |
| Ambient Operating Temperature Range ($T_A$) | −40°C to +85°C |
| Storage Temperature Range | −65°C to +150°C |

[1] Some operating modes of the device may cause the device to approach or exceed the maximum junction temperature during operation at supported ambient temperatures. Removal of heat from the device may require additional measures such as active airflow, heat sinks, or other measures.

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

## REFLOW PROFILE

The AD9163 reflow profile is in accordance with the JEDEC JESD204B criteria for Pb-free devices. The maximum reflow temperature is 260°C.

## THERMAL MANAGEMENT

The AD9163 is a high power device that can dissipate nearly 3 W depending on the user application and configuration. Because of the power dissipation, the AD9163 uses an exposed die package to give the customer the most effective method of controlling the die temperature. The exposed die allows cooling of the die directly.

Figure 3 shows the profile view of the device mounted to a user printed circuit board (PCB) and a heat sink (typically the aluminum case) to keep the junction (exposed die) below the maximum junction temperature in Table 10.

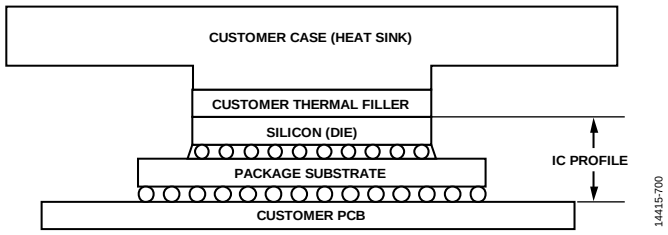


*Figure 3. Typical Thermal Management Solution*

## THERMAL RESISTANCE

Typical $\theta_{JA}$ and $\theta_{JC}$ values are specified for a 4-layer JEDEC 2S2P high effective thermal conductivity test board for balled surface-mount packages. $\theta_{JA}$ is obtained in still air conditions (JESD51-2). Airflow increases heat dissipation, effectively reducing $\theta_{JA}$. $\theta_{JC}$ is obtained with the test case temperature monitored at the bottom of the package.

$\Psi_{JT}$ is thermal characteristic parameters obtained with $\theta_{JA}$ in still air test conditions but are not applicable to the CSP_BGA package.

Estimate the junction temperature ($T_J$) using the following equations:

$$T_J = T_T + (\Psi_{JT} \times P_{DISS})$$

where:
$T_T$ is the temperature measured at the top of the package.
$P_{DISS}$ is the total device power dissipation.

**Table 11. Thermal Resistance**

| Package Type | $\theta_{JA}$ | $\theta_{JC}$ | Unit |
|---|---|---|---|
| BC-169-2 | 14.6 | 0.02 | °C/W |

## ESD CAUTION



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

# PIN CONFIGURATION AND FUNCTION DESCRIPTIONS

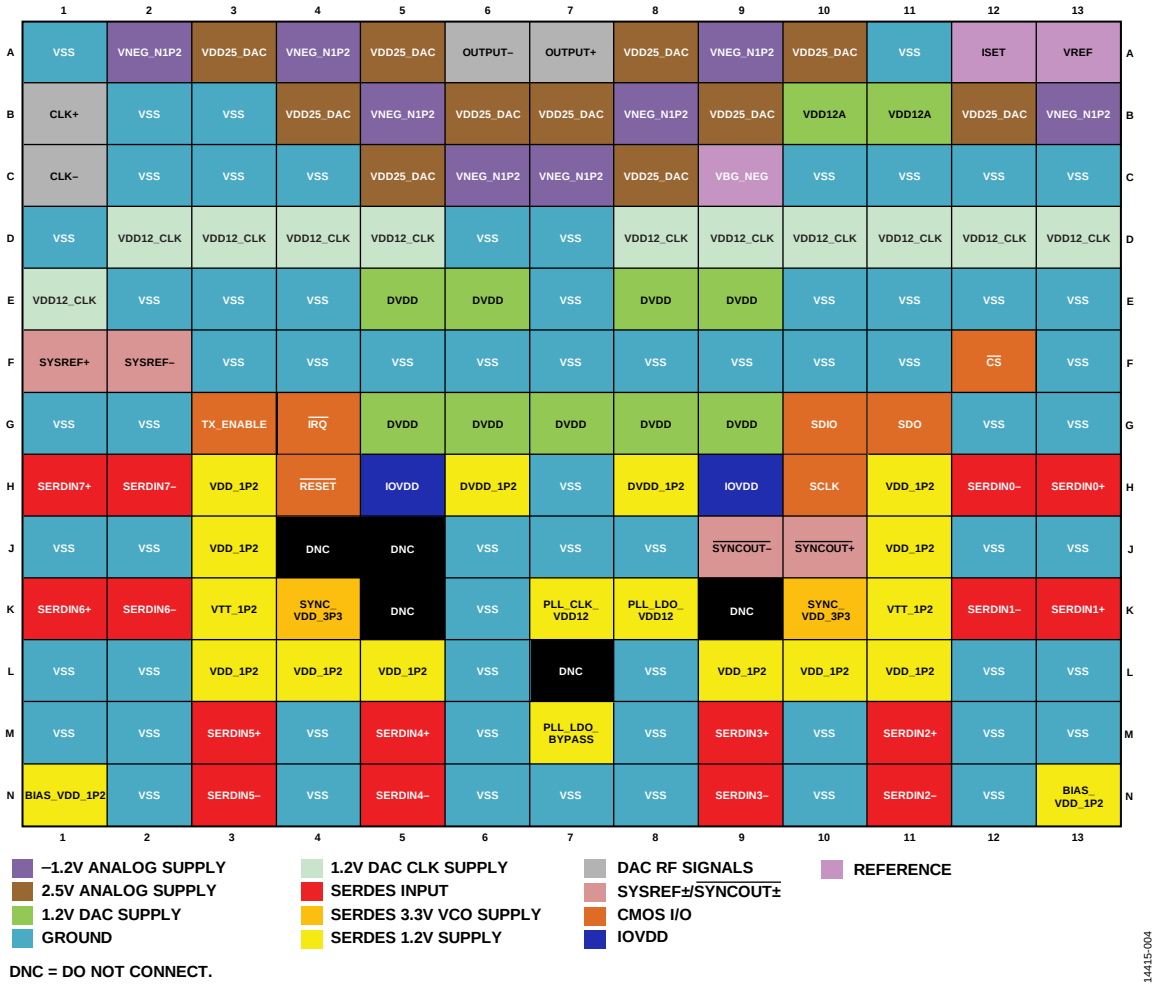


*Figure 4. 169-Ball CSP_BGA Pin Configuration*

**Table 12. 169-Ball CSP_BGA Pin Function Descriptions**

| Pin No. | Mnemonic | Description |
|---|---|---|
| A1, A11, B2, B3, C2, C3, C4, C10, C11, C12, C13, D1, D6, D7, E2, E3, E4, E7, E10, E11, E12, E13, F3, F4, F5, F6, F7, F8, F9, F10, F11, F13, G1, G2, G12, G13, H7, J1, J2, J6, J7, J8, J12, J13, K6, L1, L2, L6, L8, L12, L13, M1, M2, M4, M6, M8, M10, M12, M13, N2, N4, N6, N7, N8, N10, N12 | VSS | Supply Return. Connect these pins to ground. |
| A2, A4, A9, B5, B8, B13, C6, C7 | VNEG_N1P2 | –1.2 V Analog Supply Voltage. |
| A3, A5, A8, A10, B4, B6, B7, B9, B12, C5, C8 | VDD25_DAC | 2.5 V Analog Supply Voltage. |
| A6 | OUTPUT– | DAC Negative Current Output. |
| A7 | OUTPUT+ | DAC Positive Current Output. |
| A12 | ISET | Reference Current. Connect this pin to VNEG_N1P2 with a 9.6 kΩ resistor. |
| A13 | VREF | 1.2 V Reference Input/Output. Connect this pin to VSS with a 1 µF capacitor. |
| B1, C1 | CLK+, CLK– | Positive and Negative DAC Clock Inputs. |
| B10, B11 | VDD12A | 1.2 V Analog Supply Voltage. |
| C9 | VBG_NEG | –1.2 V Reference. Connect this pin to VNEG_N1P2 with a 0.1 µF capacitor. |
| D2, D3, D4, D5, D8, D9, D10, D11, D12, D13, E1 | VDD12_CLK | 1.2 V Clock Supply Voltage. |
| E5, E6, E8, E9, G5, G6, G7, G8, G9 | DVDD | 1.2 V Digital Supply Voltage. |

| Pin No. | Mnemonic | Description |
|---|---|---|
| F1, F2 | SYSREF+, SYSREF− | System Reference Positive and Negative Inputs. These pins are self biased for ac coupling. They can be ac-coupled or dc-coupled. |
| F12 | $\overline{CS}$ | Serial Port Chip Select Bar (Active Low) Input. CMOS levels on this pin are determined with respect to IOVDD. |
| G3 | TX_ENABLE | Transmit Enable Input. This pin can be used instead of the DAC output bias power-down bits in Register 0x040, Bits[1:0] to enable the DAC output. CMOS levels are determined with respect to IOVDD. |
| G4 | $\overline{IRQ}$ | Interrupt Request Output (Active Low, Open Drain). |
| G10 | SDIO | Serial Port Data Input/Output. CMOS levels on this pin are determined with respect to IOVDD. |
| G11 | SDO | Serial Port Data Output. CMOS levels on this pin are determined with respect to IOVDD. |
| H10 | SCLK | Serial Port Data Clock. CMOS levels on this pin are determined with respect to IOVDD. |
| H3, H11, J3, J11, L3, L4, L5, L9, L10, L11 | VDD_1P2 | 1.2 V SERDES Digital Supply. |
| H4 | $\overline{RESET}$ | Reset Bar (Active Low) Input. CMOS levels on this pin are determined with respect to IOVDD. |
| H5, H9 | IOVDD | Supply Voltage for CMOS Input/Output and SPI. Operational for 1.8 V to 3.3 V (see Table 1 for details). |
| H6, H8 | DVDD_1P2 | 1.2 V SERDES Digital Supply Voltage. |
| H1, H2 | SERDIN7+, SERDIN7− | SERDES Lane 7 Positive and Negative Inputs. |
| K1, K2 | SERDIN6+, SERDIN6− | SERDES Lane 6 Positive and Negative Inputs. |
| M3, N3 | SERDIN5+, SERDIN5− | SERDES Lane 5 Positive and Negative Inputs. |
| M5, N5 | SERDIN4+, SERDIN4− | SERDES Lane 4 Positive and Negative Inputs. |
| M9, N9 | SERDIN3+, SERDIN3− | SERDES Lane 3 Positive and Negative Inputs. |
| M11, N11 | SERDIN2+, SERDIN2− | SERDES Lane 2 Positive and Negative Inputs. |
| K12, K13 | SERDIN1−, SERDIN1+ | SERDES Lane 1 Negative and Positive Inputs. |
| H12, H13 | SERDIN0−, SERDIN0+ | SERDES Lane 0 Negative and Positive Inputs. |
| J4, J5, K5, K9, L7 | DNC | Do Not Connect. Do not connect to these pins. |
| J9, J10 | $\overline{SYNCOUT}$−, $\overline{SYNCOUT}$+ | Negative and Positive LVDS Sync (Active Low) Output Signals. |
| K3, K11 | VTT_1P2 | 1.2 V SERDES $V_{TT}$ Digital Supply Voltage. |
| K4, K10 | SYNC_VDD_3P3 | 3.3 V SERDES Sync Supply Voltage. |
| K7 | PLL_CLK_VDD12 | 1.2 V SERDES PLL Clock Supply Voltage. |
| K8 | PLL_LDO_VDD12 | 1.2 V SERDES PLL Supply. |
| M7 | PLL_LDO_BYPASS | 1.2 V SERDES PLL Supply Voltage Bypass. |
| N1, N13 | BIAS_VDD_1P2 | 1.2 V SERDES Supply Voltage. |

# TYPICAL PERFORMANCE CHARACTERISTICS

## STATIC LINEARITY

$I_{OUTFS}$ = 40 mA, nominal supplies, $T_A$ = 25°C, unless otherwise noted.

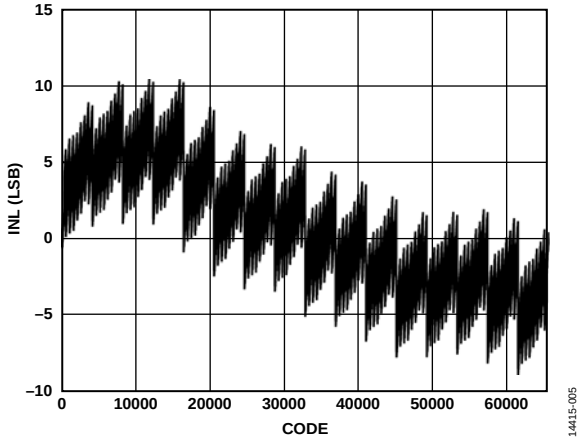


Figure 5. INL, $I_{OUTFS}$ = 20 mA

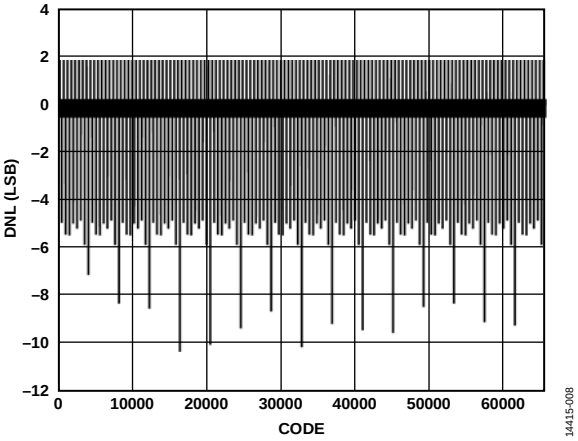


Figure 8. DNL, $I_{OUTFS}$ = 20 mA

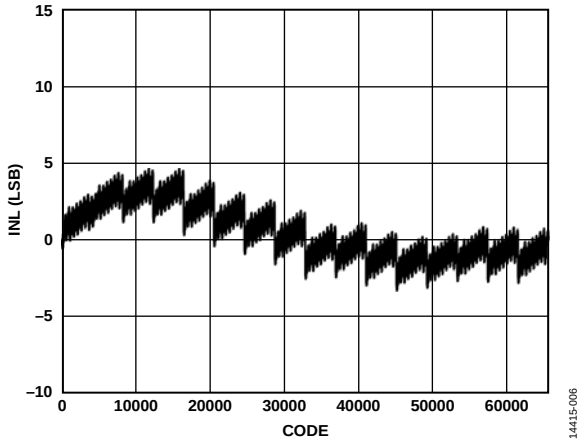


Figure 6. INL, $I_{OUTFS}$ = 30 mA

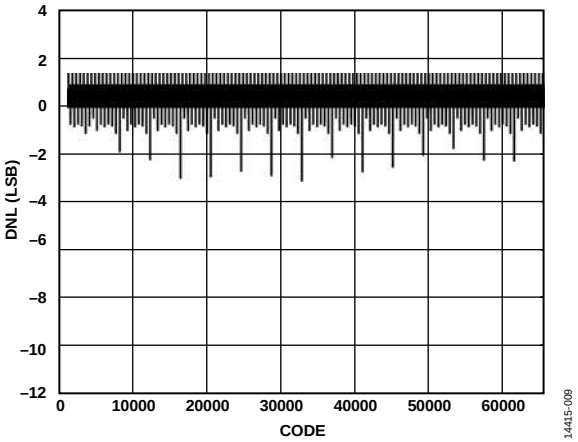


Figure 9. DNL, $I_{OUTFS}$ = 30 mA

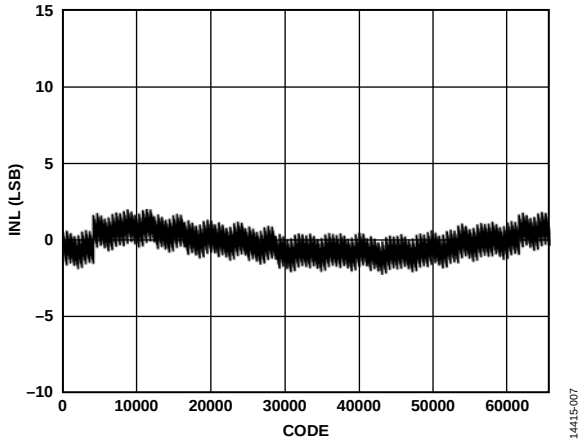


Figure 7. INL, $I_{OUTFS}$ = 40 mA

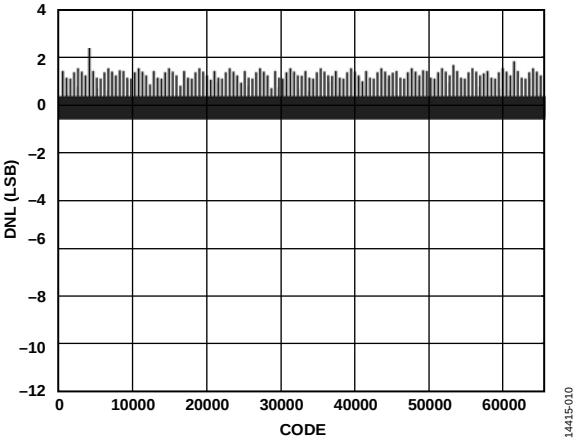


Figure 10. DNL, $I_{OUTFS}$ = 40 mA

## AC PERFORMANCE (NRZ MODE)

$I_{OUTFS}$ = 40 mA, $f_{DAC}$ = 5.0 GSPS, nominal supplies, $T_A$ = 25°C, unless otherwise noted.

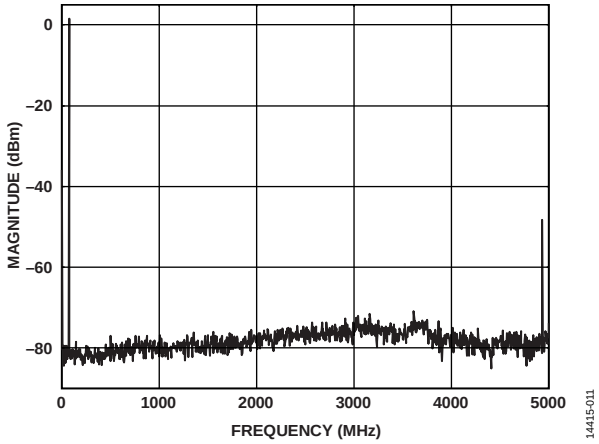


Figure 11. Single-Tone Spectrum at $f_{OUT}$ = 70 MHz

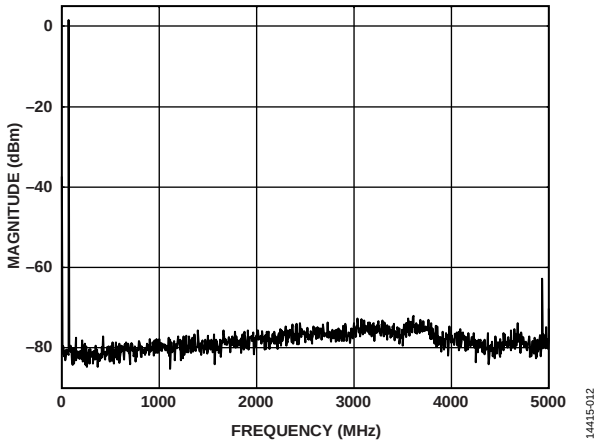


Figure 12. Single-Tone Spectrum at $f_{OUT}$ = 70 MHz (FIR85 Enabled)

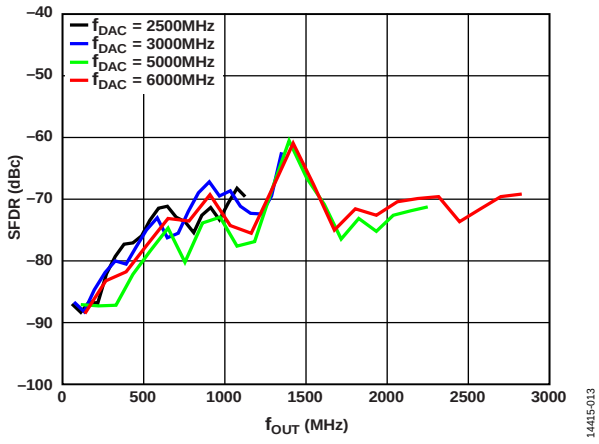


Figure 13. SFDR vs. $f_{OUT}$ over $f_{DAC}$

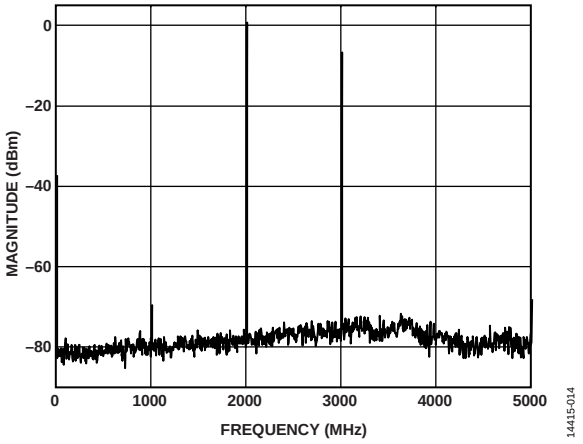


Figure 14. Single-Tone Spectrum at $f_{OUT}$ = 2000 MHz

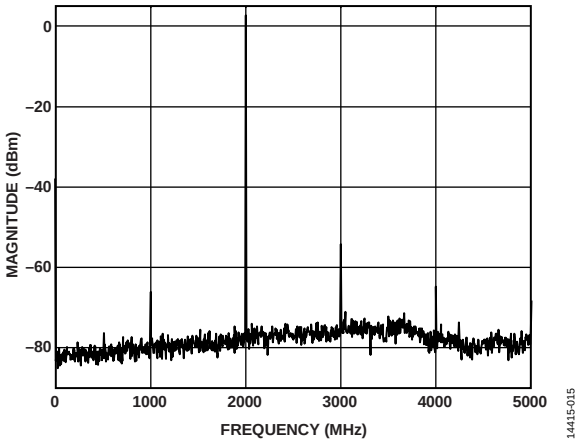


Figure 15. Single-Tone Spectrum at $f_{OUT}$ = 2000 MHz (FIR85 Enabled)

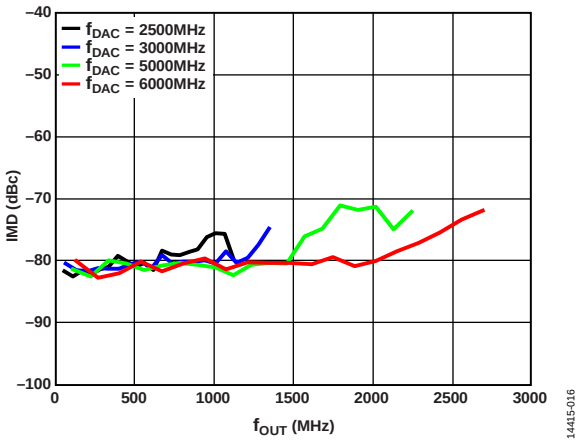


Figure 16. IMD vs. $f_{OUT}$ over $f_{DAC}$

$I_{OUTFS}$ = 40 mA, $f_{DAC}$ = 5.0 GSPS, nominal supplies, $T_A$ = 25°C, unless otherwise noted.

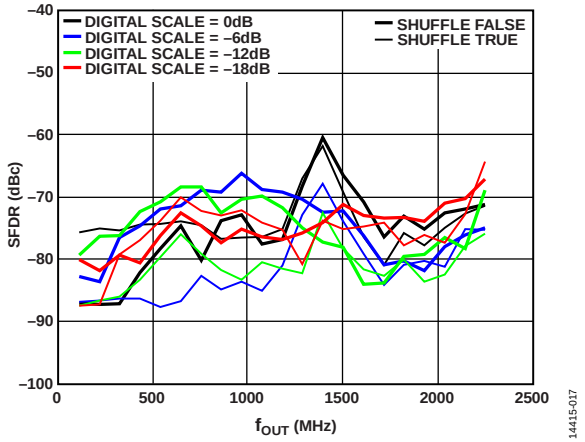


Figure 17. SFDR vs. $f_{OUT}$ over Digital Scale

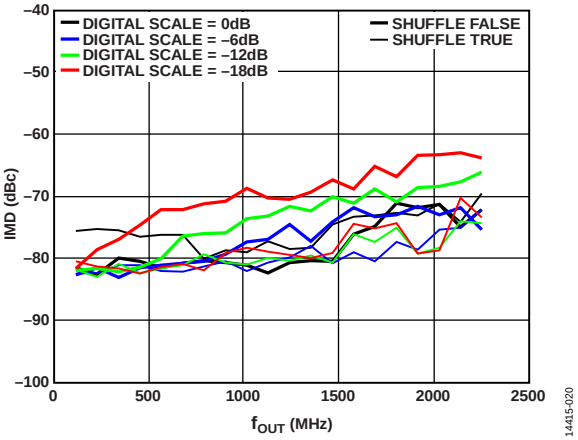


Figure 20. IMD vs. $f_{OUT}$ over Digital Scale

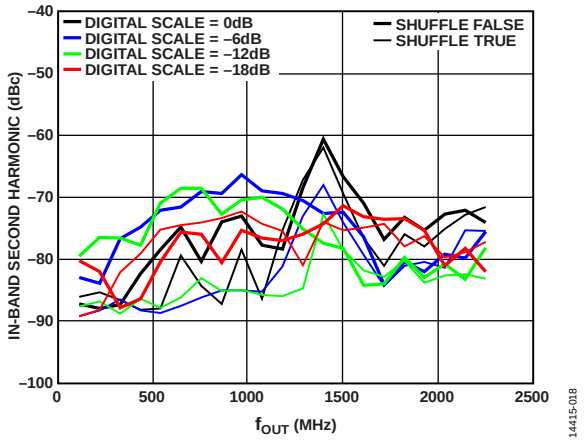


Figure 18. SFDR for In-Band Second Harmonic vs. $f_{OUT}$ over Digital Scale

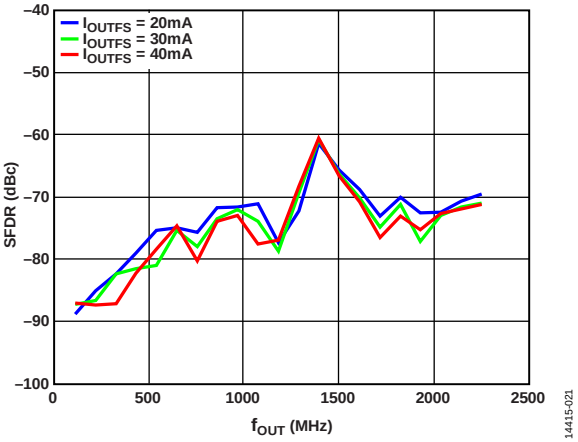


Figure 21. SFDR vs. $f_{OUT}$ over DAC $I_{OUTFS}$

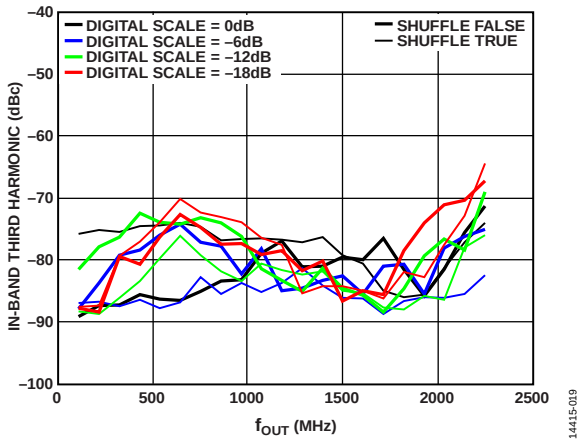


Figure 19. SFDR for In-Band Third Harmonic vs. $f_{OUT}$ over Digital Scale

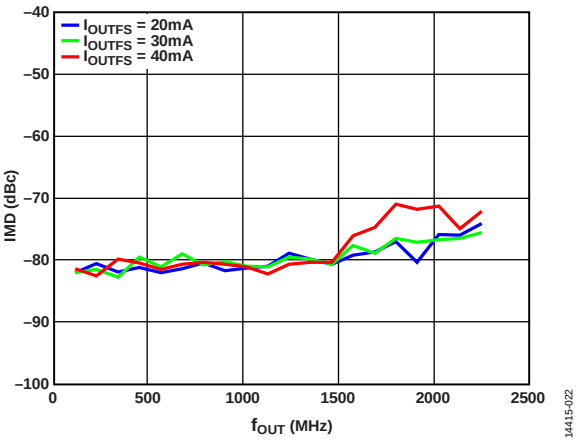


Figure 22. IMD vs. $f_{OUT}$ over DAC $I_{OUTFS}$

$I_{OUTFS}$ = 40 mA, $f_{DAC}$ = 5.0 GSPS, nominal supplies, $T_A$ = 25°C, unless otherwise noted.

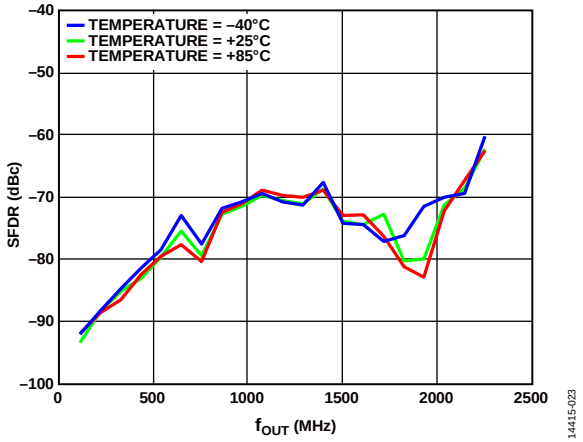


Figure 23. SFDR vs. $f_{OUT}$ over Temperature

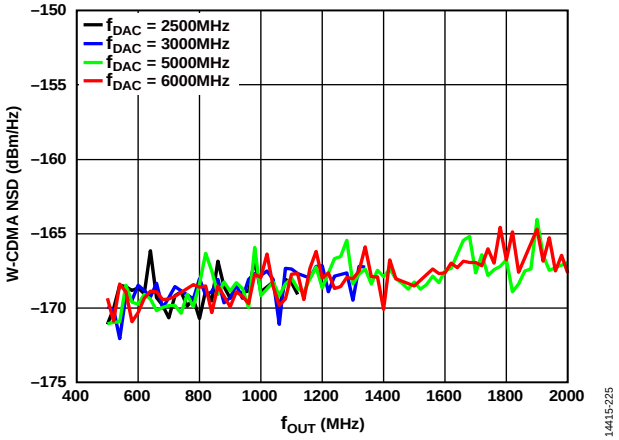


Figure 26. W-CDMA NSD Measured at 70 MHz vs. $f_{OUT}$ over $f_{DAC}$

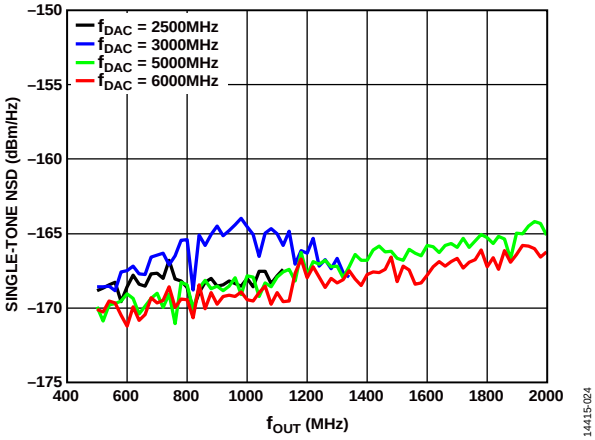


Figure 24. Single-Tone NSD Measured at 70 MHz vs. $f_{OUT}$ over $f_{DAC}$

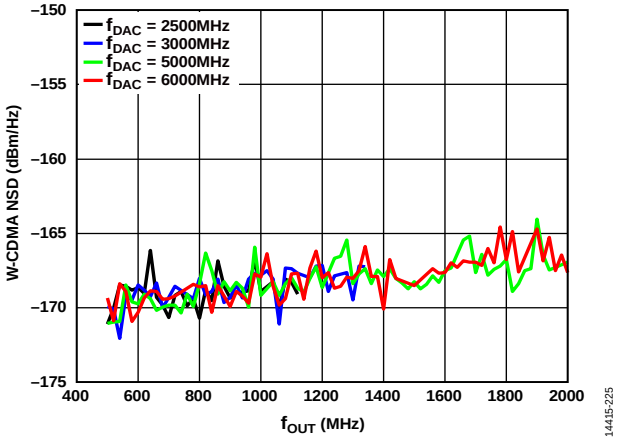


Figure 27. W-CDMA NSD Measured at 10% Offset from $f_{OUT}$ vs. $f_{OUT}$ over $f_{DAC}$

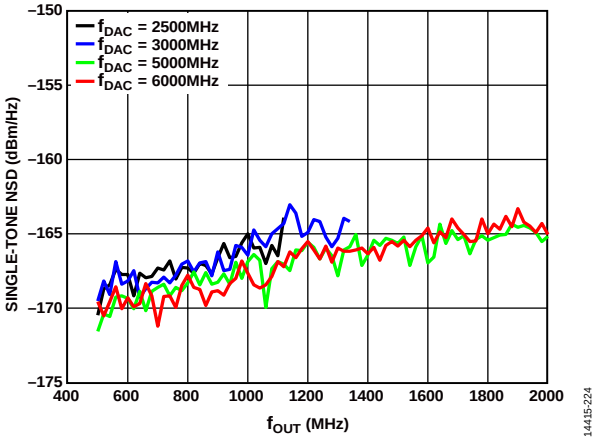


Figure 25. Single-Tone NSD Measured at 10% Offset from $f_{OUT}$ vs. $f_{OUT}$ over $f_{DAC}$

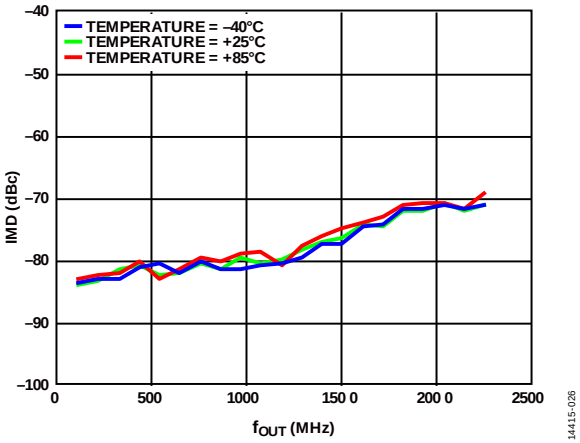


Figure 28. IMD vs. $f_{OUT}$ over Temperature

$I_{OUTFS}$ = 40 mA, $f_{DAC}$ = 5.0 GSPS, nominal supplies, $T_A$ = 25°C, unless otherwise noted.

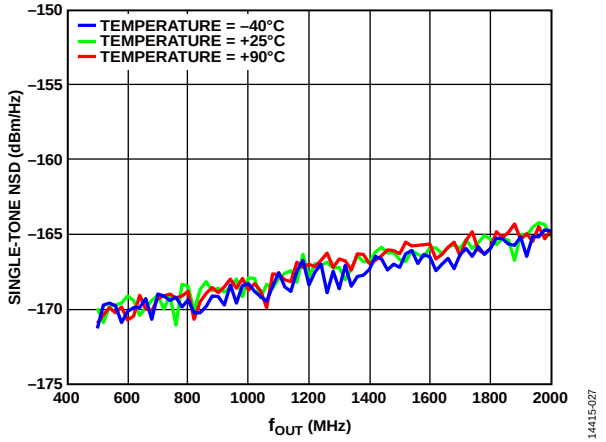


Figure 29. Single-Tone NSD Measured at 70 MHz vs. $f_{OUT}$ over Temperature

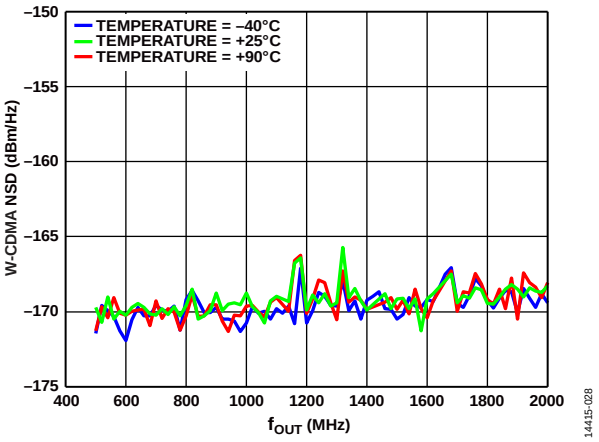


Figure 32. W-CDMA NSD Measured at 70 MHz vs. $f_{OUT}$ over Temperature

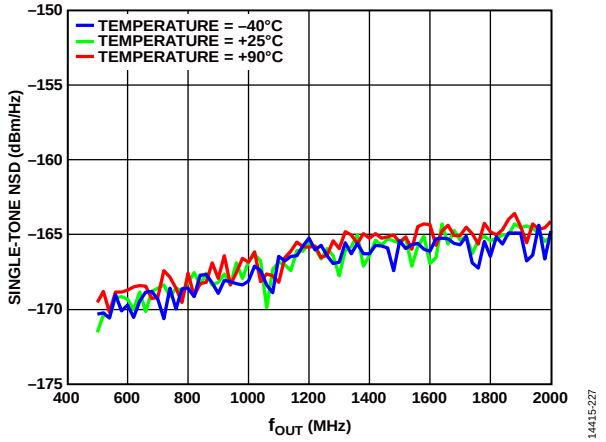


Figure 30. Single-Tone NSD Measured at 10% Offset from $f_{OUT}$ vs. $f_{OUT}$ over Temperature

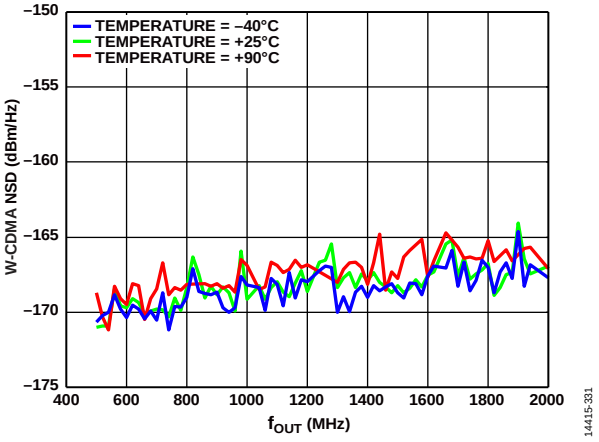


Figure 33. W-CDMA NSD Measured at 10% Offset from $f_{OUT}$ vs. $f_{OUT}$ over Temperature

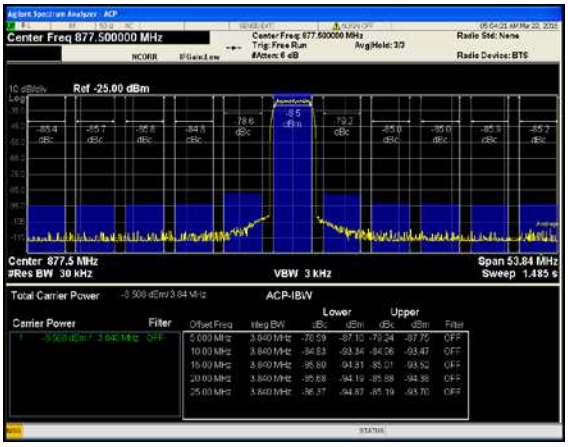


Figure 31. Single-Carrier W-CDMA at 877.5 MHz

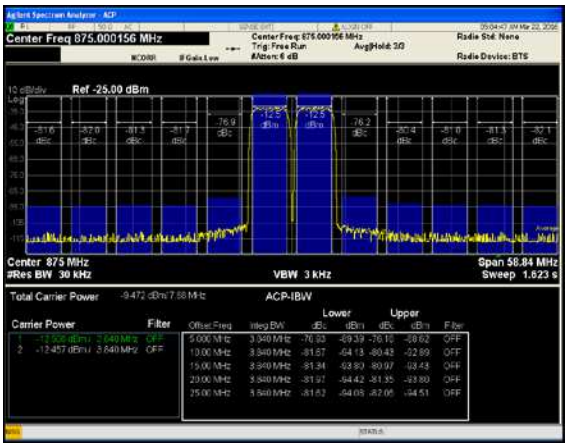


Figure 34. Two-Carrier W-CDMA at 875 MHz

$I_{OUTFS}$ = 40 mA, $f_{DAC}$ = 5.0 GSPS, nominal supplies, $T_A$ = 25°C, unless otherwise noted.

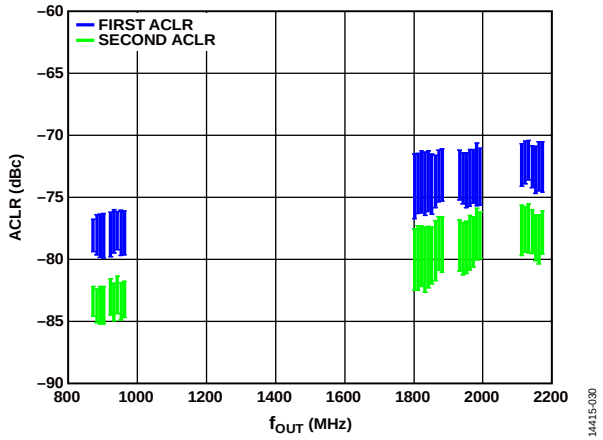


Figure 35. Single-Carrier, W-CDMA Adjacent Channel Leakage Ratio (ACLR) vs. $f_{OUT}$ (First ACLR, Second ACLR)

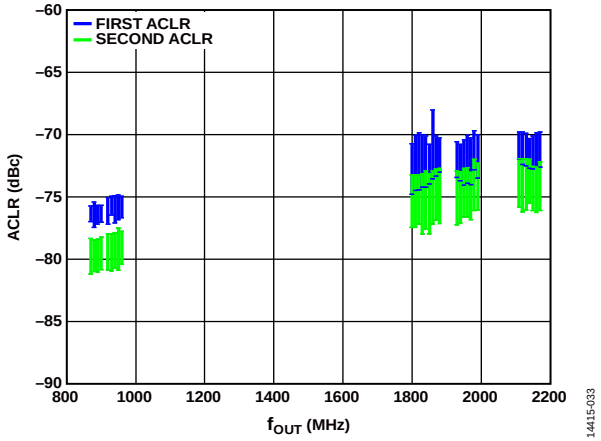


Figure 38. Two-Carrier, W-CDMA ACLR vs. $f_{OUT}$ (First ACLR, Second ACLR)

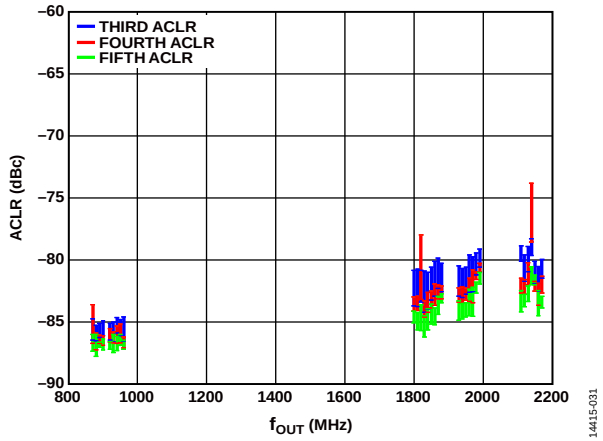


Figure 36. Single-Carrier, W-CDMA ACLR vs. $f_{OUT}$ (Third ACLR, Fourth ACLR, Fifth ACLR)

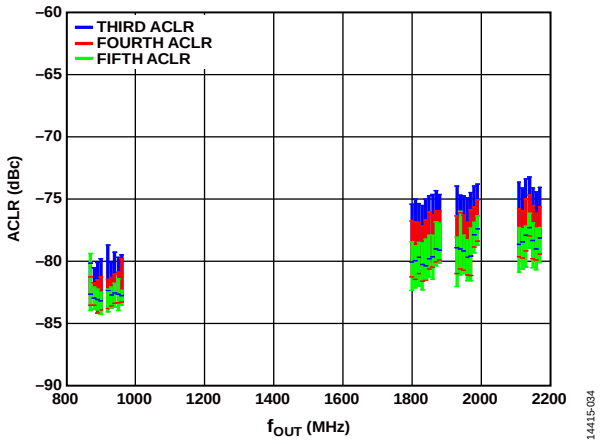


Figure 39. Two-Carrier, W-CDMA ACLR vs. $f_{OUT}$ (Third ACLR, Fourth ACLR, Fifth ACLR)

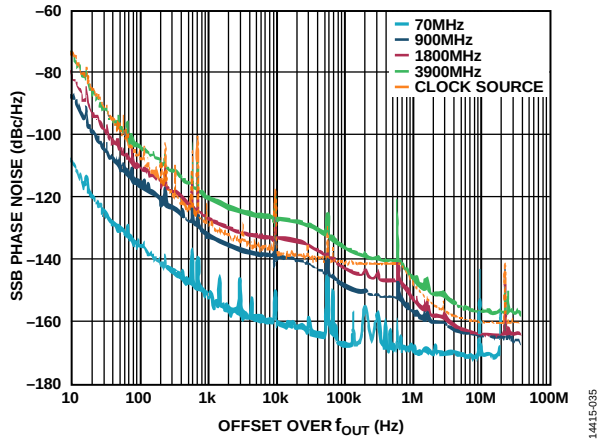


Figure 37. SSB Phase Noise vs. Offset over $f_{OUT}$, $f_{DAC}$ = 4000 MSPS (Two Different DAC Clock Sources Used for Best Composite Curve)

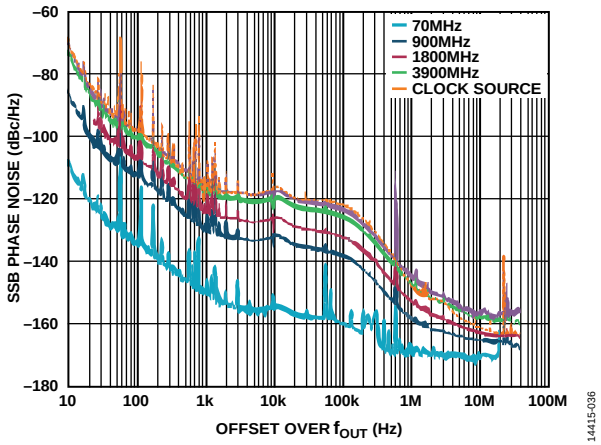


Figure 40. SSB Phase Noise vs. Offset over $f_{OUT}$, $f_{DAC}$ = 6000 MSPS

## AC (MIX-MODE)

$I_{OUTFS}$ = 40 mA, $f_{DAC}$ = 5.0 GSPS, nominal supplies, $T_A$ = 25°C, unless otherwise noted.

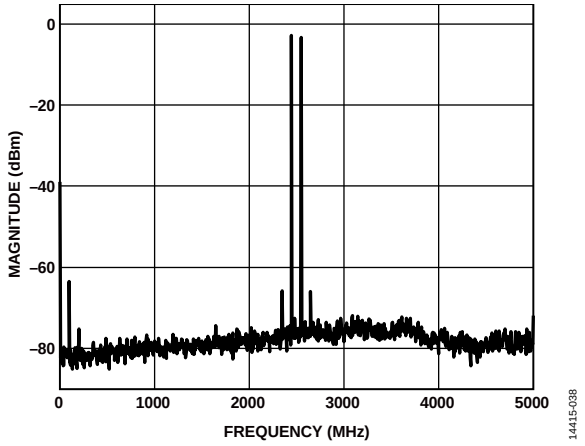


Figure 41. Single-Tone Spectrum at $f_{OUT}$ = 2350 MHz

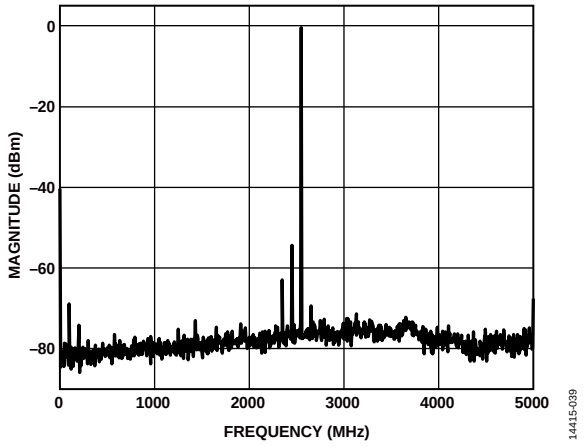


Figure 42. Single-Tone Spectrum at $f_{OUT}$ = 2350 MHz (FIR85 Enabled)

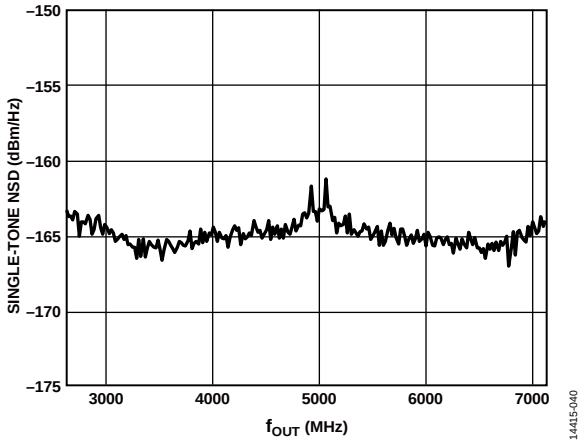


Figure 43. Single-Tone NSD vs. $f_{OUT}$

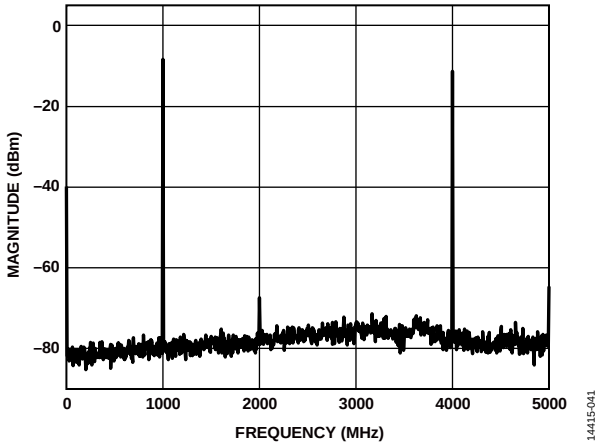


Figure 44. Single-Tone Spectrum at $f_{OUT}$ = 4000 MHz

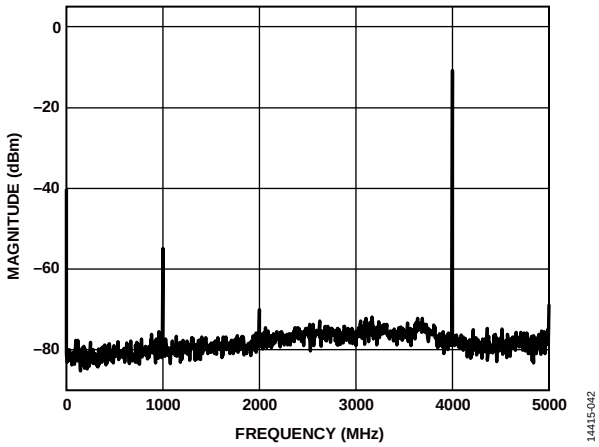


Figure 45. Single-Tone Spectrum at $f_{OUT}$ = 4000 MHz (FIR85 Enabled)

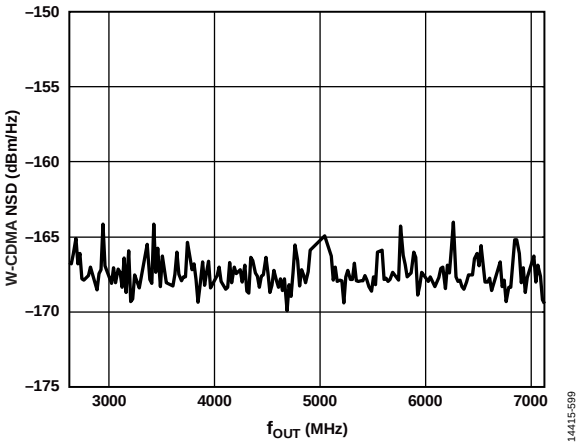


Figure 46. W-CDMA NSD vs. $f_{OUT}$

$I_{OUTFS}$ = 40 mA, $f_{DAC}$ = 5.0 GSPS, nominal supplies, $T_A$ = 25°C, unless otherwise noted.

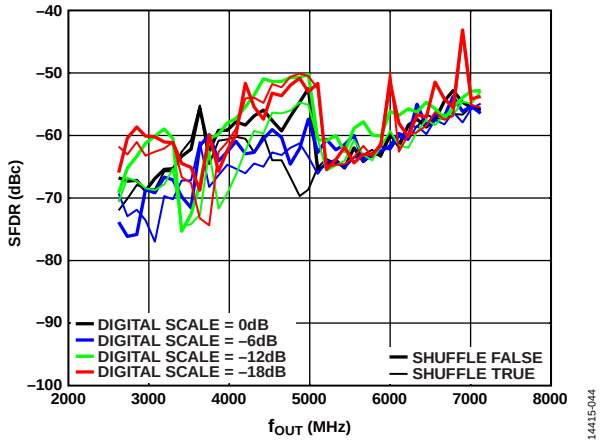


Figure 47. SFDR vs. $f_{OUT}$ over Digital Scale

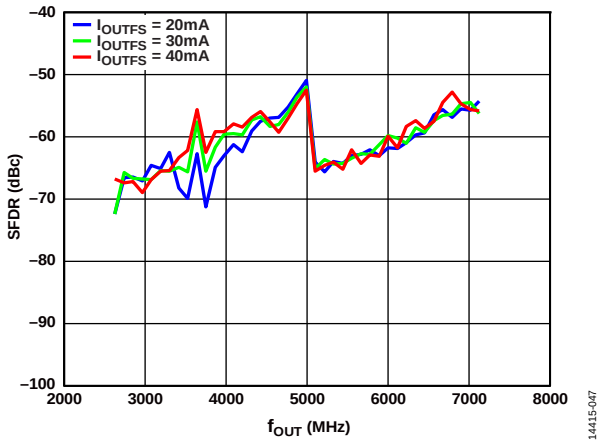


Figure 50. SFDR vs. $f_{OUT}$ over DAC $I_{OUTFS}$

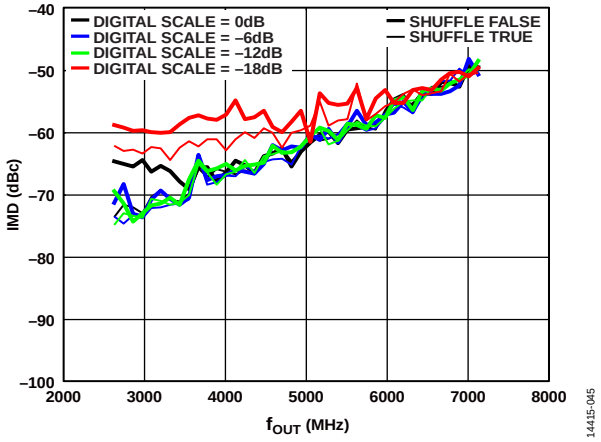


Figure 48. IMD vs. $f_{OUT}$ over Digital Scale

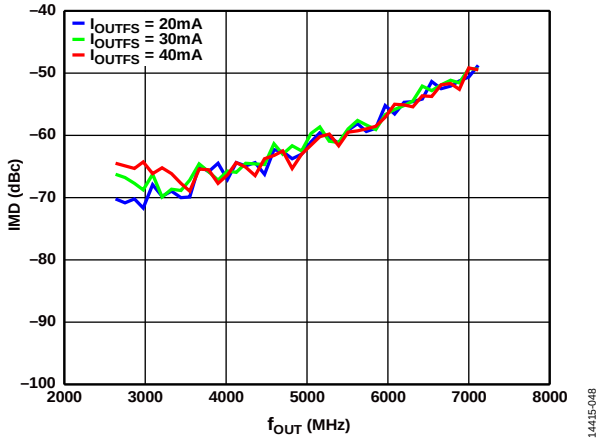


Figure 51. IMD vs. $f_{OUT}$ over DAC $I_{OUTFS}$

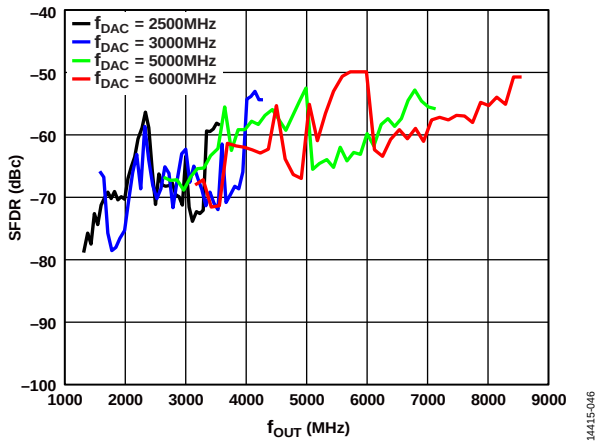


Figure 49. SFDR vs. $f_{OUT}$ over $f_{DAC}$

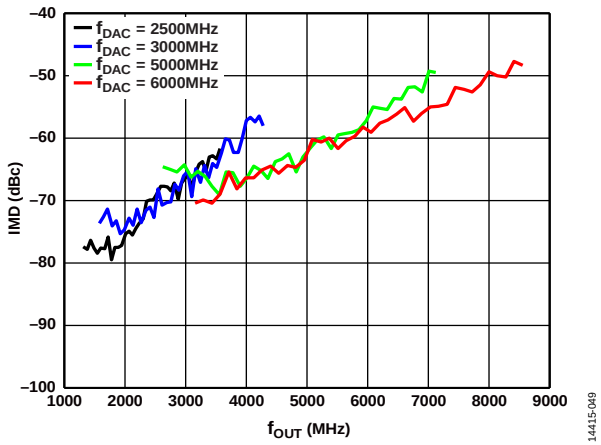


Figure 52. IMD vs. $f_{OUT}$ over $f_{DAC}$

$I_{OUTFS}$ = 40 mA, $f_{DAC}$ = 5.0 GSPS, nominal supplies, $T_A$ = 25°C, unless otherwise noted.

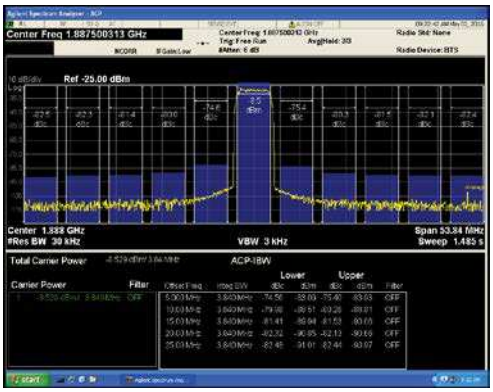


Figure 53. Single-Carrier W-CDMA at 1887.5 MHz

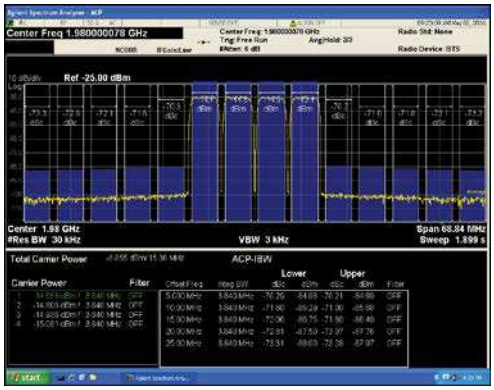


Figure 56. Four-Carrier W-CDMA at 1980 MHz

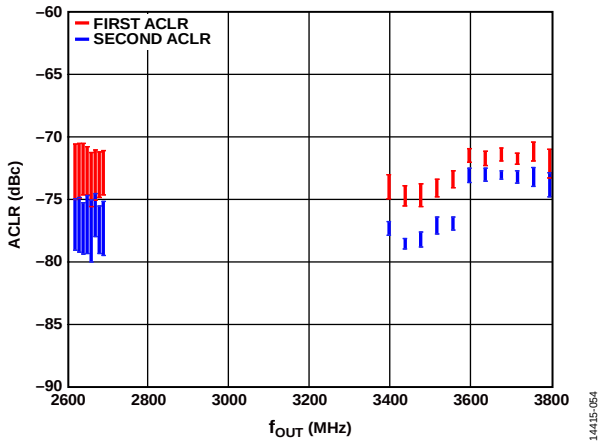


Figure 54. Single-Carrier, W-CDMA ACLR vs. $f_{OUT}$ (First ACLR, Second ACLR)

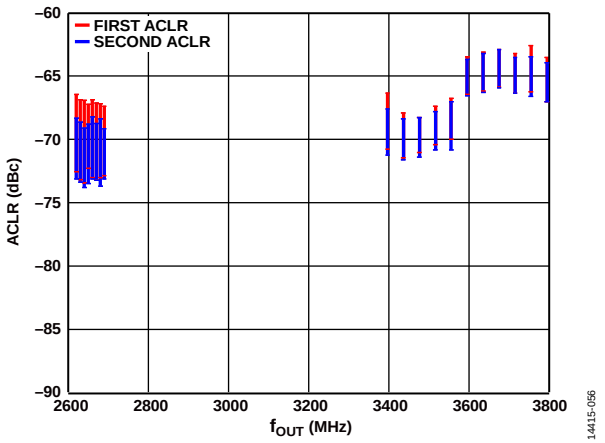


Figure 57. Four-Carrier, W-CDMA ACLR vs. $f_{OUT}$ (First ACLR, Second ACLR)

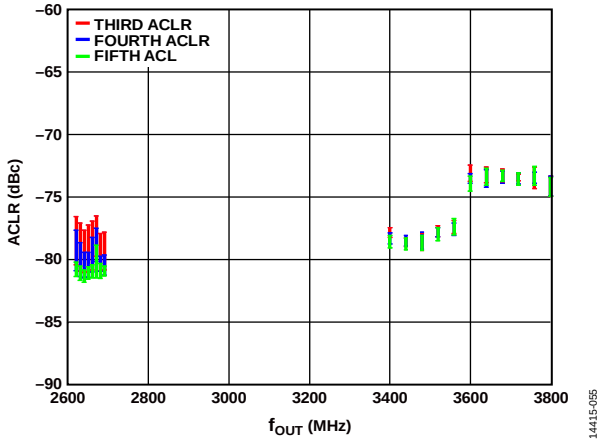


Figure 55. Single-Carrier, W-CDMA ACLR vs. $f_{OUT}$ (Third ACLR, Fourth ACLR, Fifth ACLR)

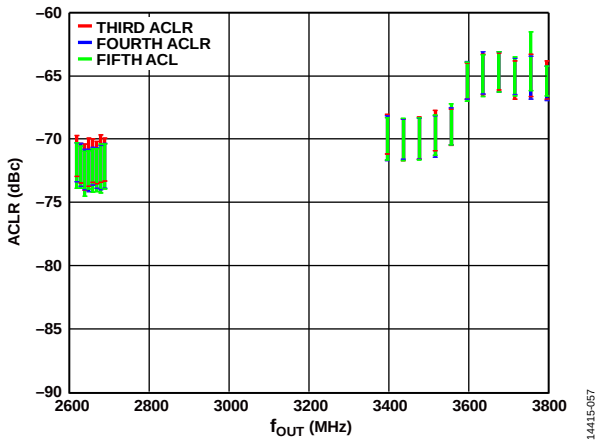


Figure 58. Four-Carrier, W-CDMA ACLR vs. $f_{OUT}$ (Third ACLR, Fourth ACLR, Fifth ACLR)

## DOCSIS PERFORMANCE (NRZ MODE)

$I_{OUTFS}$ = 40 mA, $f_{DAC}$ = 3.076 GSPS, nominal supplies, FIR85 enabled, $T_A$ = 25°C, unless otherwise noted.

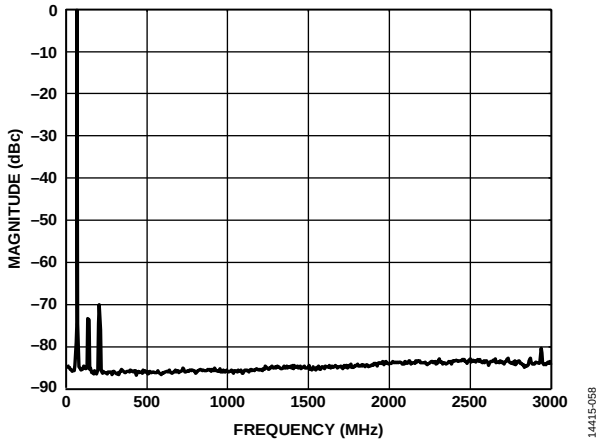


Figure 59. Single Carrier at 70 MHz Output

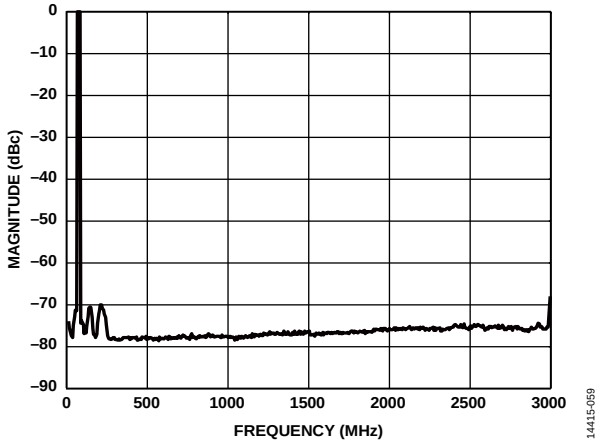


Figure 60. Four Carriers at 70 MHz Output

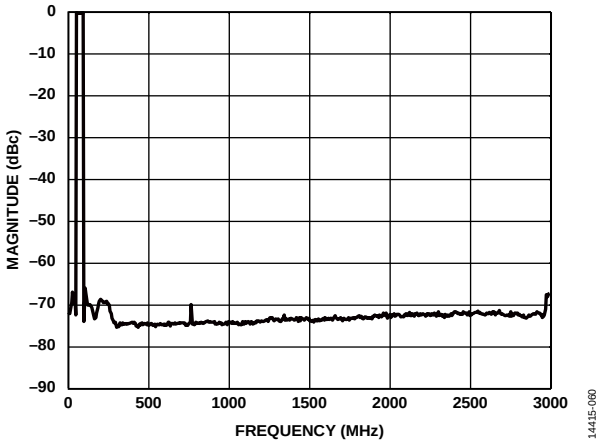


Figure 61. Eight Carriers at 70 MHz Output

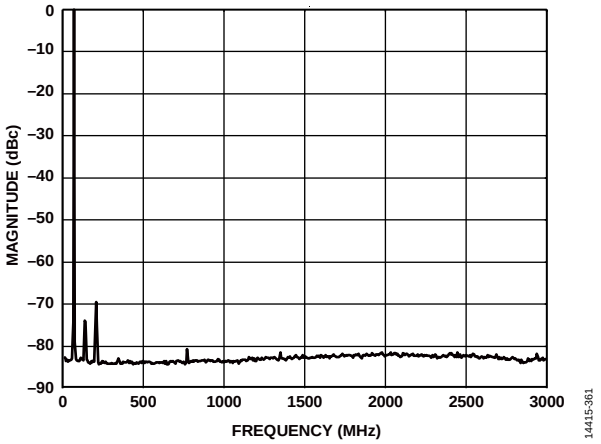


Figure 62. Single Carrier at 70 MHz Output (Shuffle On)

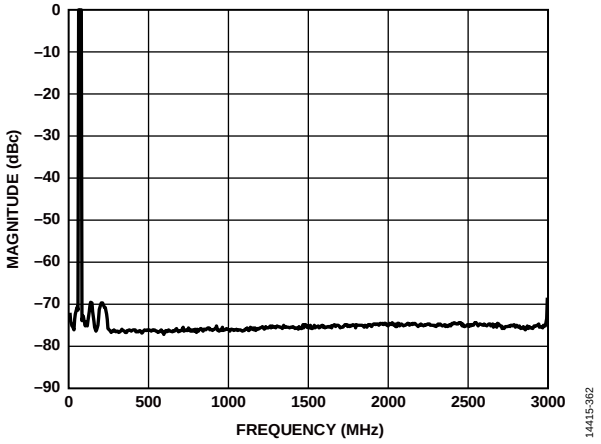


Figure 63. Four Carriers at 70 MHz Output (Shuffle On)

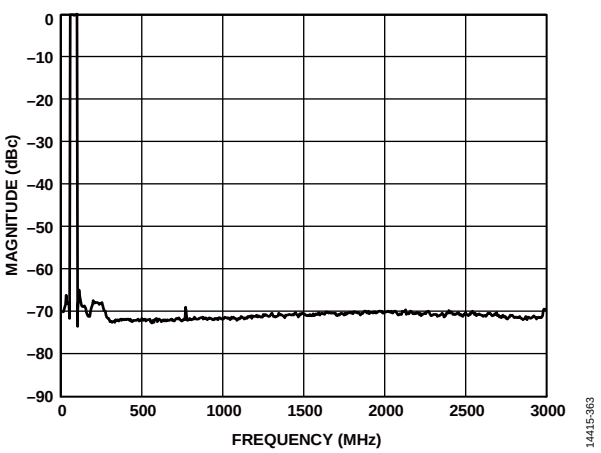


Figure 64. Eight Carriers at 70 MHz Output (Shuffle On)

$I_{OUTFS}$ = 40 mA, $f_{DAC}$ = 3.076 GSPS, nominal supplies, FIR85 enabled, $T_A$ = 25°C, unless otherwise noted.

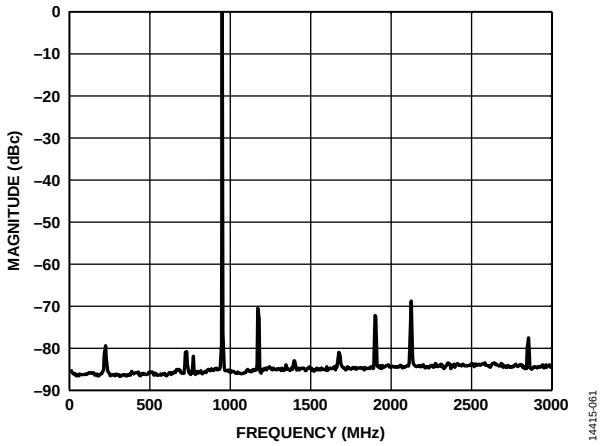


Figure 65. Single Carrier at 950 MHz Output

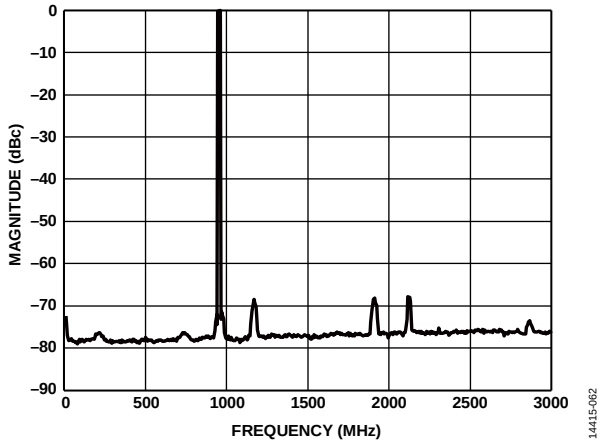


Figure 66. Four Carriers at 950 MHz Output

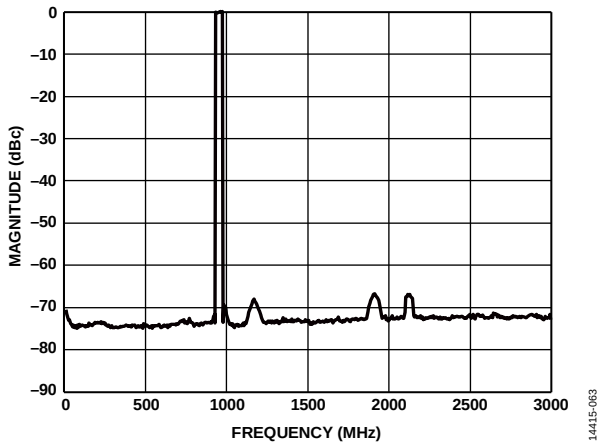


Figure 67. Eight Carriers at 950 MHz Output

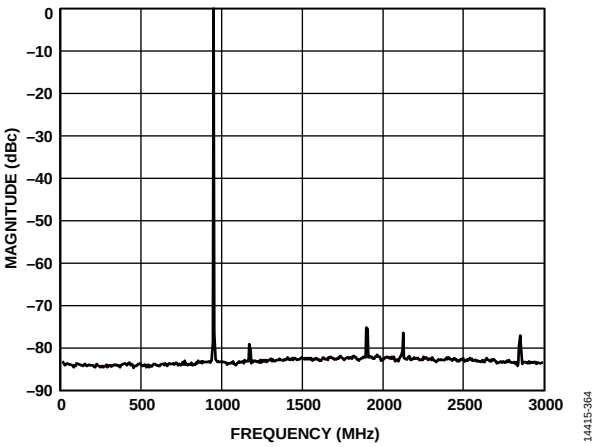


Figure 68. Single Carrier at 950 MHz Output (Shuffle On)

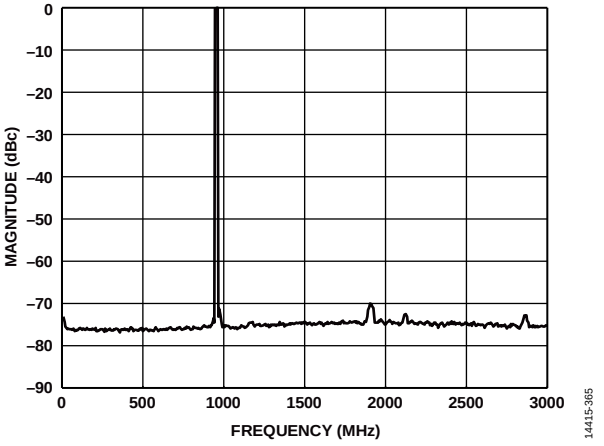


Figure 69. Four Carriers at 950 MHz Output (Shuffle On)

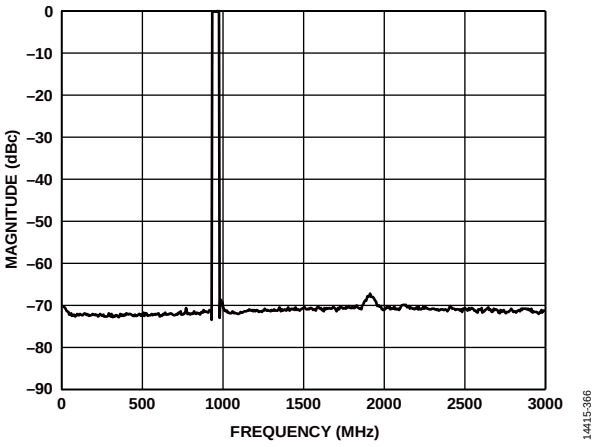


Figure 70. Eight Carriers at 950 MHz Output (Shuffle On)

$I_{OUTFS}$ = 40 mA, $f_{DAC}$ = 3.076 GSPS, nominal supplies, FIR85 enabled, $T_A$ = 25°C, unless otherwise noted.

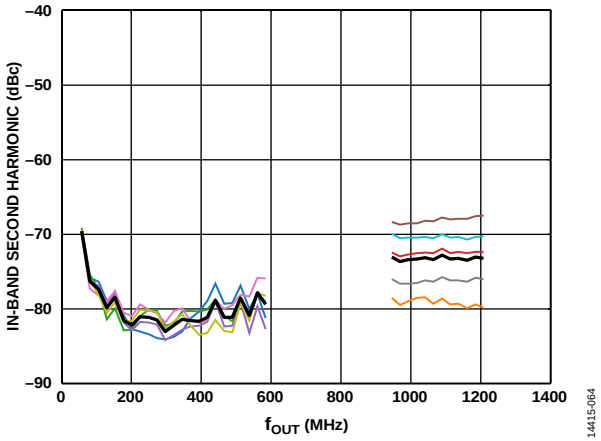


Figure 71. In-Band Second Harmonic vs. $f_{OUT}$ Performance for One DOCSIS Carrier

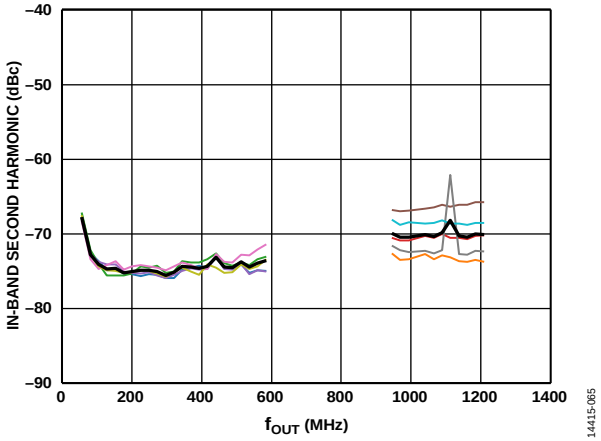


Figure 72. In-Band Second Harmonic vs. $f_{OUT}$ Performance for Four DOCSIS Carriers

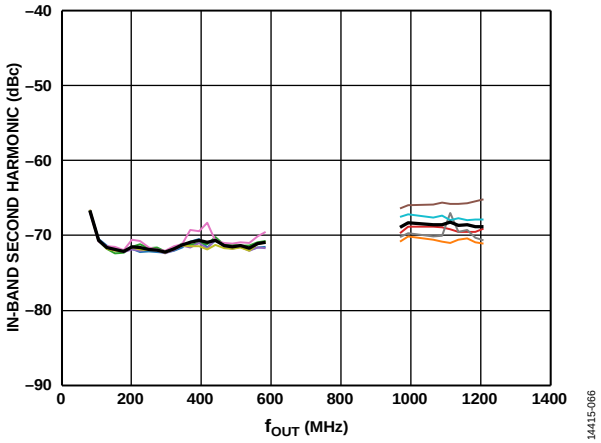


Figure 73. In-Band Second Harmonic vs. $f_{OUT}$ Performance for Eight DOCSIS Carriers

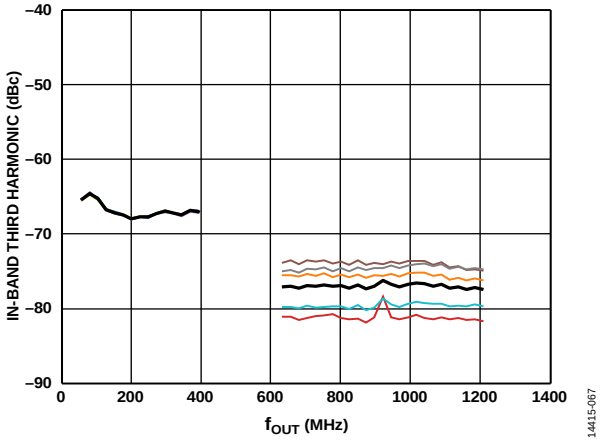


Figure 74. In-Band Third Harmonic vs. $f_{OUT}$ Performance for One DOCSIS Carrier

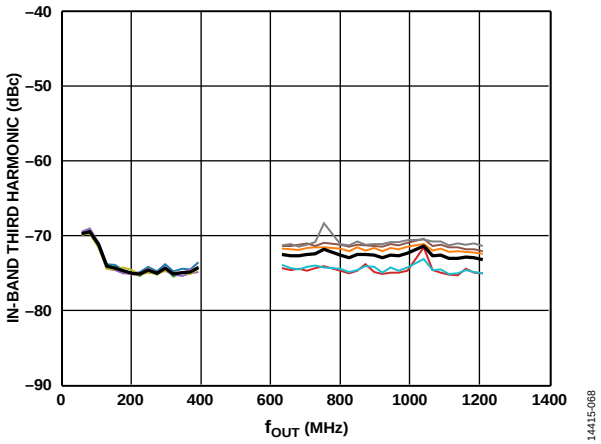


Figure 75. In-Band Third Harmonic vs. $f_{OUT}$ Performance for Four DOCSIS Carriers

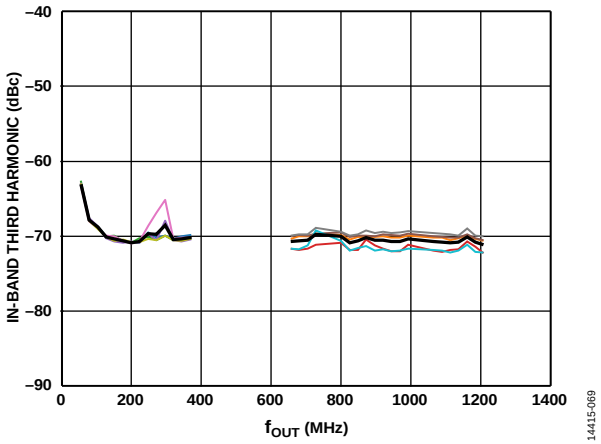


Figure 76. In-Band Third Harmonic vs. $f_{OUT}$ Performance for Eight DOCSIS Carriers

$I_{OUTFS}$ = 40 mA, $f_{DAC}$ = 3.076 GSPS, nominal supplies, FIR85 enabled, $T_A$ = 25°C, unless otherwise noted.

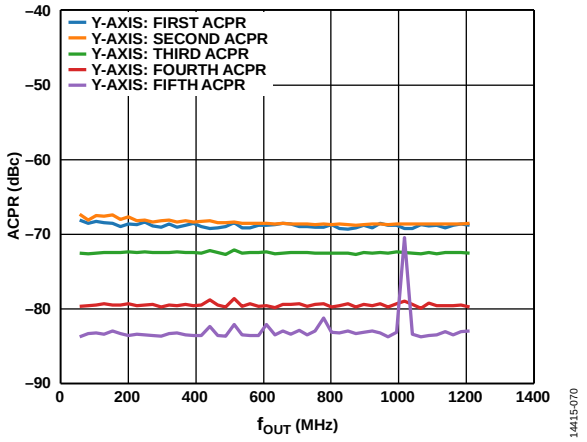


Figure 77. Single-Carrier Adjacent Channel Power Ratio (ACPR) vs. $f_{OUT}$

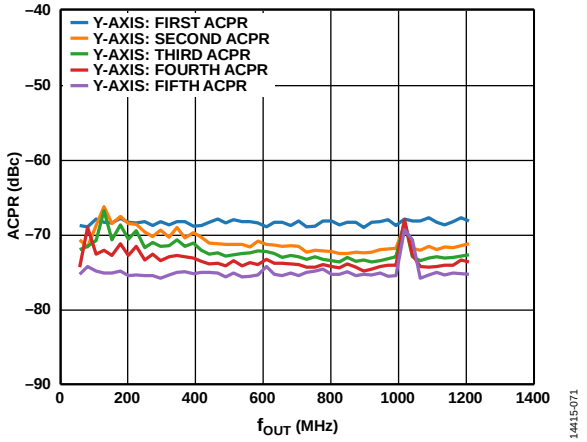


Figure 78. Four-Carrier ACPR vs. $f_{OUT}$

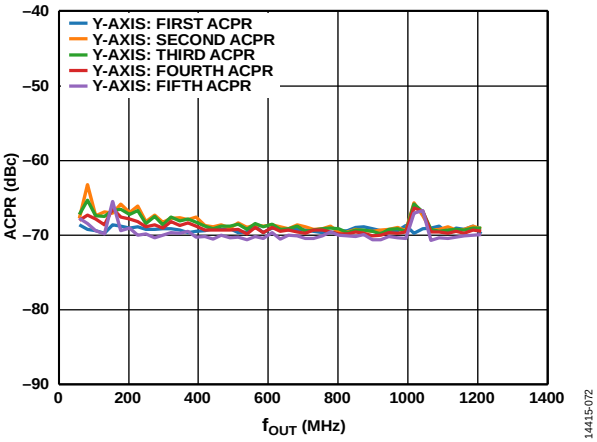


Figure 79. Eight-Carrier ACPR vs. $f_{OUT}$

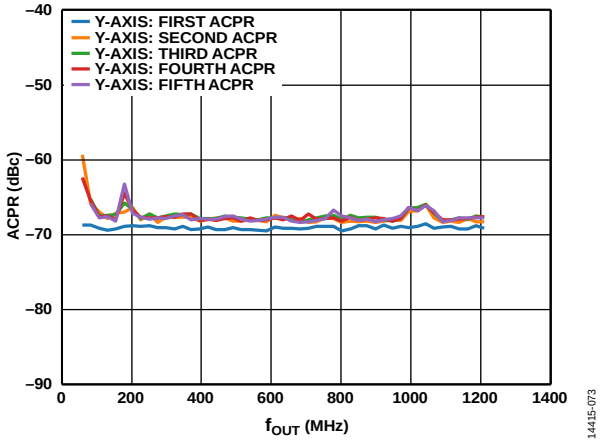


Figure 80. 16-Carrier ACPR vs. $f_{OUT}$

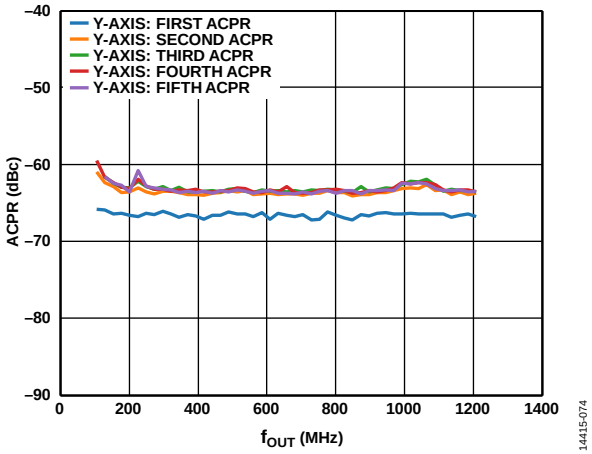


Figure 81. 32-Carrier ACPR vs. $f_{OUT}$

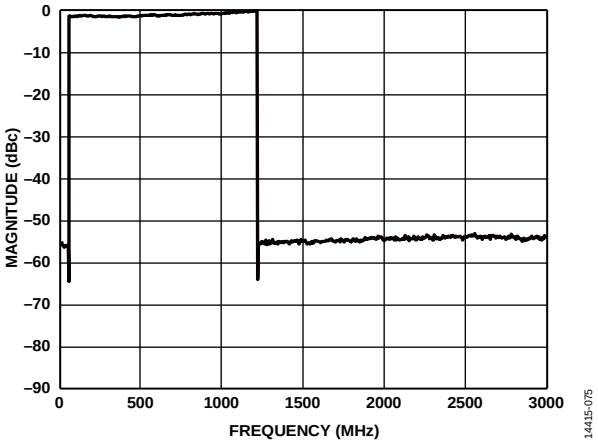


Figure 82. 194-Carrier, Sinc Enabled, FIR85 Enabled

$I_{OUTFS}$ = 40 mA, $f_{DAC}$ = 3.076 GSPS, nominal supplies, FIR85 enabled, $T_A$ = 25°C, unless otherwise noted.

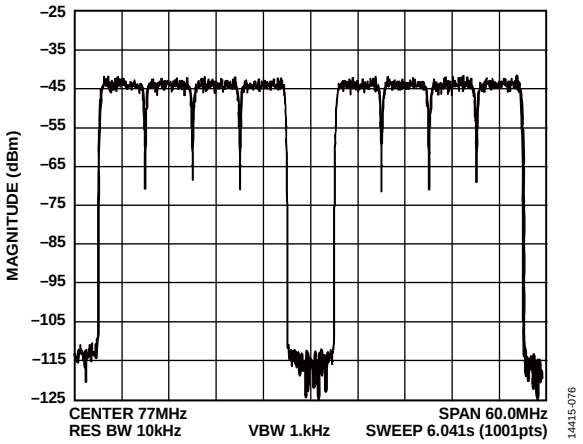


*Figure 83. Gap Channel ACLR at 77 MHz*

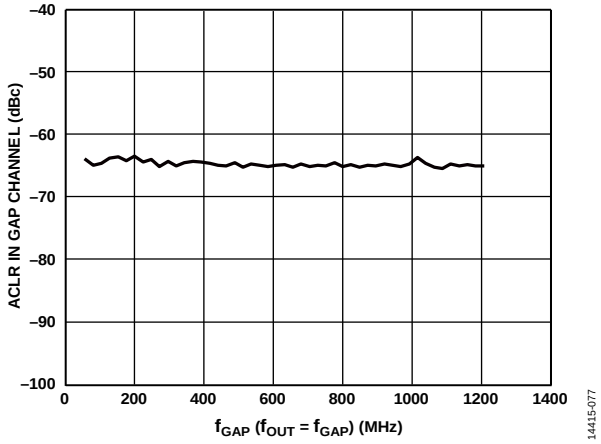


*Figure 84. ACLR in Gap Channel vs. $f_{GAP}$*

# 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 OUTPUT+, 0 mA output is expected when all inputs are set to 0. For OUTPUT−, 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 the input is at its minimum code and the output when the input is at its maximum code.

**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.

**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, energy in this band is rejected by the interpolation filters. 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 the interpolation rate ($f_{DATA}$), a digital filter can be constructed that has a sharp transition band near $f_{DATA}/2$. Images that typically appear around the output data rate ($f_{DAC}$) 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.

**Adjusted DAC Update Rate**
The adjusted DAC update rate is the DAC update rate divided by the smallest interpolating factor. For clarity on DACs with multiple interpolating factors, the adjusted DAC update rate for each interpolating factor may be given.

**Physical Lane**
Physical Lane x refers to SERDINx±.

**Logical Lane**
Logical Lane x refers to physical lanes after optionally being remapped by the crossbar block (Register 0x308 to Register 0x30B).

**Link Lane**
Link Lane x refers to logical lanes considered in the link.

# THEORY OF OPERATION

The AD9163 is a 16-bit single RF DAC and digital upconverter with a SERDES interface. Figure 1 shows a detailed functional block diagram of the AD9163. Eight high speed serial lanes carry data at a maximum speed of 12.5 Gbps, and either a 5 GSPS real input or a 2.5 GSPS complex input data rate to the DAC. Compared to either LVDS or CMOS interfaces, the SERDES interface simplifies pin count, board layout, and input clock requirements to the device.

The clock for the input data is derived from the DAC clock, or device clock (required by the JESD204B specification). This device clock is sourced with a high fidelity direct external DAC sampling clock. The performance of the DAC can be optimized by using on-chip adjustments to the clock input, accessible through the SPI port. The device can be configured to operate in one-lane, two-lane, three-lane, four-lane, six-lane, or eight-lane modes, depending on the required input data rate.

The digital datapath of the AD9163 offers several interpolation modes (6×, 8×, 12×, 16×, and 24×) through either an initial half-band (2×) or third-band (3×) filter with programmable 80% or 90% bandwidth, and three subsequent half-band filters (all 90%) with a maximum DAC sample rate of 6 GSPS. An inverse sinc filter is provided to compensate for sinc related roll-off. An additional half-band filter, FIR85, takes advantage of the quad-switch architecture to interpolate on the falling edge of the clock, and effectively double the DAC update rate in 2× NRZ mode. A 48-bit programmable modulus NCO is provided to enable digital frequency shifts of signals with near infinite precision. The NCO can be operated with digital data from the SERDES interface and digital datapath. The 100 MHz speed of the SPI write interface enables rapid updating of the frequency tuning word of the NCO.

The AD9163 DAC core provides a fully differential current output with a nominal full-scale current of 38.76 mA. The full-scale output current, $I_{OUTFS}$, is user adjustable from 8 mA to 38.76 mA, typically. The differential current outputs are complementary. The DAC uses the patented quad-switch architecture, which enables DAC decoder options to extend the output frequency range into the second and third Nyquist zones with Mix-Mode, return to zero (RZ) mode, and 2× NRZ mode (with FIR85 enabled). Mix-Mode can be used to access 1.5 GHz to around 7.5 GHz. In the interpolation modes, the output can range from 0 Hz to 6 GHz in 2× NRZ mode using the NCO to shift a signal of up to 1.8 GHz instantaneous bandwidth to the desired $f_{OUT}$.

The AD9163 is capable of multichip synchronization that can both synchronize multiple DACs and establish a constant and deterministic latency (latency locking) path for the DACs. The latency for each of the DACs remains constant to within several DAC clock cycles from link establishment to link establishment. An external alignment (SYSREF±) signal makes the AD9163 Subclass 1 compliant. Several modes of SYSREF± signal handling are available for use in the system.

An SPI configures the various functional blocks and monitors their statuses. The various functional blocks and the data interface must be set up in a specific sequence for proper operation (see the Start-Up Sequence section). Simple SPI initialization routines set up the JESD204B link and are included in the evaluation board package. This data sheet describes the various blocks of the AD9163 in greater detail. Descriptions of the JESD204B interface, control parameters, and various registers to set up and monitor the device are provided. The recommended start-up routine reliably sets up the data link.

# SERIAL PORT OPERATION

The serial port is a flexible, synchronous serial communications port that allows easy interfacing with many industry-standard microcontrollers and microprocessors. The serial input/output (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 AD9163. MSB first or LSB first transfer formats are supported. The serial port interface can be configured as a 4-wire interface or a 3-wire interface in which the input and output share a single-pin I/O (SDIO).

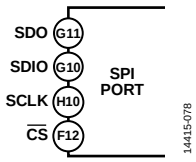


*Figure 85. Serial Port Interface Pins*

There are two phases to a communication cycle with the AD9163. 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 or more data bytes. Eight × N SCLK cycles are needed to transfer N bytes during the transfer cycle. Registers change immediately upon writing to the last bit of each transfer byte, except for the frequency tuning word (FTW) and numerically controlled oscillator (NCO) phase offsets, which change only when the frequency tuning word FTW_LOAD_REQ bit is set.

## DATA FORMAT

The instruction byte contains the information shown in Table 13.

**Table 13. 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 I14 to Bit I0 of the instruction word, determine the register that is accessed during the data transfer portion of the communication cycle. For multibyte transfers, A[14:0] is the starting address. The remaining register addresses are generated by the device based on the address increment bit. If the address increment bits are set high (Register 0x000, Bit 5 and Bit 2), multibyte SPI writes start on A[14:0] and increment by 1 every eight bits sent/received. If the address increment bits are set to 0, the address decrements by 1 every eight bits.

## 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 100 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 ($\overline{CS}$)*

An active low input starts and gates a communication cycle. $\overline{CS}$ allows more than one device to be used on the same serial communications lines. The SDIO pin goes to a high impedance state when this input is high. During the communication cycle, the chip select must stay low.

### *Serial Data I/O (SDIO)*

This pin is a bidirectional data line. In 4-wire mode, this pin acts as the data input and SDO acts as the data output.

## SERIAL PORT OPTIONS

The serial port can support both MSB first and LSB first data formats. This functionality is controlled by the LSB first bit (Register 0x000, Bit 6 and Bit 1). The default is MSB first (LSB bit = 0).

When the LSB first bits = 0 (MSB first), the instruction and data bits must be written from MSB to LSB. R/$\overline{W}$ is followed by A[14:0] as the instruction word, and D[7:0] is the data-word. When the LSB first bits = 1 (LSB first), the opposite is true. A[0:14] is followed by R/$\overline{W}$, which is subsequently followed by D[0:7].

The serial port supports a 3-wire or 4-wire interface. When the SDO active bits = 1 (Register 0x000, Bit 4 and Bit 3), a 4-wire interface with a separate input pin (SDIO) and output pin (SDO) is used. When the SDO active bits = 0, the SDO pin is unused and the SDIO pin is used for both the input and the output.

Multibyte data transfers can be performed as well by holding the $\overline{CS}$ pin low for multiple data transfer cycles (eight SCLKs) after the first data transfer word following the instruction cycle. The first eight SCLKs following the instruction cycle read from or write to the register provided in the instruction cycle. For each additional eight SCLK cycles, the address is either incremented or decremented and the read/write occurs on the new register.

The direction of the address can be set using ADDRINC or ADDRINC_M (Register 0x000, Bit 5 and Bit 2). When ADDRINC or ADDRINC_M is 1, the multicycle addresses are incremented. When ADDRINC or ADDRINC_M is 0, the addresses are decremented. A new write cycle can always be initiated by bringing $\overline{CS}$ high and then low again.

To prevent confusion and to ensure consistency between devices, the chip tests the first nibble following the address phase, ignoring the second nibble. This test is completed independently from the LSB first bits and ensures that there are extra clock cycles following the soft reset bits (Register 0x000, Bit 0 and Bit 7). This test of the first nibble only applies when writing to Register 0x000.

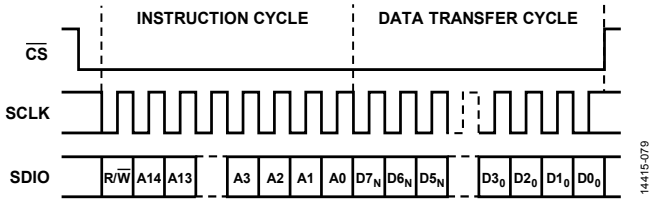


*Figure 86. Serial Register Interface Timing, MSB First, Register 0x000, Bit 5 and Bit 2 = 0*

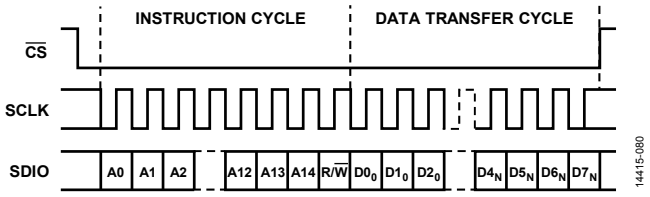


*Figure 87. Serial Register Interface Timing, LSB First, Register 0x000, Bit 5 and Bit 2 = 1*

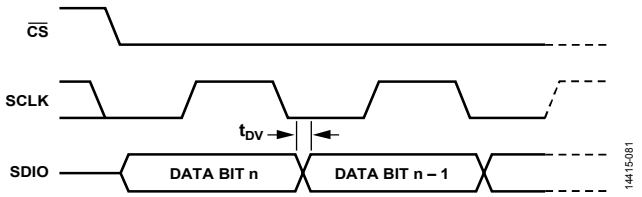


*Figure 88. Timing Diagram for Serial Port Register Read*

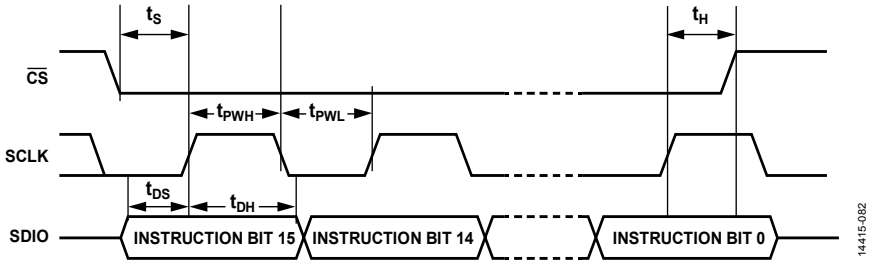


*Figure 89. Timing Diagram for Serial Port Register Write*

# JESD204B SERIAL DATA INTERFACE

## JESD204B OVERVIEW

The AD9163 has eight JESD204B data ports that receive data. The eight JESD204B ports can be configured as part of a single JESD204B link that uses a single system reference (SYSREF±) and device clock (CLK±).

The JESD204B serial interface hardware consists of three layers: the physical layer, the data link layer, and the transport layer. These sections of the hardware are described in subsequent sections, including information for configuring every aspect of the interface. Figure 90 shows the communication layers implemented in the AD9163 serial data interface to recover the clock and deserialize, descramble, and deframe the data before it is sent to the digital signal processing section of the device.

The physical layer establishes a reliable channel between the transmitter (Tx) and the receiver (Rx), the data link layer is responsible for unpacking the data into octets and descrambling the data. The transport layer receives the descrambled JESD204B frames and converts them to DAC samples.

A number of JESD204B parameters (L, F, K, M, N, NP, S, HD) define how the data is packed and tell the device how to turn the serial data into samples. These parameters are defined in detail in the Transport Layer section. The AD9163 also has a descrambling option (see the Descrambler section for more information).

The various combinations of JESD204B parameters that are supported depend solely on the number of lanes. Thus, a unique set of parameters can be determined by selecting the lane count to be used. In addition, the interpolation rate and number of lanes can be used to define the rest of the configuration needed to set up the AD9163. The interpolation rate and the number of lanes are selected in Register 0x110.

The AD9163 has a single DAC output; however, for the purposes of the complex signal processing on chip, the converter count is defined as M = 2 whenever interpolation is used.

For a particular application, the number of converters to use (M) and the DataRate variable are known. The LaneRate variable and number of lanes (L) can be traded off as follows:

$$DataRate = (DACRate)/(InterpolationFactor)$$
$$LaneRate = (20 \times DataRate \times M)/L$$

where *LaneRate* must be between 750 Mbps and 12.5 Gbps.

Achieving and recovering synchronization of the lanes is very important. To simplify the interface to the transmitter, the AD9163 designate a master synchronization signal for each JESD204B link. The SYNCOUT± pin is used as the master signal for all lanes. If any lane in a link loses synchronization, a resynchronization request is sent to the transmitter via the synchronization signal of the link. The transmitter stops sending data and instead sends synchronization characters to all lanes in that link until resynchronization is achieved.

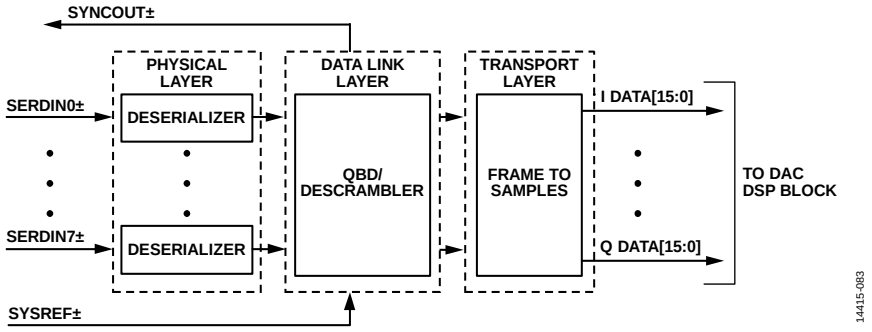


*Figure 90. Functional Block Diagram of Serial Link Receiver*

**Table 14. Single-Link JESD204B Operating Modes**

| | Number of Lanes (L) | | | | | |
|---|---|---|---|---|---|---|
| **Parameter** | **1** | **2** | **3** | **4** | **6** | **8** |
| L (Lane Count) | 1 | 2 | 3 | 4 | 6 | 8 |
| M (Converter Count) | 2 | 2 | 2 | 2 | 2 | 2 |
| F (Octets per Frame per Lane) | 4 | 2 | 4 | 1 | 2 | 1 |
| S (Samples per Converter per Frame) | 1 | 1 | 3 | 1 | 3 | 2 |

**Table 15. Data Structure per Lane for JESD204B Operating Modes[1]**

| JESD204B Parameters | Lane No. | Frame 0 | Frame 1 | Frame 2 | Frame 3 |
|---|---|---|---|---|---|
| L = 8, M = 2, F = 1, S = 2 | Lane 0 | M0S0[15:8] | | | |
| | Lane 1 | M0S0[7:0] | | | |
| | Lane 2 | M0S1[15:8] | | | |
| | Lane 3 | M0S1[7:0] | | | |
| | Lane 4 | M1S0[15:8] | | | |
| | Lane 5 | M1S0[7:0] | | | |
| | Lane 6 | M1S1[15:8] | | | |
| | Lane 7 | M1S1[7:0] | | | |
| L = 6, M = 2, F = 2, S = 3 | Lane 0 | M0S0[15:8] | M0S0[7:0] | | |
| | Lane 1 | M0S1[15:8] | M0S1[7:0] | | |
| | Lane 2 | M0S2[15:8] | M0S2[7:0] | | |
| | Lane 3 | M1S0[15:8] | M1S0[7:0] | | |
| | Lane 4 | M1S1[15:8] | M1S1[7:0] | | |
| | Lane 5 | M1S2[15:8] | M1S2[7:0] | | |
| L = 4, M = 2, F = 1, S = 1 | Lane 0 | M0S0[15:8] | | | |
| | Lane 1 | M0S0[7:0] | | | |
| | Lane 2 | M1S0[15:8] | | | |
| | Lane 3 | M1S0[7:0] | | | |
| L = 3, M = 2, F = 4, S = 3 | Lane 0 | M0S0[15:8] | M0S0[7:0] | M0S1[15:8] | M0S1[7:0] |
| | Lane 1 | M0S2[15:8] | M0S2[7:0] | M1S0[15:8] | M1S0[7:0] |
| | Lane 2 | M1S1[15:8] | M1S1[7:0] | M1S2[15:8] | M1S2[7:0] |
| L = 2, M = 2, F = 2, S = 1 | Lane 0 | M0S0[15:8] | M0S0[7:0] | | |
| | Lane 1 | M1S0[15:8] | M1S0[7:0] | | |
| L = 1, M = 2, F = 4, S = 1 | Lane 0 | M0S0[15:8] | M0S0[7:0] | M1S0[15:8] | M1S0[7:0] |

[1] Mx is the converter number and Sy is the sample number. For example, M0S0 means Converter 0, Sample 0. Blank cells are not applicable.

## PHYSICAL LAYER

The physical layer of the JESD204B interface, hereafter referred to as the deserializer, has eight identical channels. Each channel consists of the terminators, an equalizer, a clock and data recovery (CDR) circuit, and the 1:40 demux function (see Figure 91).

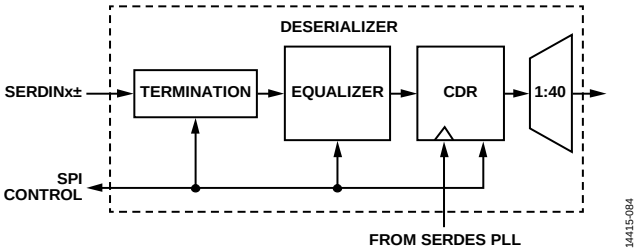


*Figure 91. Deserializer Block Diagram*

JESD204B data is input to the AD9163 via the SERDINx± 1.2 V differential input pins as per the JESD204B specification.

### *Interface Power-Up and Input Termination*

Before using the JESD204B interface, it must be powered up by setting Register 0x200, Bit 0 = 0. In addition, each physical lane (PHY) that is not being used (SERDINx±) must be powered down. To do so, set the corresponding Bit x for Physical Lane x in Register 0x201 to 0 if the physical lane is being used, and to 1 if it is not being used.

The AD9163 autocalibrates the input termination to 50 Ω. Before running the termination calibration, Register 0x2A7 and Register 0x2AE must be written as described in Table 16 to guarantee proper calibration. The termination calibration begins when Register 0x2A7, Bit 0 and Register 0x2AE, Bit 0 transition from low to high. Register 0x2A7 controls autocalibration for PHY 0, PHY 1, PHY 6, and PHY 7. Register 0x2AE controls autocalibration for PHY 2, PHY 3, PHY 4, and PHY 5.

The PHY termination autocalibration routine is as shown in Table 16.

**Table 16. PHY Termination Autocalibration Routine**

| Address | Value | Description |
|---|---|---|
| 0x2A7 | 0x01 | Autotune PHY 0, PHY 1, PHY 6, and PHY 7 terminations |
| 0x2AE | 0x01 | Autotune PHY 2, PHY 3, PHY 4, and PHY 5 terminations |

The input termination voltage of the DAC is sourced externally via the VTT_1P2 pins (K3 and K11). Set $V_{TT}$, the termination voltage, by connecting it to VDD_1P2. It is recommended that the JESD204B inputs be ac-coupled to the JESD204B transmit device using 100 nF capacitors.

The calibration code of the termination can be read from Bits[3:0] in Register 0x2AC (PHY 0, PHY 1, PHY 6, PHY 7) and Register 0x2B3 (PHY 2, PHY 3, PHY 4, PHY 5). If needed, the termination values can be adjusted or set using several registers. The TERM_BLKx_CTRLREG1 registers (Register 0x2A8 and Register 0x2AF), can override the autocalibrated value. When set to 0xXXX0XXXX, the termination block autocalibrates, which is the normal, default setting. When set to 0xXXX1XXXX, the autocalibration value is overwritten with the value in Bits[3:1] of Register 0x2A8 and Register 0x2AF. Individual offsets from the autocalibration value for each lane can be programmed in Bits[3:0] of Register 0x2BB to Register 0x2C2. The value is a signed magnitude, with Bit 3 as the sign bit. The total range of the termination resistor value is about 94 Ω to 120 Ω, with approximately 3.5% increments across the range (for example, smaller steps at the bottom of the range than at the top).

### Receiver Eye Mask

The AD9163 complies with the JESD204B specification regarding the receiver eye mask and is capable of capturing data that complies with this mask. Figure 92 shows the receiver eye mask normalized to the data rate interval with a 600 mV $V_{TT}$ swing. See the JESD204B specification for more information regarding the eye mask and permitted receiver eye opening.

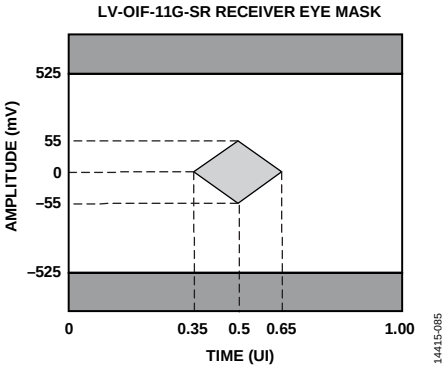


*Figure 92. Receiver Eye Mask for 600 mV $V_{TT}$ Swing*

### Clock Relationships

The following clocks rates are used throughout the rest of the JESD204B section. The relationship between any of the clocks can be derived from the following equations:

$$DataRate = (DACRate)/(InterpolationFactor)$$

$$LaneRate = (20 \times DataRate \times M)/L$$

$$ByteRate = LaneRate/10$$

This relationship comes from 8-bit/10-bit encoding, where each byte is represented by 10 bits.

$$PCLK\ Rate = ByteRate/4$$

The processing clock is used for a quad-byte decoder.

$$FrameRate = ByteRate/F$$

where $F$ is defined as octets per frame per lane.

$$PCLK\ Factor = FrameRate/PCLK\ Rate = 4/F$$

where:
$M$ is the JESD204B parameter for converters per link.
$L$ is the JESD204B parameter for lanes per link.
$F$ is the JESD204B parameter for octets per frame per lane.

### SERDES PLL

#### Functional Overview of the SERDES PLL

The independent SERDES PLL uses integer N techniques to achieve clock synthesis. The entire SERDES PLL is integrated on chip, including the VCO and the loop filter. The SERDES PLL VCO operates over the range of 6 GHz to 12.5 GHz.

In the SERDES PLL, a VCO divider block divides the VCO clock by 2 to generate a 3 GHz to 6.25 GHz quadrature clock for the deserializer cores. This clock is the input to the clock and data recovery block that is described in the Clock and Data Recovery section.

The reference clock to the SERDES PLL is always running at a frequency, $f_{REF}$, that is equal to 1/40 of the lane rate (PCLK Rate). This clock is divided by a DivFactor value (set by SERDES_PLL_DIV_FACTOR) to deliver a clock to the phase frequency detector (PFD) block that is between 35 MHz and 80 MHz. Table 17 includes the respective SERDES_PLL_DIV_FACTOR register settings for each of the desired PLL_REF_CLK_RATE options available.

**Table 17. SERDES PLL Divider Settings**

| Lane Rate (Gbps) | PLL_REF_CLK_RATE, Register 0x084, Bits[5:4] | SERDES_PLL_DIV_FACTOR Register 0x289, Bits[1:0] |
|---|---|---|
| 0.750 to 1.5625 | 0b01 = 2× | 0b10 = ÷1 |
| 1.5 to 3.125 | 0b00 = 1× | 0b10 = ÷1 |
| 3 to 6.25 | 0b00 = 1× | 0b01 = ÷2 |
| 6 to 12.5 | 0b00 = 1× | 0b00 = ÷4 |

Register 0x280 controls the synthesizer enable and recalibration.

To enable the SERDES PLL, first set the PLL divider register (see Table 17). Then enable the SERDES PLL by writing Register 0x280, Bit 0 = 1. If a recalibration is needed, write Register 0x280, Bit 2 = 0b1 and then reset the bit to 0b0. The rising edge of the bit causes a recalibration to begin.

Confirm that the SERDES PLL is working by reading Register 0x281. If Register 0x281, Bit 0 = 1, the SERDES PLL has locked. If Register 0x281, Bit 3 = 1, the SERDES PLL was successfully calibrated. If Register 0x281, Bit 4 or Bit 5 is high, the PLL reaches the lower or upper end of its calibration band and must be recalibrated by writing 0 and then 1 to Register 0x280, Bit 2.

### Clock and Data Recovery

The deserializer is equipped with a CDR circuit. Instead of recovering the clock from the JESD204B serial lanes, the CDR recovers the clocks from the SERDES PLL. The 3 GHz to 6.25 GHz output from the SERDES PLL, shown in Figure 94, is the input to the CDR.

A CDR sampling mode must be selected to generate the lane rate clock inside the device. If the desired lane rate is greater than 6.25 GHz, half rate CDR operation must be used. If the desired lane rate is less than 6.25 GHz, disable half rate operation. If the lane rate is less than 3 GHz, disable full rate and enable 2× oversampling to recover the appropriate lane rate clock. Table 18 gives a breakdown of CDR sampling settings that must be set depending on the lane rate value.

**Table 18. CDR Operating Modes**

| Lane Rate (Gbps) | SPI_ENHALFRATE Register 0x230, Bit 5 | SPI_DIVISION_RATE, Register 0x230, Bits[2:1] |
|---|---|---|
| 0.750 to 1.5625 | 0 (full rate) | 10b (divide by 4) |
| 1.5 to 3.125 | 0 (full rate) | 01b (divide by 2) |
| 3 to 6.25 | 0 (full rate) | 00b (no divide) |
| 6 to 12.5 | 1 (half rate) | 00b (no divide) |

The CDR circuit synchronizes the phase used to sample the data on each serial lane independently. This independent phase adjustment per serial interface ensures accurate data sampling and eases the implementation of multiple serial interfaces on a PCB.

After configuring the CDR circuit, reset it and then release the reset by writing 1 and then 0 to Register 0x206, Bit 0.

### Power-Down Unused PHYs

Note that any unused and enabled lanes consume extra power unnecessarily. Each lane that is not being used (SERDINx±) must be powered off by writing a 1 to the corresponding bit of PHY_PD (Register 0x201).

### Equalization

To compensate for signal integrity distortions for each PHY channel due to PCB trace length and impedance, the AD9163 employs an easy to use, low power equalizer on each JESD204B channel. The AD9163 equalizers can compensate for insertion losses far greater than required by the JESD204B specification. The equalizers have two modes of operation that are determined by the EQ_POWER_MODE register setting in Register 0x268, Bits[7:6]. In low power mode (Register 0x268, Bits[7:6] = 2b'01) and operating at the maximum lane rate of 12.5 Gbps, the equalizer can compensate for up to 11.5 dB of insertion loss. In normal mode (Register 0x268, Bits[7:6] = 2b'00), the equalizer can compensate for up to 17.2 dB of insertion loss. This performance is shown in Figure 93 as an overlay to the JESD204B specification for insertion loss. Figure 93 shows the equalization performance at 12.5 Gbps, near the maximum baud rate for the AD9163.

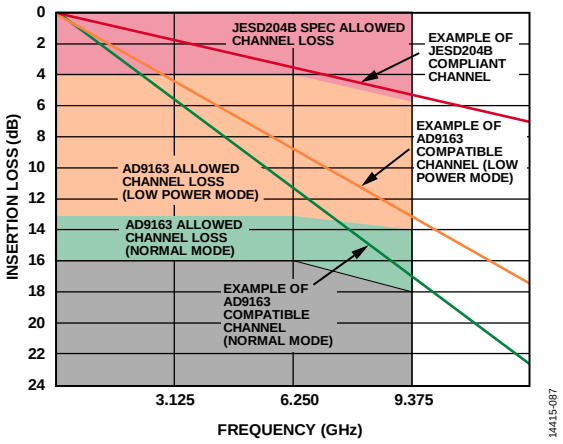


*Figure 93. Insertion Loss Allowed*

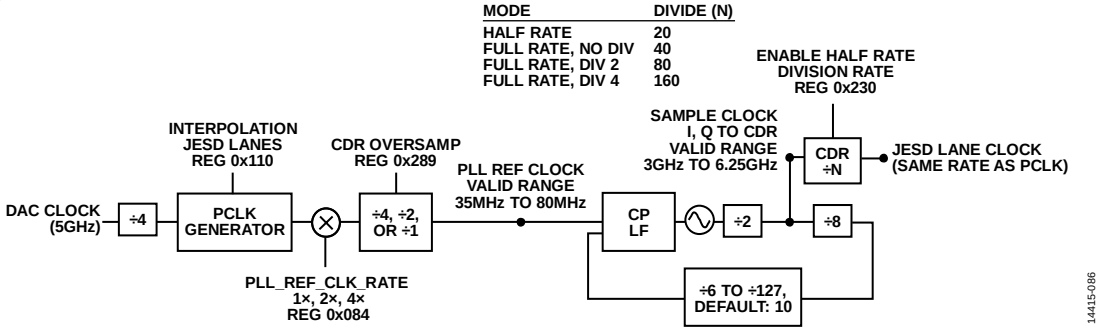


*Figure 94. SERDES PLL Synthesizer Block Diagram Including VCO Divider Block*

Figure 95 and Figure 96 are provided as points of reference for hardware designers and show the insertion loss for various lengths of well laid out stripline and microstrip transmission lines, respectively. See the Hardware Considerations section for specific layout recommendations for the JESD204B channel.

Low power mode is recommended if the insertion loss of the JESD204B PCB channels is less than that of the most lossy supported channel for low power mode (shown in Figure 93). If the insertion loss is greater than that, but still less than that of the most lossy supported channel for normal mode (shown in Figure 93), use normal mode. At 12.5 Gbps operation, the equalizer in normal mode consumes about 4 mW more power per lane used than in low power equalizer mode. Note that either mode can be used in conjunction with transmitter preemphasis to ensure functionality and/or optimize for power.

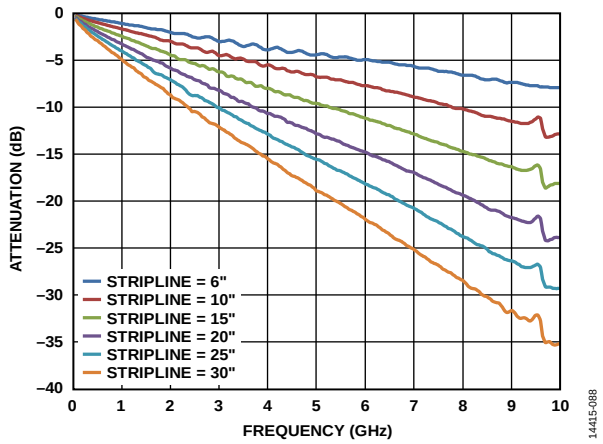


*Figure 95. Insertion Loss of 50 Ω Striplines on FR4*

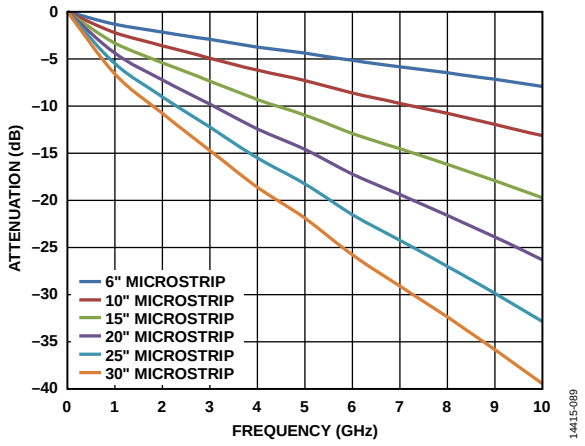


*Figure 96. Insertion Loss of 50 Ω Microstrips on FR4*

## DATA LINK LAYER

The data link layer of the AD9163 JESD204B interface accepts the deserialized data from the PHYs and deframes and descrambles them so that data octets are presented to the transport layer to be put into DAC samples. The architecture of the data link layer is shown in Figure 97. The data link layer consists of a synchronization FIFO for each lane, a crossbar switch, a deframer, and a descrambler.

The AD9163 can operate as a single-link high speed JESD204B serial data interface. All eight lanes of the JESD204B interface handle link layer communications such as code group synchronization (CGS), frame alignment, and frame synchronization.

The AD9163 decodes 8-bit/10-bit control characters, allowing marking of the start and end of the frame and alignment between serial lanes. Each AD9163 serial interface link can issue a synchronization request by setting its SYNCOUT± signal low. The synchronization protocol follows Section 4.9 of the JESD204B standard. When a stream of four consecutive /K/ symbols is received, the AD9163 deactivates the synchronization request by setting the SYNCOUT± signal high at the next internal LMFC rising edge. Then, the AD9163 waits for the transmitter to issue an initial lane alignment sequence (ILAS). During the ILAS, all lanes are aligned using the /A/ to /R/ character transition as described in the JESD204B Serial Link Establishment section. Elastic buffers hold early arriving lane data until the alignment character of the latest lane arrives. At this point, the buffers for all lanes are released and all lanes are aligned (see Figure 98).

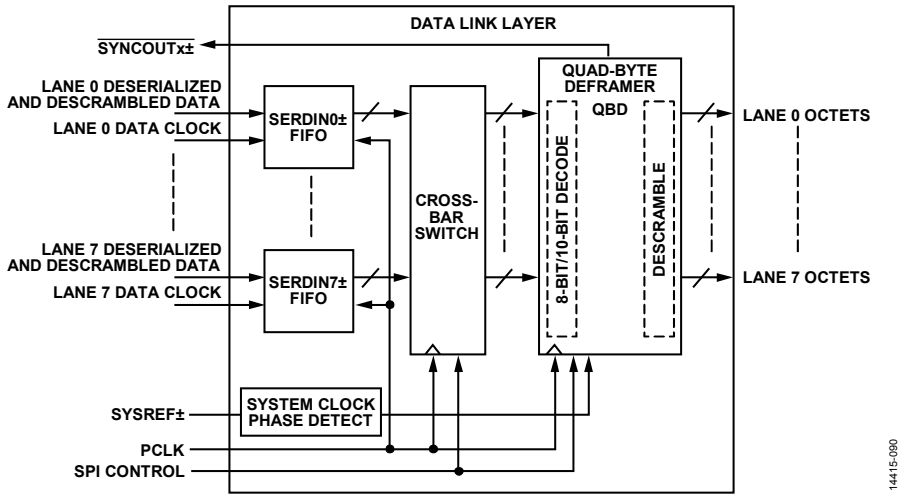


*Figure 97. Data Link Layer Block Diagram*

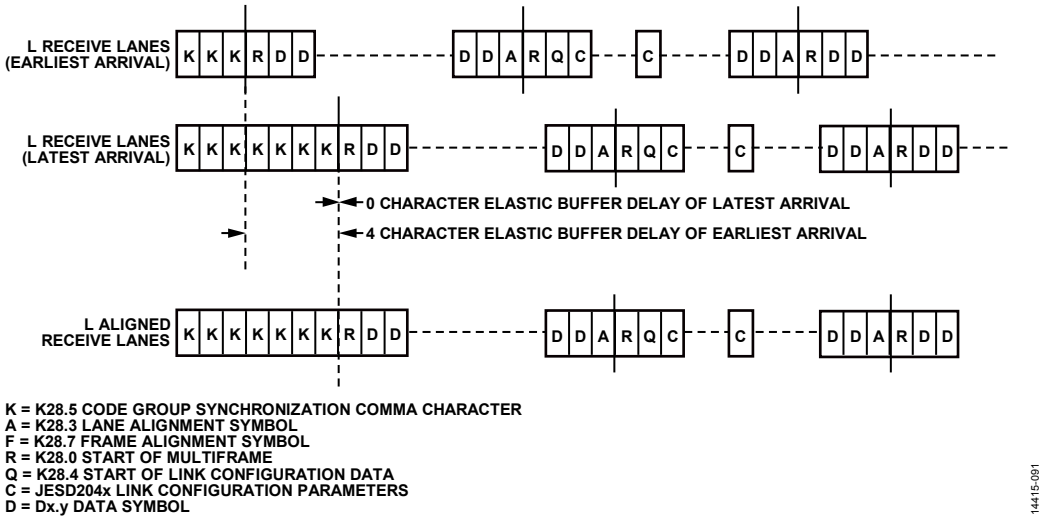


*Figure 98. Lane Alignment During ILAS*

### *JESD204B Serial Link Establishment*

A brief summary of the high speed serial link establishment process for Subclass 1 is provided. See Section 5.3.3 of the JESD204B specifications document for complete details.

**Step 1: Code Group Synchronization**

Each receiver must locate /K/ (K28.5) characters in its input data stream. After four consecutive /K/ characters are detected on all link lanes, the receiver block deasserts the $\overline{\text{SYNCOUT}}$± signal to the transmitter block at the receiver LMFC edge.

The transmitter captures the change in the $\overline{\text{SYNCOUT}}$± signal and at a future transmitter LMFC rising edge starts the ILAS.

**Step 2: Initial Lane Alignment Sequence**

The main purposes of this phase are to align all the lanes of the link and to verify the parameters of the link.

Before the link is established, write each of the link parameters to the receiver device to designate how data is sent to the receiver block.

The ILAS consists of four or more multiframes. The last character of each multiframe is a multiframe alignment character, /A/. The first, third, and fourth multiframes are populated with predetermined data values. Note that Section 8.2 of the JESD204B specifications document describes the data ramp that is expected during ILAS. The AD9163 does not require this ramp. The deframer uses the final /A/ of each lane to align the ends of the multiframes within the receiver. The second multiframe contains an /R/ (K.28.0), /Q/ (K.28.4), and then data corresponding to the link parameters. Additional multiframes can be added to the ILAS if needed by the receiver. By default, the AD9163 uses four multiframes in the ILAS (this can be changed in Register 0x478). If using Subclass 1, exactly four multiframes must be used.

After the last /A/ character of the last ILAS, multiframe data begins streaming. The receiver adjusts the position of the /A/

character such that it aligns with the internal LMFC of the receiver at this point.

**Step 3: Data Streaming**

In this phase, data is streamed from the transmitter block to the receiver block.

Optionally, data can be scrambled. Scrambling does not start until the very first octet following the ILAS.

The receiver block processes and monitors the data it receives for errors, including the following:

- Bad running disparity (8-bit/10-bit error)
- Not in table (8-bit/10-bit error)
- Unexpected control character
- Bad ILAS
- Interlane skew error (through character replacement)

If any of these errors exist, they are reported back to the transmitter in one of the following ways (see the JESD204B Error Monitoring section for details):

- $\overline{\text{SYNCOUT}}$± signal assertion: resynchronization ($\overline{\text{SYNCOUT}}$± signal pulled low) is requested at each error for the last two errors. For the first three errors, an optional resynchronization request can be asserted when the error counter reaches a set error threshold.
- For the first three errors, each multiframe with an error in it causes a small pulse on $\overline{\text{SYNCOUT}}$±.
- Errors can optionally trigger an interrupt request (IRQ) event, which can be sent to the transmitter.

For more information about the various test modes for verifying the link integrity, see the JESD204B Test Modes section.

### *Lane First In/First Out (FIFO)*

The FIFOs in front of the crossbar switch and deframer synchronize the samples sent on the high speed serial data interface with the deframer clock by adjusting the phase of the incoming data. The FIFO absorbs timing variations between the data source and the deframer; this allows up to two PCLK cycles of drift from the transmitter. The FIFO_STATUS_REG_0 register and FIFO_STATUS_REG_1 register (Register 0x30C and Register 0x30D, respectively) can be monitored to identify whether the FIFOs are full or empty.

**Lane FIFO IRQ**

An aggregate lane FIFO error bit is also available as an IRQ event. Use Register 0x020, Bit 2 to enable the FIFO error bit, and then use Register 0x024, Bit 2 to read back its status and reset the IRQ signal. See the Interrupt Request Operation section for more information.

### *Crossbar Switch*

Register 0x308 to Register 0x30B allow arbitrary mapping of physical lanes (SERDINx±) to logical lanes used by the SERDES deframers.

**Table 19. Crossbar Registers**

| Address | Bits | Logical Lane |
|---|---|---|
| 0x308 | [2:0] | SRC_LANE0 |
| 0x308 | [5:3] | SRC_LANE1 |
| 0x309 | [2:0] | SRC_LANE2 |
| 0x309 | [5:3] | SRC_LANE3 |
| 0x30A | [2:0] | SRC_LANE4 |
| 0x30A | [5:3] | SRC_LANE5 |
| 0x30B | [2:0] | SRC_LANE6 |
| 0x30B | [5:3] | SRC_LANE7 |

Write each SRC_LANEy with the number (x) of the desired physical lane (SERDINx±) from which to obtain data. By default, all logical lanes use the corresponding physical lane as their data source. For example, by default, SRC_LANE0 = 0; therefore, Logical Lane 0 obtains data from Physical Lane 0 (SERDIN0±). To use SERDIN4± as the source for Logical Lane 0 instead, the user must write SRC_LANE0 = 4.

### *Lane Inversion*

Register 0x334 allows inversion of desired logical lanes, which can be used to ease routing of the SERDINx± signals. For each Logical Lane x, set Bit x of Register 0x334 to 1 to invert it.

### *Deframer*

The AD9163 consists of one quad-byte deframer (QBD). The deframer accepts the 8-bit/10-bit encoded data from the deserializer (via the crossbar switch), decodes it, and descrambles it into JESD204B frames before passing it to the transport layer to be converted to DAC samples. The deframer processes four symbols (or octets) per processing clock (PCLK) cycle.

The deframer uses the JESD204B parameters that the user has programmed into the register map to identify how the data is packed, and unpacks it. The JESD204B parameters are described in detail in the Transport Layer section; many of the parameters are also needed in the transport layer to convert JESD204B frames into samples.

**Descrambler**

The AD9163 provides an optional descrambler block using a self synchronous descrambler with the following polynomial: $1 + x^{14} + x^{15}$.

Enabling data scrambling reduces spectral peaks that are produced when the same data octets repeat from frame to frame. It also makes the spectrum data independent so that possible frequency selective effects on the electrical interface do not cause data dependent errors. Descrambling of the data is enabled by setting the SCR bit (Register 0x453, Bit 7) to 1.

### *Syncing LMFC Signals*

The first step in guaranteeing synchronization across links and devices begins with syncing the LMFC signals. In Subclass 0, the LMFC signal is synchronized to an internal processing clock. In Subclass 1, LMFC signals are synchronized to an external SYSREF± signal.

#### SYSREF± Signal

The SYSREF± signal is a differential source synchronous input that synchronizes the LMFC signals in both the transmitter and receiver in a JESD204B Subclass 1 system to achieve deterministic latency.

The SYSREF± signal is a rising edge sensitive signal that is sampled by the device clock rising edge. It is best practice that the device clock and SYSREF± signals be generated by the same source, such as the HMC7044 clock generator, so that the phase alignment between the signals is fixed. When designing for optimum deterministic latency operation, consider the timing distribution skew of the SYSREF± signal in a multipoint link system (multichip).

The AD9163 supports a periodic SYSREF± signal. The periodicity can be continuous, strobed, or gapped periodic. The SYSREF± signal can always be dc-coupled (with a common-mode voltage of 0 V to 1.25 V). When dc-coupled, a small amount of common-mode current (<500 µA) is drawn from the SYSREF± pins. See Figure 99 for the SYSREF± internal circuit.

To avoid this common-mode current draw, use a 50% duty cycle periodic SYSREF± signal with ac coupling capacitors. If ac-coupled, the ac coupling capacitors combine with the resistors shown in Figure 99 to make a high-pass filter with an RC time constant of $\tau = RC$. Select C such that $\tau > 4/\text{SYSREF}\pm$ frequency. In addition, the edge rate must be sufficiently fast to meet the SYSREF± vs. DAC clock keep out window (KOW) requirements.

It is possible to use ac-coupled mode without meeting the frequency to time constant constraints ($\tau = RC$ and $\tau > 4/\text{SYSREF}\pm$ frequency) by using SYSREF± hysteresis (Register 0x088 and Register 0x089). However, using hystereis increases the DAC clock KOW (Table 6 does not apply) by an amount depending on the SYSREF± frequency, level of hysteresis, capacitor choice, and edge rate.

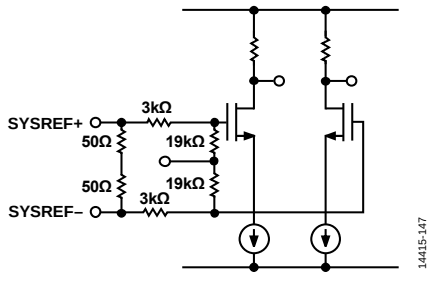


*Figure 99. SYSREF± Input Circuit*

#### Sync Processing Modes Overview

The AD9163 supports several LMFC sync processing modes. These modes are one-shot, continuous, and monitor modes.

All sync processing modes perform a phase check to confirm that the LMFC is phase aligned to an alignment edge. In Subclass 1, the SYSREF± rising edge acts as the alignment edge; in Subclass 0, an internal processing clock acts as the alignment edge.

The SYSREF± signal is sampled by a divide by 4 version of the DAC clock. After SYSREF± is sampled, the phase of the DAC clock/4 used to sample SYSREF± is stored in Register 0x037, Bits[7:0] and Register 0x038, Bits[3:0] as a thermometer code. This offset can be used by the SERDES data transmitter (for example, FPGA) to align multiple DACs by accounting for this clock offset when transmitting data. See the Sync Procedure section for details on the procedure for syncing the LMFC signals.

#### One-Shot Sync Mode (SYNC_MODE = Register 0x03A, Bits[1:0] = 0b10)

In one-shot sync mode, a phase check occurs on only the first alignment edge that is received after the sync machine is armed. After the phase is aligned on the first edge, the AD9163 transitions to monitor mode. Though an LMFC synchronization occurs only once, the SYSREF± signal can still be continuous. In this case, the phase is monitored and reported, but no clock phase adjustment occurs.

#### Continuous Sync Mode (SYNC_MODE = Register 0x03A, Bits[1:0] = 0b01)

Continuous mode must be used in Subclass 1 only with a periodic SYSREF± signal. In continuous mode, a phase check/alignment occurs on every alignment edge.

Continuous mode differs from one-shot mode in two ways. First, no SPI cycle is required to arm the device; the alignment edge seen after continuous mode is enabled results in a phase check. Second, a phase check occurs on every alignment edge in continuous mode.

#### Monitor Sync Mode (SYNC_MODE = Register 0x03A, Bits[1:0]) = 0b00)

In monitor mode, the user can monitor the phase error in real time. Use this sync mode with a periodic SYSREF± signal. The phase is monitored and reported, but no clock phase adjustment occurs.

When an alignment request (SYSREF± edge) occurs, snapshots of the last phase error are placed into readable registers for reference (Register 0x037 and Register 0x038, Bits[3:0]), and the IRQ_SYSREF_JITTER interrupt is set, if appropriate.

#### Sync Procedure

The procedure for enabling the sync is as follows:

1. Set up the DAC; the SERDES PLL locks it, and enables the CDR (see the Start-Up Sequence section).
2. Set Register 0x039 (SYSREF± jitter window). A minimum of 4 DAC clock cycles is recommended. See Table 21 for settings.
3. Optionally, read back the SYSREF± count to check whether the SYSREF± pulses are being received.

   a. Set Register 0x036 = 0. Writing anything to SYSREF_COUNT resets the count.
   b. Set Register 0x034 = 0. Writing anything to SYNC_LMFC_STAT0 saves the data for readback and registers the count.
   c. Read SYSREF_COUNT from the value from Register 0x036.
4. Perform a one-shot sync.
   a. Set Register 0x03A = 0x00. Clear one-shot mode if already enabled.
   b. Set Register 0x03A = 0x02. Enable one-shot sync mode. The state machine enters monitor mode after a sync occurs.
5. Optionally, read back the sync SYNC_LMFC_STATx registers to verify that sync completed correctly.
   a. Set Register 0x034 = 0. Register 0x034 must be written to read the value.
   b. Read Register 0x035 and Register 0x034 to find the value of SYNC_LMFC_STATx. It is recommended to set SYNC_LMFC_STATx to 0 but it can be set to 4, or a LMFC period in DAC clocks – 4, due to jitter.
6. Optionally, read back the sync SYSREF_PHASEx register to identify which phase of the divide by 4 was used to sample SYSREF±. Read Register 0x038 and Register 0x037 as thermometer code. The MSBs of Register 0x037, Bits[7:4], normally show the thermometer code value.
7. Turn the link on (Register 0x300, Bit 0 = 1).
8. Read back Register 0x302 (dynamic link latency).
9. Repeat the reestablishment of the link several times (Step 1 to Step 7) and note the dynamic link latency values. Based on the values, program the LMFC delay (Register 0x304) and the LMFC variable (Register 0x306), and then restart the link.

**Table 20. Sync Processing Modes**

| Sync Processing Mode | SYNC_MODE (Register 0x03A, Bits[1:0]) |
|---|---|
| No synchronization | 0b00 |
| One shot | 0b10 |
| Continuous | 0b01 |

**Table 21. SYSREF± Jitter Window Tolerance**

| SYSREF± Jitter Window Tolerance (DAC Clock Cycles) | SYSREF_JITTER_WINDOW (Register 0x039, Bits[5:0])[1] |
|---|---|
| ±½ | 0x00 |
| ±4 | 0x04 |
| ±8 | 0x08 |
| ±12 | 0x0C |
| ±16 | 0x10 |
| ±20 | 0x14 |
| +24 | 0x18 |
| ±28 | 0x1C |

[1] The two least significant digits are ignored because the SYSREF± signal is sampled with a divide by 4 version of the DAC clock. As a result, the jitter window is set by this divide by 4 clock rather than the DAC clock. It is recommended that at least a four-DAC clock SYSREF± jitter window be chosen.

### *Deterministic Latency*

JESD204B systems contain various clock domains distributed throughout its system. Data traversing from one clock domain to a different clock domain can lead to ambiguous delays in the JESD204B link. These ambiguities lead to nonrepeatable latencies across the link from power cycle to power cycle with each new link establishment. Section 6 of the JESD204B specification addresses the issue of deterministic latency with mechanisms defined as Subclass 1 and Subclass 2.

The AD9163 supports JESD204B Subclass 0 and Subclass 1 operation, but not Subclass 2. Write the subclass to Register 0x458, Bits[7:5].

#### Subclass 0

This mode gives deterministic latency to within 32 DAC clock cycles. It does not require any signal on the SYSREF± pins, which can be left disconnected.

Subclass 0 still requires that all lanes arrive within the same LMFC cycle and the dual DACs must be synchronized to each other.

#### Subclass 1

This mode gives deterministic latency and allows the link to be synced to within four DAC clock periods. It requires an external SYSREF± signal that is accurately phase aligned to the DAC clock.

#### Deterministic Latency Requirements

Several key factors are required for achieving deterministic latency in a JESD204B Subclass 1 system.

- SYSREF± signal distribution skew within the system must be less than the desired uncertainty.
- SYSREF± setup and hold time requirements must be met for each device in the system.
- The total latency variation across all lanes, links, and devices must be ≤10 PCLK periods, which includes both variable delays and the variation in fixed delays from lane to lane, link to link, and device to device in the system.

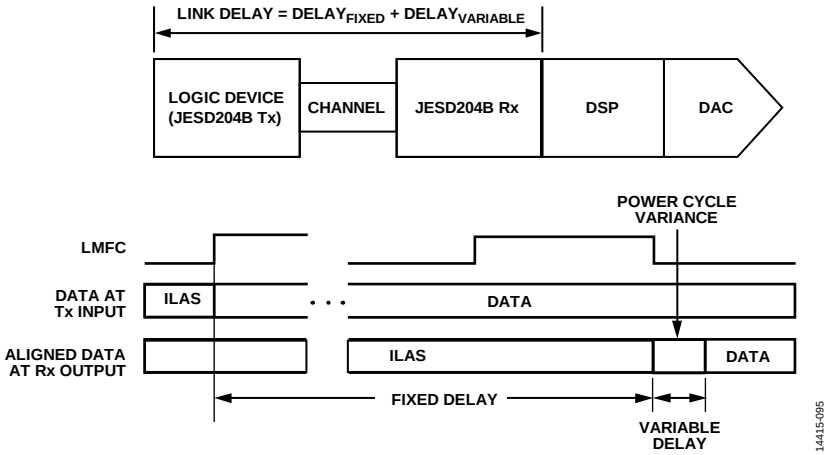


*Figure 100. JESD204B Link Delay = Fixed Delay + Variable Delay*

### *Link Delay*

The link delay of a JESD204B system is the sum of the fixed and variable delays from the transmitter, channel, and receiver as shown in Figure 100.

For proper functioning, all lanes on a link must be read during the same LMFC period. Section 6.1 of the JESD204B specification states that the LMFC period must be larger than the maximum link delay. For the AD9163, this is not necessarily the case; instead, the AD9163 uses a local LMFC for each link ($LMFC_{Rx}$) that can be delayed from the SYSREF± aligned LMFC. Because the LMFC is periodic, this delay can account for any amount of fixed delay. As a result, the LMFC period must only be larger than the variation in the link delays, and the AD9163 can achieve proper performance with a smaller total latency. Figure 101 and Figure 102 show a case where the link delay is greater than an LMFC period. Note that it can be accommodated by delaying $LMFC_{Rx}$.

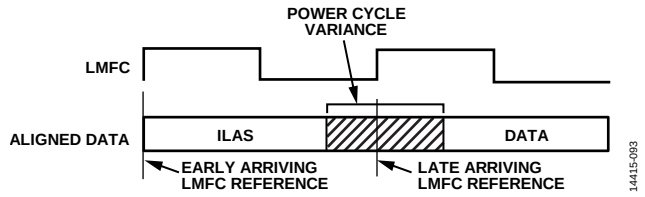


*Figure 101. Link Delay > LMFC Period Example*

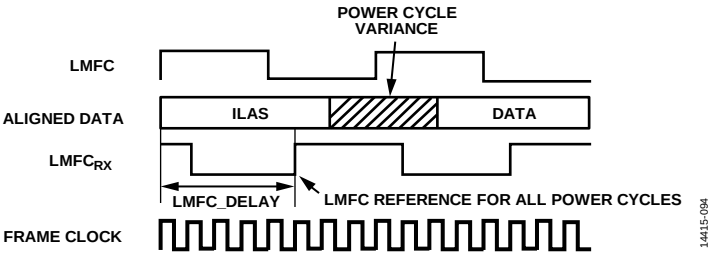


*Figure 102. LMFC_DELAY_x to Compensate for Link Delay > LMFC*

The method to select the LMFCDel (Register 0x304) and LMFCVar (Register 0x306) variables is described in the Link Delay Setup Example, With Known Delays section.

Setting LMFCDel appropriately ensures that all the corresponding data samples arrive in the same LMFC period. Then LMFCVar is written into the receive buffer delay (RBD) to absorb all link delay variation. This write ensures that all data samples have arrived before reading. By setting these to fixed values across runs and devices, deterministic latency is achieved.

The RBD described in the JESD204B specification takes values from 1 frame clock cycle to K frame clock cycles, and the RBD of the AD9163 takes values from 0 PCLK cycle to 10 PCLK cycles. As a result, up to 10 PCLK cycles of total delay variation can be absorbed. LMFCVar and LMFCDel are both in PCLK cycles. The PCLK factor, or number of frame clock cycles per PCLK cycle, is equal to 4/F. For more information on this relationship, see the Clock Relationships section.

Two examples follow that show how to determine LMFCVar and LMFCDel. After they are calculated, write LMFCDel into Register 0x304 for all devices in the system, and write LMFCVar to Register 0x306 for all devices in the system.

#### Link Delay Setup Example, With Known Delays

All the known system delays can be used to calculate LMFCVar and LMFCDel.

The example shown in Figure 103 is demonstrated in the following steps. Note that this example is in Subclass 1 to achieve deterministic latency, which has a PCLK factor (4/F) of 2 frame clock cycles per PCLK cycle, and uses K = 32 (frames/multiframe). Because PCBFixed << PCLK Period, PCBFixed is negligible in this example and not included in the calculations.

1. Find the receiver delays using Table 7.
   *RxFixed* = 12 PCLK cycles
   *RxVar* = 2 PCLK cycles
2. Find the transmitter delays. The equivalent table in the example JESD204B core (implemented on a GTH or GTX gigabit transceiver on a Virtex-6 FPGA) states that the delay is 56 ± 2 byte clock cycles.

3. Because the PCLK Rate = ByteRate/4 as described in the Clock Relationships section, the transmitter delays in PCLK cycles are calculated as follows:
   *TxFixed* = 54/4 = 13.5 PCLK cycles
   *TxVar* = 4/4 = 1 PCLK cycle
4. Calculate MinDelayLane as follows:
   *MinDelayLane* = floor(*RxFixed* + *TxFixed* + *PCBFixed*)
   = floor(12 + 13.5 + 0)
   = floor(25.5)
   *MinDelayLane* = 25
5. Calculate MaxDelayLane as follows:
   *MaxDelayLane* = ceiling(*RxFixed* + *RxVar* + *TxFixed* + *TxVar* + *PCBFixed*))
   = ceiling(12 + 2 + 13.5 + 1 + 0)
   = ceiling(28.5)
   *MaxDelayLane* = 29
6. Calculate LMFCVar as follows:
   *LMFCVar* = (*MaxDelay* + 1) − (*MinDelay* − 1)
   = (29 + 1) − (25 − 1) = 30 − 24
   *LMFCVar* = 6 PCLK cycles
7. Calculate LMFCDel as follows:
   *LMFCDel* = (*MinDelay* − 1) % (*K*/*PClockFactor*)
   = (30 − 1) % (32/2)
   = 29 % 16
   *LMFCDel* = 13 PCLK cycles
8. Write LMFCDel to Register 0x304 for all devices in the system. Write LMFCVar to Register 0x306 for all devices in the system.

### Link Delay Setup Example, Without Known Delay

If the system delays are not known, the AD9163 can read back the link latency between $LMFC_{RX}$ for each link and the SYSREF± aligned LMFC. This information is then used to calculate LMFCVar and LMFCDel.

Figure 105 shows how DYN_LINK_LATENCY_0 (Register 0x302) provides a readback showing the delay (in PCLK cycles) between $LMFC_{RX}$ and the transition from ILAS to the first data sample. By repeatedly power cycling and taking this measurement, the minimum and maximum delays across power cycles can be determined and used to calculate LMFCVar and LMFCDel.

In Figure 105, for Link A, Link B, and Link C, the system containing the AD9163 (including the transmitter) is power cycled and configured 20 times. The AD9163 is configured as described in the Sync Procedure section. Because the purpose of this exercise is to determine LMFCDel and LMFCVar, the LMFCDel value is programmed to 0 and the DYN_LINK_LATENCY_0 value is read from Register 0x302. The variation in the link latency over the 20 runs is shown in Figure 105, described as follows:

- Link A gives readbacks of 6, 7, 0, and 1. Note that the set of recorded delay values rolls over the edge of a multiframe at the boundary of K/PCLK factor = 8. Add the number of PCLK cycles per multiframe = 8 to the readback values of 0 and 1 because they rolled over the edge of the multiframe. Delay values range from 6 to 9.
- Link B gives delay values from 5 to 7.
- Link C gives delay values from 4 to 7.

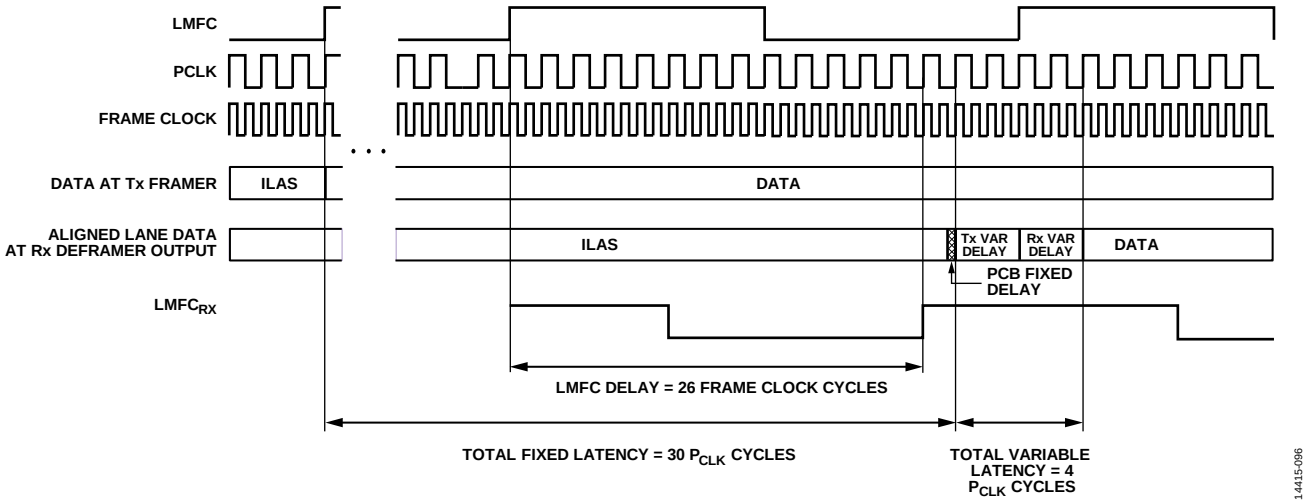


*Figure 103. LMFC Delay Calculation Example*

The example shown in Figure 105 is demonstrated in the following steps. Note that this example is in Subclass 1 to achieve deterministic latency, which has a PCLK factor (FrameRate ÷ PCLK Rate) of 4 and uses K = 32; therefore PCLK cycles per multiframe = 8.

1. Calculate the minimum of all delay measurements across all power cycles, links, and devices as follows:
   *MinDelay* = min(all *Delay* values) = 4
2. Calculate the maximum of all delay measurements across all power cycles, links, and devices as follows:
   *MaxDelay* = max(all *Delay* values) = 9
3. Calculate the total delay variation (with guard band) across all power cycles, links, and devices as follows:
   *LMFCVar* = (*MaxDelay* + 1) − (*MinDelay* − 1)
   = (9 + 1) − (4 − 1) = 10 − 3 = 7 PCLK cycles
4. Calculate the minimum delay in PCLK cycles (with guard band) across all power cycles, links, and devices as follows:
   *LMFCDel* = (*MinDelay* − 1) % (*K*/*PCLK Factor*)
   = (4 − 1) % 32/4
   = 3 % 8 = 3 PCLK cycles
5. Write LMFCDel to Register 0x304 for all devices in the system. Write LMFCVar to Register 0x306 for all devices in the system.

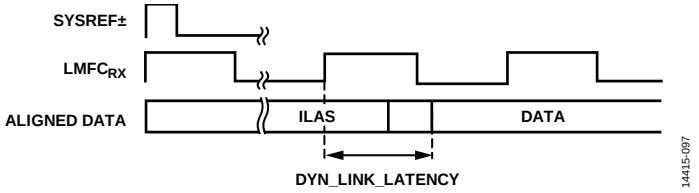


*Figure 104. DYN_LINK_LATENCY_x Illustration*

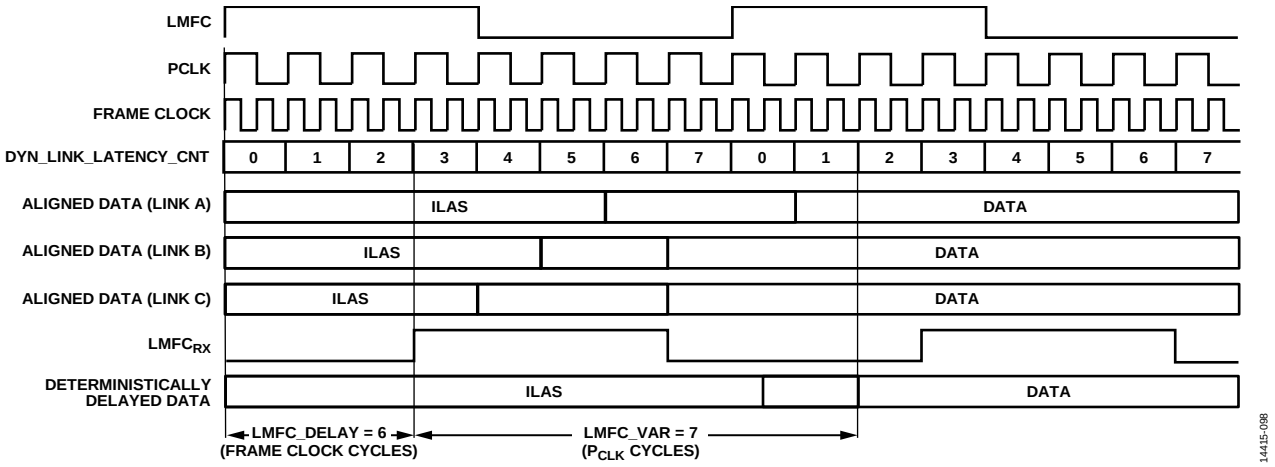


*Figure 105. Multilink Synchronization Settings, Derived Method Example*

## TRANSPORT LAYER

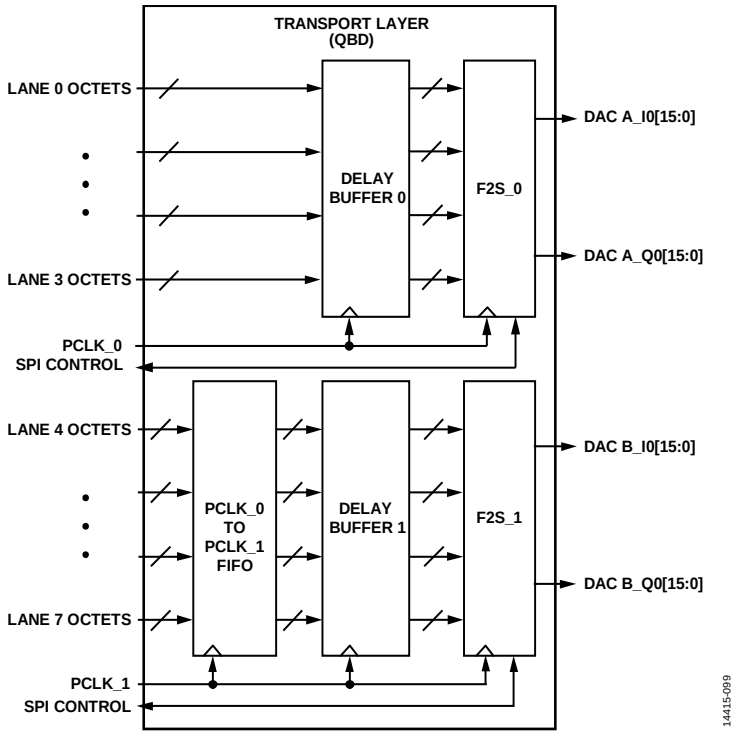


*Figure 106. Transport Layer Block Diagram*

The transport layer receives the descrambled JESD204B frames and converts them to DAC samples based on the programmed JESD204B parameters shown in Table 22. The device parameters are defined in Table 23.

**Table 22. JESD204B Transport Layer Parameters**

| Parameter | Description |
|---|---|
| F | Number of octets per frame per lane: 1, 2, or 4 |
| K | Number of frames per multiframe: K = 32 |
| L | Number of lanes per converter device (per link), as follows: 4 or 8 |
| M | Number of converters per device (per link), as follows: 2 (2 is used for the AD9163) |
| S | Number of samples per converter, per frame: 1 or 2 |

**Table 23. JESD204B Device Parameters**

| Parameter | Description |
|---|---|
| CF | Number of control words per device clock per link. Not supported, must be 0. |
| CS | Number of control bits per conversion sample. Not supported, must be 0. |
| HD | High density user data format. Used when samples must be split across lanes. Set to 1 always, even when F does not equal 1. Otherwise, a link configuration error triggers and the IRQ_ILAS flag is set. |
| N | Converter resolution = 16. |
| N' (or NP) | Total number of bits per sample = 16. |

Certain combinations of these parameters are supported by the AD9163. See Table 26 for a list of supported interpolation rates and the number of lanes that is supported for each rate. Table 26 lists the JESD204B parameters for each of the interpolation and number of lanes configuration, and gives an example lane rate for a 5 GHz DAC clock. Table 25 lists JESD204B parameters that have fixed values. A value of yes in Table 24 means the interpolation rate is supported for the number of lanes. A blank cell means it is not supported.

**Table 24. Interpolation Rates and Number of Lanes**

| Interpolation | 8 | 6 | 4 | 3 | 2 | 1 |
|---|---|---|---|---|---|---|
| 6× | Yes | Yes | Yes | Yes | | |
| 8× | Yes | Yes | Yes | Yes | Yes | |
| 12× | Yes | Yes | Yes | Yes | Yes | |
| 16× | Yes | Yes | Yes | Yes | Yes | Yes |
| 24× | Yes | Yes | Yes | Yes | Yes | Yes |

**Table 25. JESD204B Parameters with Fixed Values**

| Parameter | Value |
|---|---|
| K | 32 |
| N | 16 |
| NP | 16 |
| CF | 0 |
| HD | 1 |
| CS | 0 |

**Table 26. JESD204B Parameters for Interpolation Rate and Number of Lanes**

| Interpolation Rate | No. of Lanes | M | F | S | PCLK Period (DAC Clocks) | LMFC Period (DAC Clocks) | Lane Rate at 5 GHz DAC Clock (GHz) |
|---|---|---|---|---|---|---|---|
| 6 | 3 | 2 | 4 | 3 | 18 | 576 | 11.11 |
| 6 | 4 | 2 | 1 | 1 | 24 | 192 | 8.33 |
| 6 | 6 | 2 | 2 | 3 | 36 | 576 | 5.55 |
| 6 | 8 | 2 | 1 | 2 | 48 | 384 | 4.16 |
| 8 | 2 | 2 | 2 | 1 | 16 | 256 | 12.5 |
| 8 | 3 | 2 | 4 | 3 | 24 | 768 | 8.33 |
| 8 | 4 | 2 | 1 | 1 | 32 | 256 | 6.25 |
| 8 | 6 | 2 | 2 | 3 | 48 | 768 | 4.16 |
| 8 | 8 | 2 | 1 | 2 | 64 | 512 | 3.12 |
| 12 | 2 | 2 | 2 | 1 | 24 | 384 | 8.33 |
| 12 | 3 | 2 | 4 | 3 | 36 | 1152 | 5.55 |
| 12 | 4 | 2 | 1 | 1 | 48 | 384 | 4.16 |
| 12 | 6 | 2 | 2 | 3 | 72 | 1152 | 2.77 |
| 12 | 8 | 2 | 1 | 2 | 96 | 768 | 2.08 |
| 16 | 1 | 2 | 4 | 1 | 16 | 512 | 12.5 |
| 16 | 2 | 2 | 2 | 1 | 32 | 512 | 6.25 |
| 16 | 3 | 2 | 4 | 3 | 48 | 1536 | 4.16 |
| 16 | 4 | 2 | 1 | 1 | 64 | 512 | 3.12 |
| 16 | 6 | 2 | 2 | 3 | 96 | 1536 | 2.08 |
| 16 | 8 | 2 | 1 | 2 | 128 | 1024 | 1.56 |
| 24 | 1 | 2 | 4 | 1 | 24 | 768 | 8.33 |
| 24 | 2 | 2 | 2 | 1 | 48 | 768 | 4.16 |
| 24 | 3 | 2 | 4 | 3 | 72 | 2304 | 2.77 |
| 24 | 4 | 2 | 1 | 1 | 96 | 768 | 2.08 |
| 24 | 6 | 2 | 2 | 3 | 144 | 2304 | 1.38 |
| 24 | 8 | 2 | 1 | 2 | 192 | 1536 | 1.04 |

### Configuration Parameters

The AD9163 modes refer to the link configuration parameters for L, K, M, N, NP, S, and F. Table 27 provides the description and addresses for these settings.

**Table 27. Configuration Parameters**

| JESD204B Setting | Description | Address |
|---|---|---|
| L – 1 | Number of lanes minus 1. | Register 0x453, Bits[4:0] |
| F – 1 | Number of ((octets per frame) per lane) minus 1. | Register 0x454, Bits[7:0] |
| K – 1 | Number of frames per multiframe minus 1. | Register 0x455, Bits[4:0] |
| M – 1 | Number of converters minus 1. | Register 0x456, Bits[7:0] |
| N – 1 | Converter bit resolution minus 1. | Register 0x457, Bits[4:0] |
| NP – 1 | Bit packing per sample minus 1. | Register 0x458, Bits[4:0] |
| S – 1 | Number of ((samples per converter) per frame) minus 1. | Register 0x459, Bits[4:0] |
| HD | High density format. Set to 1 if F = 1. Leave at 0 if F ≠ 1. | Register 0x45A, Bit 7 |
| DID | Device ID. Match the device ID sent by the transmitter. | Register 0x450, Bits[7:0] |
| BID | Bank ID. Match the bank ID sent by the transmitter. | Register 0x451, Bits[7:0] |
| LID0 | Lane ID for Lane 0. Match the Lane ID sent by the transmitter on Logical Lane 0. | Register 0x452, Bits[4:0] |
| JESDV | JESD204x version. Match the version sent by the transmitter (0x0 = JESD204A, 0x1 = JESD204B). | Register 0x459, Bits[7:5] |

### Data Flow Through the JESD204B Receiver

The link configuration parameters determine how the serial bits on the JESD204B receiver interface are deframed and passed on to the DACs as data samples.

### Deskewing and Enabling Logical Lanes

After proper configuration, the logical lanes are automatically deskewed. All logical lanes are enabled or not based on the lane number setting in Register 0x110, Bits[7:4]. The physical lanes are all powered up by default.

To disable power to physical lanes that are not being used, set Bit x in Register 0x201 to 1 to disable Physical Lane x, and keep it at 0 to enable it.

## JESD204B TEST MODES

### PHY PRBS Testing

The JESD204B receiver on the AD9163 includes a PRBS pattern checker on the back end of its physical layer. This functionality enables bit error rate (BER) testing of each physical lane of the JESD204B link. The PHY PRBS pattern checker does not require that the JESD204B link be established. It can synchronize with a PRBS7, PRBS15, or PRBS31 data pattern. PRBS pattern verification can be done on multiple lanes at once. The error counts for failing lanes are reported for one JESD204B lane at a time. The process for performing PRBS testing on the AD9163 is as follows:

1. Start sending a PRBS7, PRBS15, or PRBS31 pattern from the JESD204B transmitter.
2. Select and write the appropriate PRBS pattern to Register 0x316, Bits[3:2], as shown in Table 28.
3. Enable the PHY test for all lanes being tested by writing to PHY_TEST_EN (Register 0x315). Each bit of Register 0x315 enables the PRBS test for the corresponding lane. For example, writing a 1 to Bit 0 enables the PRBS test for Physical Lane 0.
4. Toggle PHY_TEST_RESET (Register 0x316, Bit 0) from 0 to 1 then back to 0.
5. Set PHY_PRBS_TEST_THRESHOLD_xBITS (Bits[23:0], Register 0x319 to Register 0x317) as desired.
6. Write a 0 and then a 1 to PHY_TEST_START (Register 0x316, Bit 1). The rising edge of PHY_TEST_START starts the test.
   a. (Optional) In some cases, it may be necessary to repeat Step 4 at this point. Toggle PHY_TEST_RESET (Register 0x316, Bit 0) from 0 to 1, then back to 0.
7. Wait 500 ms.
8. Stop the test by writing PHY_TEST_START (Register 0x316, Bit 1) = 0.
9. Read the PRBS test results.
   a. Each bit of PHY_PRBS_PASS (Register 0x31D) corresponds to one SERDES lane (0 = fail, 1 = pass).

The number of PRBS errors seen on each failing lane can be read by writing the lane number to check (0 to 7) in PHY_SRC_ERR_CNT (Register 0x316, Bits[6:4]) and reading the PHY_PRBS_ERR_COUNT (Register 0x31C to Register 0x31A). The maximum error count is $2^{24-1}$. If all bits of Register 0x31C to Register 0x31A are high, the maximum error count on the selected lane is exceeded.

**Table 28. PHY PRBS Pattern Selection**

| PHY_PRBS_PAT_SEL Setting (Register 0x316, Bits[3:2]) | PRBS Pattern |
|---|---|
| 0b00 (default) | PRBS7 |
| 0b01 | PRBS15 |
| 0b10 | PRBS31 |

### Transport Layer Testing

The JESD204B receiver in the AD9163 supports the short transport layer (STPL) test as described in the JESD204B standard. This test can be used to verify the data mapping between the JESD204B transmitter and receiver. To perform this test, this function must be implemented in the logic device and enabled there. Before running the test on the receiver side, the link must be established and running without errors.

The STPL test ensures that each sample from each converter is mapped appropriately according to the number of converters (M) and the number of samples per converter (S). As specified in the JESD204B standard, the converter manufacturer specifies what test samples are transmitted. Each sample must have a

unique value. For example, if M = 2 and S = 2, four unique samples are transmitted repeatedly until the test is stopped. The expected sample must be programmed into the device and the expected sample is compared to the received sample one sample at a time until all are tested. The process for performing this test on the AD9163 is described as follows:

1. Synchronize the JESD204B link.
2. Enable the STPL test at the JESD204B Tx.
3. Depending on JESD204B case, there may be up to two DACs, and each frame may contain up to four DAC samples. Configure the SHORT_TPL_REF_SP_MSB bits (Register 0x32E) and SHORT_TPL_REF_SP_LSB bits (Register 0x32D) to match one of the samples for one converter within one frame.
4. Set SHORT_TPL_SP_SEL (Register 0x32C, Bits[7:4]) to select the sample within one frame for the selected converter according to Table 29.
5. Set SHORT_TPL_TEST_EN (Register 0x32C, Bit 0) to 1.
6. Set SHORT_TPL_TEST_RESET (Register 0x32C, Bit 1) to 1, then back to 0.
7. Wait for the desired time. The desired time is calculated as 1/(sample rate × BER). For example, given a bit error rate of BER = $1 \times 10^{-10}$ and a sample rate = 1 GSPS, the desired time = 10 sec. Then, set SHORT_TPL_TEST_EN to 0.
8. Read the test result at SHORT_TPL_FAIL (Register 0x32F, Bit 0).
9. Choose another sample for the same or another converter to continue with the test, until all samples for both converters from one frame are verified. (Note that the converter count is M = 2 for all interpolator modes on the AD9163 to enable complex signal processing.)

Consult Table 29 for a guide to the test sample alignment. Note that the sample order for 1×, eight-lane mode has Sample 1 and Sample 2 swapped. Also, the STPL test for the three-lane and six-lane options is not functional and always fails.

**Table 29. Short TPL Test Samples Assignment[1]**

| JESD204x Mode | Required Samples from JESD204x Tx | Samples Assignment |
|---|---|---|
| 6× Eight-Lane (L = 8, M = 2, F = 1, S = 2)<br>8× Eight-Lane (L = 8, M = 2, F = 1, S = 2)<br>12× Eight-Lane e (L = 8, M = 2, F = 1, S = 2)<br>16× Eight-Lane (L = 8, M = 2, F = 1, S = 2)<br>24× Eight-Lane (L = 8, M = 2, F = 1, S = 2) | Send four samples: M0S0, M0S1, M1S0, M1S1, and repeat | SP0: M0S0, SP4: M0S0, SP8: M0S0, SP12: M0S0<br>SP1: M1S0, SP5: M1S0, SP9: M1S0, SP13: M1S0<br>SP2: M0S1, SP6: M0S1, SP10: M0S1, SP14: M0S1<br>SP3: M1S1, SP7: M1S1, SP11: M1S1, SP15: M1S1 |
| 6× Six-Lane (L = 6, M = 2, F = 2, S = 3)<br>8× Six-Lane (L = 6, M = 2, F = 2, S = 3)<br>12× Six-Lane (L = 6, M = 2, F = 2, S = 3)<br>16× Six-Lane (L = 6, M = 2, F = 2, S = 3)<br>24× Six-Lane (L = 6, M = 2, F = 2, S = 3)<br>4× Three-Lane (L = 3, M = 2, F = 4, S = 3)<br>6× Three-Lane (L = 3, M = 2, F = 4, S = 3)<br>8× Three-Lane (L = 3, M = 2, F = 4, S = 3)<br>12× Three-Lane (L = 3, M = 2, F = 4, S = 3)<br>16× Three-Lane (L = 3, M = 2, F = 4, S = 3)<br>24× Three-Lane (L = 3, M = 2, F = 4, S = 3) | Send six samples: M0S0, M0S1, M0S2, M1S0, M1S1, M1S2, and repeat | Test hardware is not functional; STPL always fails |
| 6× Four-Lane (L = 4, M = 2, F = 1, S = 1)<br>8× Four-Lane (L = 4, M = 2, F = 1, S = 1)<br>12× Four-Lane (L = 4, M = 2, F = 1, S = 1)<br>16× Four-Lane (L = 4, M = 2, F = 1, S = 1)<br>24× Four-Lane (L = 4, M = 2, F = 1, S = 1)<br>8× Two-Lane (L = 2, M = 2, F = 2, S = 1)<br>12× Two-Lane (L = 2, M = 2, F = 2, S = 1)<br>16× Two-Lane (L = 2, M = 2, F = 2, S = 1)<br>24× Two-Lane (L = 2, M = 2, F = 2, S = 1)<br>16× One-Lane (L = 1, M = 2, F = 4, S = 1)<br>24× One-Lane (L = 1, M = 2, F = 4, S = 1) | Send two samples: M0S0, M1S0, repeat | SP0: M0S0, SP4: M0S0, SP8: M0S0, SP12: M0S0<br>SP1: M1S0, SP5: M1S0, SP9: M1S0, SP13: M1S0<br>SP2: M0S0, SP6: M0S0, SP10: M0S0, SP14: M0S0<br>SP3: M1S0, SP7: M1S0, SP11: M1S0, SP15: M1S0 |

[1] Mx is the converter number and Sy is the sample number. For example, M0S0 means Converter 0, Sample 0. SPx is the sample pattern word number. For example, SP0 means Sample Pattern Word 0.

### *Repeated CGS and ILAS Test*

As per Section 5.3.3.8.2 of the JESD204B specification, the AD9163 can check that a constant stream of /K28.5/ characters is being received, or that CGS followed by a constant stream of ILAS is being received.

To run a repeated CGS test, send a constant stream of /K28.5/ characters to the AD9163 SERDES inputs. Next, set up the device and enable the links. Ensure that the /K28.5/ characters are being received by verifying that SYNCOUT± is deasserted and that CGS has passed for all enabled link lanes by reading Register 0x470.

To run the CGS followed by a repeated ILAS sequence test, follow the procedure to set up the links, but before performing the last write (enabling the links), enable the ILAS test mode by writing a 1 to Register 0x477, Bit 7. Then, enable the links. When the device recognizes four CGS characters on each lane, it deasserts the SYNCOUT±. At this point, the transmitter starts sending a repeated ILAS sequence.

Read Register 0x473 to verify that initial lane synchronization has passed for all enabled link lanes.

## JESD204B ERROR MONITORING

### *Disparity, Not in Table, and Unexpected Control (K) Character Errors*

As per Section 7.6 of the JESD204B specification, the AD9163 can detect disparity errors, not in table (NIT) errors, and unexpected control character errors, and can optionally issue a sync request and reinitialize the link when errors occur.

Note that the disparity error counter counts all characters with invalid disparity, regardless of whether they are in the 8-bit/10-bit decoding table. This is a minor deviation from the JESD204B specification, which only counts disparity errors when they are in the 8-bit/10-bit decoding table.

Several other interpretations of the JESD204B specification are noted in this section. When three NIT errors are injected to one lane and QUAL_RDERR (Register 0x476, Bit 4) = 1, the readback values of the bad disparity error (BDE) count register is 1. Reporting of disparity errors that occur at the same character position of an NIT error is disabled. No such disabling is performed for the disparity errors in the characters after an NIT error. Therefore, it is expected behavior that an NIT error may result in a BDE error.

A resync is triggered when four NIT errors are injected with Register 0x476, Bit 4 = 1. When this bit is set, the error counter does not distinguish between a concurrent invalid symbol with the wrong running disparity but is in the 8-bit/10-bit decoding table, and an NIT error. Thus, a resync can be triggered when four NIT errors are injected because they are not distinguished from disparity errors.

#### Checking Error Counts

The error count can be checked for disparity errors, NIT errors, and unexpected control character errors. The error counts are on a per lane and per error type basis. Each error type and lane has a register dedicated to it. To check the error count, the following steps must be performed:

1. Choose and enable which errors to monitor by selecting them in Register 0x480, Bits[5:3] to Register 0x487, Bits[5:3]. Unexpected K (UEK) character, BDE, and NIT error monitoring can be selected for each lane by writing a 1 to the appropriate bit, as described in the register map. These bits are enabled by default.
2. The corresponding error counter reset bits are in Register 0x480, Bits[2:0] to Register 0x487, Bits[2:0]. Write a 0 to the corresponding bit to reset that error counter.
3. Registers 0x488, Bits[2:0] to Register 0x48F, Bits[2:0] have the terminal count hold indicator for each error counter. If this flag is enabled, when the terminal error count of 0xFF is reached, the counter ceases counting and holds that value until reset. Otherwise, it wraps to 0x00 and continues counting. Select the desired behavior and program the corresponding register bits per lane.

#### Check for Error Count Over Threshold

To check for the error count over threshold, follow these steps:

1. Define the error counter threshold. The error counter threshold can be set to a user defined value in Register 0x47C, or left to the default value of 0xFF. When the error threshold is reached, an IRQ is generated or SYNCOUT± is asserted or both, depending on the mask register settings. This one error threshold is used for all three types of errors (UEK, NIT, and BDE).
2. Set the SYNC_ASSERT_MASK bits. The SYNCOUT± assertion behavior is set in Register 0x47D, Bits[2:0]. By default, when any error counter of any lane is equal to the threshold, it asserts SYNCOUT± (Register 0x47D, Bits[2:0] = 0b111).
3. Read the error count reached indicator. Each error counter has a terminal count reached indicator, per lane. This indicator is set to 1 when the terminal count of an error counter for a particular lane has been reached. These status bits are located in Register 0x490, Bits[2:0] to Register 0x497, Bits[2:0]. These registers also indicate whether a particular lane is active by setting Bit 3 = 0b1.

#### Error Counter and IRQ Control

For error counter and IRQ control, follow these steps:

1. Enable the JESD204B interrupts. The interrupts for the UEK, NIT, and BDE error counters are in Register 0x4B8, Bits[7:5]. There are other interrupts to monitor when bringing up the link, such as lane deskewing, initial lane sync, good check sum, frame sync, code group sync (Register 0x4B8, Bits[4:0], and configuration mismatch (Register 0x4B9, Bit 0). These bits are off by default but can be enabled by writing 0b1 to the corresponding bit.

2. Read the JESD204B interrupt status. The interrupt status bits are in Register 0x4BA, Bits[7:0] and Register 0x4BB, Bit 0, with the status bit position corresponding to the enable bit position.
3. It is recommended to enable all interrupts that are planned to be used prior to bringing up the JESD204B link. When the link is up, the interrupts can be reset and then used to monitor the link status.

### Monitoring Errors via $\overline{\text{SYNCOUT}}$±

When one or more disparity, NIT, or unexpected control character errors occur, the error is reported on the $\overline{\text{SYNCOUT}}$± pin as per Section 7.6 of the JESD204B specification. The JESD204B specification states that the $\overline{\text{SYNCOUT}}$± signal is asserted for exactly two frame periods when an error occurs. For the AD9163, the width of the$\overline{\text{SYNCOUT}}$± pulse can be programmed to ½, 1, or 2 PCLK cycles. The settings to achieve a $\overline{\text{SYNCOUT}}$± pulse of two frame clock cycles are given in Table 30.

**Table 30. Setting $\overline{\text{SYNCOUT}}$± Error Pulse Duration**

| F | PCLK Factor (Frames/PCLK) | $\overline{\text{SYNC}}$_ERR_DUR (Register 0x312, Bits[7:4]) Setting[1] |
|---|---|---|
| 1 | 4 | 0 (default) |
| 2 | 2 | 1 |
| 4 | 1 | 2 |

[1] These register settings assert the $\overline{\text{SYNCOUT}}$± signal for two frame clock cycle pulse widths.

### Unexpected Control Character, NIT, Disparity IRQs

For UEK character, NIT, and disparity errors, error count over the threshold events are available as IRQ events. Enable these events by writing to Register 0x4B8, Bits[7:5]. The IRQ event status can be read at Register 0x4BA, Bits[7:5] after the IRQs are enabled.

See the Error Counter and IRQ Control section for information on resetting the IRQ. See the Interrupt Request Operation section for more information on IRQs.

### Errors Requiring Reinitializing

A link reinitialization automatically occurs when four invalid disparity characters are received as per Section 7.1 of the JESD204B specification. When a link reinitialization occurs, the resync request is five frames and nine octets long.

The user can optionally reinitialize the link when the error count for disparity errors, NIT errors, or UEK character errors reaches a programmable error threshold. The process to enable the reinitialization feature for certain error types is as follows:

1. Choose and enable which errors to monitor by selecting them in Register 0x480, Bits[5:3] to Register 0x487, Bits[5:3]. UEK, BDE, and NIT error monitoring can be selected for each lane by writing a 1 to the appropriate bit, as described in Table 45. These are enabled by default.
2. Enable the sync assertion mask for each type of error by writing to SYNC_ASSERT_MASK (Register 0x47D, Bits[2:0]) according to Table 31.
3. Program the desired error counter threshold into ERRORTHRES (Register 0x47C).
4. For each error type enabled in the SYNC_ASSERT_MASK register, if the error counter on any lane reaches the programmed threshold, $\overline{\text{SYNCOUT}}$± falls, issuing a sync request. Note that all error counts are reset when a link reinitialization occurs. The IRQ does not reset and must be reset manually.

**Table 31. Sync Assertion Mask (SYNC_ASSERT_MASK)**

| Addr. | Bit No. | Bit Name | Description |
|---|---|---|---|
| 0x47D | 2 | BDE | Set to 1 to assert $\overline{\text{SYNCOUT}}$± if the disparity error count reaches the threshold |
| | 1 | NIT | Set to 1 to assert $\overline{\text{SYNCOUT}}$± if the NIT error count reaches the threshold |
| | 0 | UEK | Set to 1 to assert $\overline{\text{SYNCOUT}}$± if the UEK character error count reaches the threshold |

### *CGS, Frame Sync, Checksum, and ILAS Monitoring*

Register 0x470 to Register 0x473 can be monitored to verify that each stage of the JESD204B link establishment has occurred.

Bit x of CODE_GRP_SYNC (Register 0x470) is high if Link Lane x received at least four K28.5 characters and passed code group synchronization.

Bit x of FRAME_SYNC (Register 0x471) is high if Link Lane x completed initial frame synchronization.

Bit x of GOOD_CHECKSUM (Register 0x472) is high if the checksum sent over the lane matches the sum of the JESD204B parameters sent over the lane during ILAS for Link Lane x. The parameters can be added either by summing the individual fields in registers or summing the packed register. If Register 0x300, Bit 6 = 0 (default), the calculated checksums are the lower eight bits of the sum of the following fields: DID, BID, LID, SCR, L − 1, F − 1, K − 1, M − 1, N − 1, SUBCLASSV, NP − 1, JESDV, S − 1, and HD. If Register 0x300, Bit 6 = 1, the calculated checksums are the lower eight bits of the sum of Register 0x400 to Register 0x40C and LID.

Bit x of INIT_LANE_SYNC (Register 0x473) is high if Link Lane x passed the initial lane alignment sequence.

### CGS, Frame Sync, Checksum, and ILAS IRQs

Fail signals for CGS, frame sync, checksum, and ILAS are available as IRQ events. Enable them by writing to Register 0x4B8, Bits[3:0]. The IRQ event status can be read at Register 0x4BA, Bits[3:0] after the IRQs are enabled. Write a 1 to Register 0x4BA, Bit 0 to reset the CGS IRQ. Write a 1 to Register 0x4BA, Bit 1 to reset the frame sync IRQ. Write a 1 to Register 0x4BA, Bit 2 to reset the checksum IRQ. Write a 1 to Register 0x4BA, Bit 3 to reset the ILAS IRQ.

See the Interrupt Request Operation section for more information.

### Configuration Mismatch IRQ

The AD9163 has a configuration mismatch flag that is available as an IRQ event. Use Register 0x4B9, Bit 0 to enable the mismatch flag (it is enabled by default), and then use Register 0x4BB, Bit 0 to read back its status and reset the IRQ signal. See the Interrupt Request Operation section for more information.

The configuration mismatch event flag is high when the link configuration settings (in Register 0x450 to Register 0x45D) do not match the JESD204B transmitted settings (Register 0x400 to Register 0x40D).

This function is different from the good checksum flags in Register 0x472. The good checksum flags ensure that the transmitted checksum matches a calculated checksum based on the transmitted settings. The configuration mismatch event ensures that the transmitted settings match the configured settings.

## HARDWARE CONSIDERATIONS

See the Applications Information section for information on hardware considerations.

# MAIN DIGITAL DATAPATH

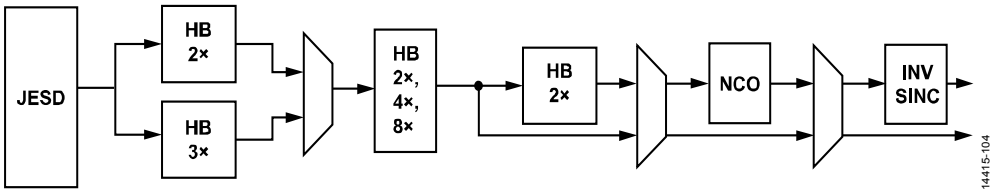


*Figure 107. Block Diagram of the Main Digital Datapath*

The block diagram in Figure 107 shows the functionality of the main digital datapath. The digital processing includes an input interpolation block with the choice of bypass (1×), 2×, or 3× interpolation, three additional 2× half-band interpolation filters, a final 2× NRZ mode interpolator filter, FIR85, that can be bypassed, and a quadrature modulator that consists of a 48-bit NCO and an inverse sinc block.

All of the interpolation filters accept in-phase (I) and quadrature (Q) data streams as a complex data stream. Similarly, the quadrature modulator and inverse sinc function also accept input data as a complex data stream. Thus, any use of the digital datapath functions requires the input data to be a complex data stream.

In bypass mode (1× interpolation), the input data stream is expected to be real data.

**Table 32. Pipeline Delay (Latency) for Various DAC Blocks**

| Mode | FIR85 On | Filter Bandwidth | Inverse Sinc | NCO | Pipeline Delay[1] ($f_{DAC}$ Clocks) |
|---|---|---|---|---|---|
| 6× | No | 80% | No | No | 332 |
| 8× | No | 80% | No | No | 602 |
| 12× | No | 80% | No | No | 674 |
| 16× | No | 80% | No | No | 1188 |
| 24× | No | 80% | No | No | 1272 |

[1] The pipeline delay given is a representative number, and may vary by a cycle or two based on the internal handoff timing conditions at startup.

The pipeline delay changes based on the digital datapath functions that are selected. See Table 32 for examples of the pipeline delay per block. These delays are in addition to the JESD204B latency.

## DATA FORMAT

The input data format for all modes on the AD9163 is 16-bit, twos complement. The digital datapath and the DAC decoder operate in twos complement format. The DAC is a current steering DAC and cannot represent 0—it must either source or sink current. As a result, when the 0 of twos complement is represented in the DAC, it is a +1, and all the positive values thereafter are shifted by +1. This mapping error introduces a ½ LSB shift in the DAC output. The leakage can become apparent when using the NCO to shift a signal that is above or below 0 Hz when synthesized. The NCO frequency is seen as a small spur at the NCO FTW.

To avoid the NCO frequency leakage, operate the DAC with a slight digital backoff of one or several codes, and then add 1 to all values in the data stream. These actions remove the NCO frequency leakage but cause a half LSB dc offset. This small dc offset is benign to the DAC and does not affect most applications because the DAC output is ac-coupled through dc blocking capacitors.

## INTERPOLATION FILTERS

The main digital path contains five half-band interpolation filters, plus a final half-band interpolation filter that is used in 2× NRZ mode. The filters are cascaded as shown in Figure 107.

The first pair of filters is a 2× (HB2) or 3× (HB3) filter. Each of these filters has two options for bandwidth, 80% or 90%. The 80% filters are lower power than the 90%. The filters default to the lower power 80% bandwidth. To select the filter bandwidth as 90%, program the FILT_BW bit in the DATAPATH_CFG register to 1 (Register 0x111, Bit 4 = 0b1).

Following the first pair of filters is a series of 2× half-band filters, each of which halves the usable bandwidth of the previous one. HB4 has 45%, HB5 has 22.5%, and HB6 has 11.25% of the $f_{DATA}$ bandwidth.

The final half-band filter, FIR85, is used in the 2× NRZ mode. It is clocked at the 2 × $f_{DAC}$ rate and has a usable bandwidth of 45% of the $f_{DAC}$ rate. The FIR85 filter is a complex filter, and therefore the bandwidth is centered at 0 Hz. The FIR85 filter is used in conjunction with the complex interpolation modes to push the DAC update rate higher and move images further from the desired signal.

Table 33 shows how to select each available interpolation mode, their usable bandwidths, and their maximum data rates. Calculate the available signal bandwidth as the interpolator filter bandwidth, $BW_{FILT}$, multiplied by $f_{DAC}$/InterpolationFactor, as follows:

$$BW_{SIGNAL} = BW_{FILT} \times (f_{DAC}/InterpolationFactor)$$

### *Filter Performance*

The interpolation filters interpolate between existing data in such a way that they minimize changes in the incoming data while suppressing the creation of interpolation images. This datapath is shown for each filter in Figure 108.

The usable bandwidth (as shown in Table 33) is defined as the frequency band over which the filters have a pass-band ripple of less than ±0.001 dB and an image rejection of greater than 85 dB. A conceptual drawing that shows the relative bandwidth of each of the filters is shown in Figure 108. The maximum pass band amplitude of all filters is the same; they are different in the illustration to improve understanding.

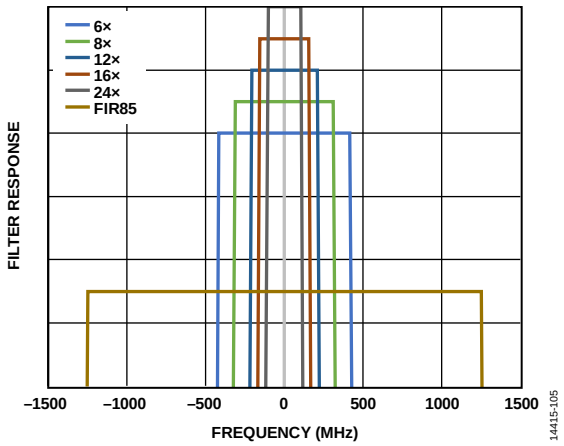


*Figure 108. All Band Responses of Interpolation Filters*

### Filter Performance Beyond Specified Bandwidth

Some of the interpolation filters are specified to 0.4 × $f_{DATA}$ (with a pass band). The filters can be used slightly beyond this ratio at the expense of increased pass-band ripple and decreased interpolation image rejection.

**Table 33. Interpolation Modes and Usable Bandwidth**

| Interpolation Mode | INTERP_MODE, Register 0x110, Bits[3:0] | Available Signal Bandwidth (BW)[1] | Maximum $f_{DATA}$ (MHz) |
|---|---|---|---|
| 6× | 0x04 | BW × $f_{DATA}$/2 | $f_{DAC}$/6 |
| 8× | 0x05 | BW × $f_{DATA}$/2 | $f_{DAC}$/8 |
| 12× | 0x06 | BW × $f_{DATA}$/2 | $f_{DAC}$/12 |
| 16× | 0x07 | BW × $f_{DATA}$/2 | $f_{DAC}$/16 |
| 24× | 0x08 | BW × $f_{DATA}$/2 | $f_{DAC}$/24 |
| 2× NRZ (Register 0x111, Bit 0 = 1) | Any combination[2] | 0.45 × $f_{DAC}$[3] | $f_{DAC}$ (real) or $f_{DAC}$/2 (complex)[3] |

[1] The data rate ($f_{DATA}$) for all interpolator modes is a complex data rate, meaning each of I data and Q data run at that rate. Available signal bandwidth is the data rate multiplied by the bandwidth of the initial 2× or 3× interpolator filters, which can be set to BW = 80% or BW = 90%. This bandwidth is centered at 0 Hz.
[2] The 2× NRZ filter, FIR85, can be used with any of the interpolator combinations.
[3] The bandwidth of the FIR85 filter is centered at 0 Hz.

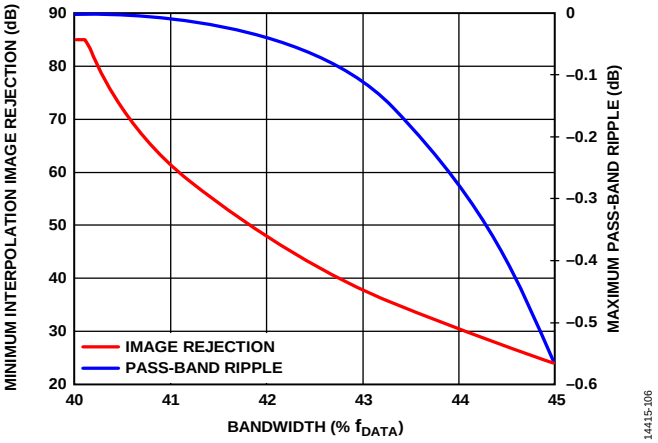


*Figure 109. Interpolation Filter Performance Beyond Specified Bandwidth for the 80% Filters*

Figure 109 shows the performance of the interpolation filters beyond 0.4 × $f_{DATA}$. The ripple increases much slower than the image rejection decreases. This means that if the application can tolerate degraded image rejection from the interpolation filters, more bandwidth can be used.

Most of the filters are specified to 0.45 × $f_{DATA}$ (with pass band). Figure 110 to Figure 117 show the filter response for each of the interpolator filters on the AD9163.

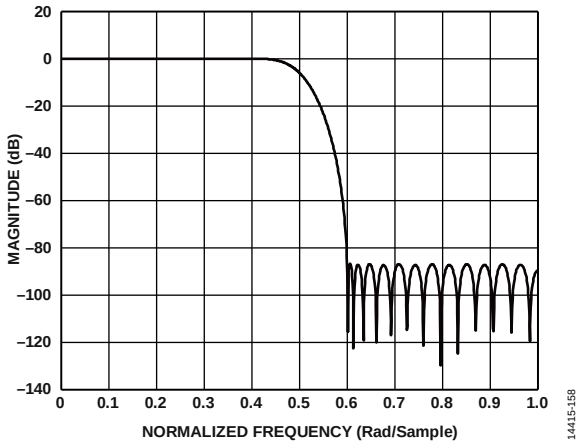


*Figure 110. First 2× Half-Band 80% Filter Response*

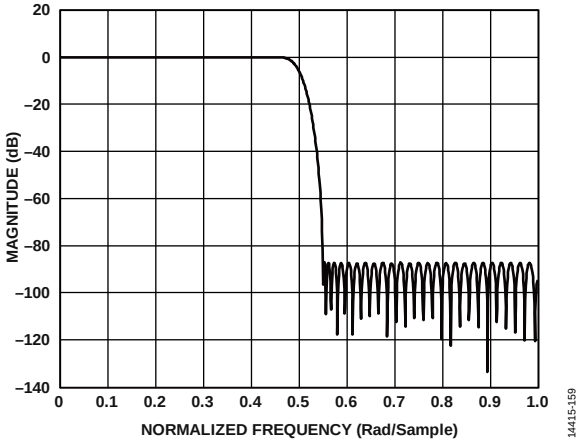


*Figure 111. First 2× Half-Band 90% Filter Response*

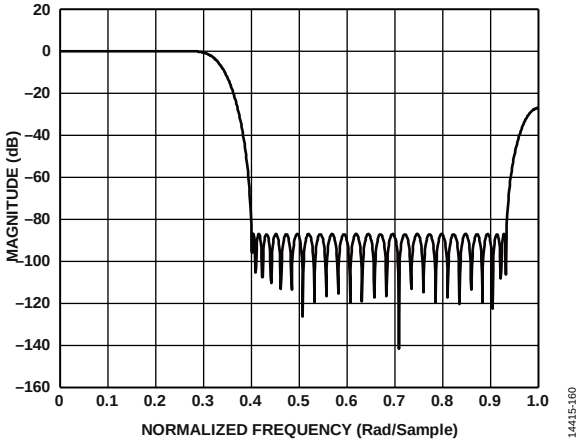


*Figure 112. 3× Third-Band 80% Filter Response*

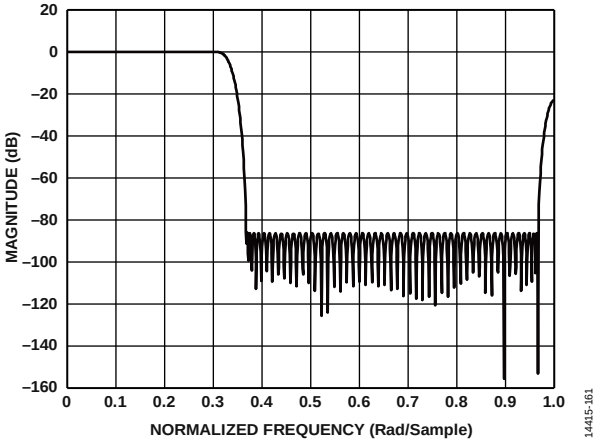


*Figure 113. 3× Third-Band 90% Filter Response*

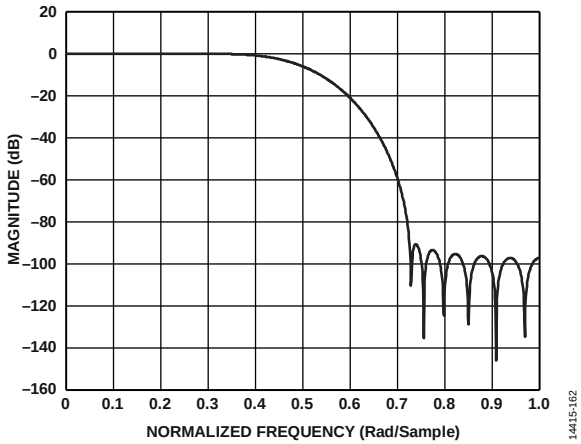


*Figure 114. Second 2× Half-Band 45% Filter Response*

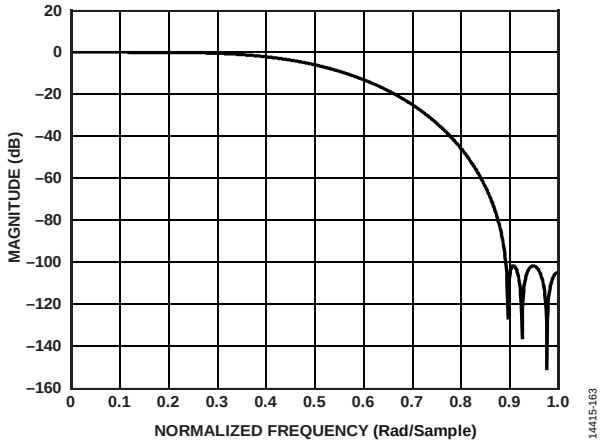


Figure 115. Third 2× Half-Band 22.5% Filter Response

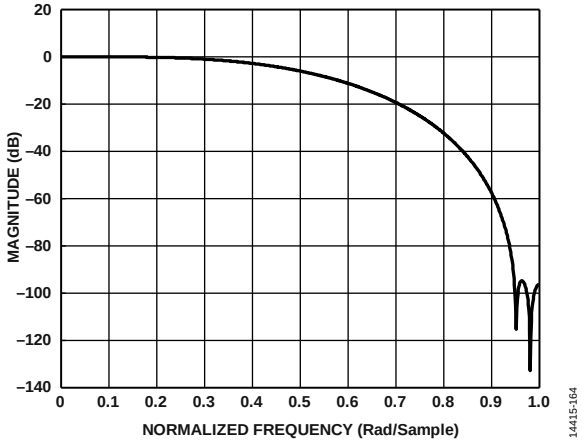


Figure 116. Fourth 2× Half-Band 11.25% Filter Response

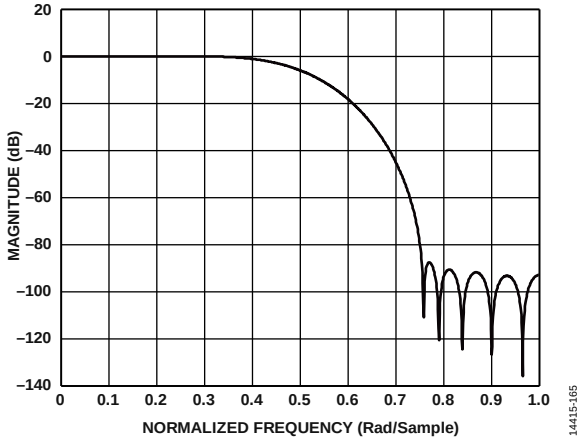


Figure 117. FIR85 2× Half-Band 45% Filter Response

## DIGITAL MODULATION

The AD9163 has digital modulation features to modulate the baseband quadrature signal to the desired DAC output frequency.

The AD9163 is equipped with several NCO modes. The default NCO is a 48-bit, integer NCO. The A/B ratio of the dual modulus NCO allows the output frequency to be synthesized with very fine precision. NCO mode is selected as shown in Table 34.

**Table 34. Modulation Mode Selection**

| | Modulation Type | |
|---|---|---|
| **Modulation Mode** | **Register 0x111, Bit 6** | **Register 0x111, Bit 2** |
| None | 0b0 | 0b0 |
| 48-Bit Integer NCO | 0b1 | 0b0 |
| 48-Bit Dual Modulus NCO | 0b1 | 0b1 |

### *48-Bit Dual Modulus NCO*

This modulation mode uses an NCO, a phase shifter, and a complex modulator to modulate the signal by a programmable carrier signal as shown in Figure 118. This configuration allows output signals to be placed anywhere in the output spectrum with very fine frequency resolution.

The NCO produces a quadrature carrier to translate the input signal to a new center frequency. A quadrature carrier is a pair of sinusoidal waveforms of the same frequency, offset 90° from each other. The frequency of the quadrature carrier is set via a FTW. The quadrature carrier is mixed with the I and Q data and then summed into the I and Q datapaths, as shown in Figure 118.

#### Integer NCO Mode

The main 48-bit NCO can be used as an integer NCO by using the following formula to create the frequency tuning word (FTW):

$$-f_{DAC}/2 \le f_{CARRIER} < +f_{DAC}/2$$

$$FTW = (f_{CARRIER}/f_{DAC}) \times 2^{48}$$

where *FTW* is a 48-bit, twos complement number.

When in 2× NRZ mode (FIR85 enabled with Register 0x111, Bit 0 = 1), the frequency tuning word is calculated as

$$0 \le f_{CARRIER} < f_{DAC}$$

$$FTW = (f_{CARRIER}/f_{DAC}) \times 2^{48}$$

where *FTW* is a 48-bit binary number.

This method of calculation causes $f_{CARRIER}$ values in the second Nyquist zone to appear to move to $f_{DAC} - f_{CARRIER}$ when flipping the FIR85 enable bit and not changing the FTW to account for the change in number format.

The intended effect is that a sweep of the NCO from 0 Hz to $f_{DAC} - f_{DAC}/2^{48}$ appears seamless when the FIR85 enable bit is set to Register 0x111, Bit 0 = 0b1 prior to $f_{CARRIER}/f_{DAC} = 0.5$. As can be seen from examination, the FTWs from 0 to less than $f_{DAC}/2$ mean the same in either case, but they mean different $f_{CARRIER}$ values from $f_{DAC}/2$ to $f_{DAC} - f_{DAC}/2^{48}$. This effect must be considered when constructing FTW values and using the 2× NRZ mode.

The frequency tuning word is set as shown in Table 35.

**Table 35. NCO FTW Registers**

| Address | Value | Description |
|---|---|---|
| 0x114 | FTW[7:0] | 8 LSBs of FTW |
| 0x115 | FTW[15:8] | Next 8 bits of FTW |
| 0x116 | FTW[23:16] | Next 8 bits of FTW |
| 0x117 | FTW[31:24] | Next 8 bits of FTW |
| 0x118 | FTW[39:32] | Next 8 bits of FTW |
| 0x119 | FTW[47:40] | 8 MSBs of FTW |

Unlike other registers, the FTW registers are not updated immediately upon writing. Instead, the FTW registers update on the rising edge of FTW_LOAD_REQ (Register 0x113, Bit 0). After an update request, FTW_LOAD_ACK (Register 0x113, Bit 1) must be high to acknowledge that the FTW has updated.

The SEL_SIDEBAND bit (Register 0x111, Bit 1 = 0b1) is a convenience bit that can be set to use the lower sideband modulation result, which is equivalent to flipping the sign of the FTW.

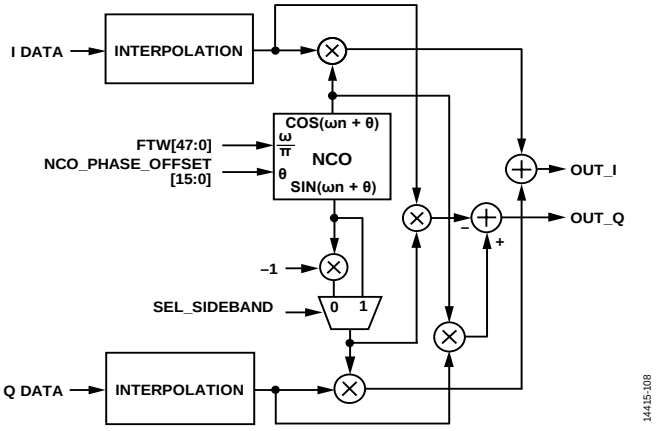


*Figure 118. NCO Modulator Block Diagram*

### Modulus NCO Mode

The main 48-bit NCO can also be used in a dual modulus mode to create fractional frequencies beyond the 48-bit accuracy. The modulus mode is enabled by programming the MODULUS_EN bit in the DATAPATH_CFG register to 1 (Register 0x111, Bit 2 = 0b1).

The frequency ratio for the programmable modulus direct digital synthesis (DDS) is very similar to that of the typical accumulator-based DDS. The only difference is that N is not required to be a power of two for the programmable modulus, but can be an arbitrary integer. In practice, hardware constraints place limits on the range of values for N. As a result, the modulus extends the use of the NCO to applications that require exact rational frequency synthesis. The underlying function of the programmable modulus technique is to alter the accumulator modulus.

Implementation of the programmable modulus function within the AD9163 is such that the fraction, M/N, is expressible per Equation 1. Note that the form of the equation implies a compound frequency tuning word with X representing the integer part and A/B representing the fractional part.

$$\frac{f_{CARRIER}}{f_{DAC}} = \frac{M}{N} = \frac{X + \frac{A}{B}}{2^{48}} \quad (1)$$

where:
*X* is programmed in Register 0x114 to Register 0x119.
*A* is programmed in Register 0x12A to Register 0x12F.
*B* is programmed in Register 0x124 to Register 0x129.

### Programmable Modulus Example

Consider the case in which $f_{DAC}$ = 2500 MHz and the desired value of $f_{CARRIER}$ is 250 MHz. This scenario synthesizes an output frequency that is not a power of two submultiple of the sample rate, namely $f_{CARRIER} = (1/10)\ f_{DAC}$, which is not possible with a typical accumulator-based DDS. The frequency ratio, $f_{CARRIER}/f_{DAC}$, leads directly to M and N, which are determined by reducing the fraction (250,000,000/2,500,000,000) to its lowest terms, that is,

$$M/N = 250{,}000{,}000/2{,}500{,}000{,}000 = 1/10$$

Therefore, M = 1 and N = 10.

After calculation, X = 28147497671065, A = 3, and B = 5. Programming these values into the registers for X, A, and B (X is programmed in Register 0x114 to Register 0x119, B is programmed in Register 0x124 to Register 0x129, and A is programmed in Register 0x12A to Register 0x12F) causes the NCO to produce an output frequency of exactly 250 MHz given a 2500 MHz sampling clock. For more details, refer to the AN-953 Application Note on the Analog Devices, Inc., website.

### NCO Reset

Resetting the NCO can be useful when determining the start time and phase of the NCO. The NCO can be reset by several different methods, including a SPI write, using the TX_ENABLE pin, or by the SYSREF± signal. Due to internal timing variations from device to device, these methods achieve an accuracy of ±6 DAC clock cycles.

Program Register 0x800, Bits[7:6] to 0b01 to set the NCO in phase discontinuous switching mode via a write to the SPI port. Then, any time the frequency tuning word is updated, the NCO phase accumulator resets and the NCO begins counting at the new FTW.

### *Changing the NCO Frequency*

In the 48-bit NCO, the mode of updating the frequency tuning word can be changed from requiring a write to the FTW_LOAD_REQ bit (Register 0x113, Bit 0) to an automatic update mode. In the automatic update mode, the FTW is updated as soon as the chosen FTW word is written.

To set the automatic FTW update mode, write the appropriate word to the FTW_REQ_MODE bits (Register 0x113, Bits[6:4]), choosing the particular FTW word that causes the automatic update. For example, if relatively coarse frequency steps are needed, it may be sufficient to write a single word to the MSB byte of the FTW, and therefore the FTW_REQ_MODE bits can be programmed to 110 (Register 0x113, Bits[6:4] = 0b110). Then, each time the most significant byte, FTW5, is written, the NCO FTW is automatically updated.

The FTW_REQ_MODE bits can be configured to use any of the FTW words as the automatic update trigger word. This configuration provides convenience when choosing the order in which to program the FTW registers.

The speed of the SPI port write function is guaranteed, and is a minimum of 100 MHz (see Table 4). Thus, the NCO FTW can be updated in as little as 240 ns with a one register write in automatic update mode.

The manner in which the NCO transitions to the new frequency is determined by the frequency change mode selection.

The NCO supports several modes of changing frequency: phase continuous or phase discontinuous. The modes are given in Table 36.

**Table 36. NCO Frequency Change Mode**

| Register 0x800, Bits[7:6] | Description |
|---|---|
| 0b00 | Phase continuous switch |
| 0b01 | Phase discontinuous switch (reset NCO accumulator) |

In phase continuous switching, the frequency tuning word of the NCO is updated and the phase accumulator continues to accumulate to the new frequency. In phase discontinuous mode, the FTW of the NCO is updated and the phase accumulator is reset, making an instantaneous jump to the new frequency.

## INVERSE SINC

The AD9163 provides a digital inverse sinc filter to compensate the DAC roll-off over frequency. The filter is enabled by setting the INVSINC_EN bit (Register 0x111, Bit 7) and is disabled by default.

The inverse sinc ($sinc^{-1}$) filter is a seven-tap FIR filter. Figure 119 shows the frequency response of sin(x)/x roll-off, the inverse sinc filter, and the composite response. The composite response has less than ±0.05 dB pass-band ripple up to a frequency of $0.4 \times f_{DACCLK}$. When 2× NRZ mode is enabled, the inverse sinc filter operates to $0.4 \times f_{2\times DACCLK}$.

To provide the necessary peaking at the upper end of the pass band, the inverse sinc filter shown has an intrinsic insertion loss of about 3.8 dB.

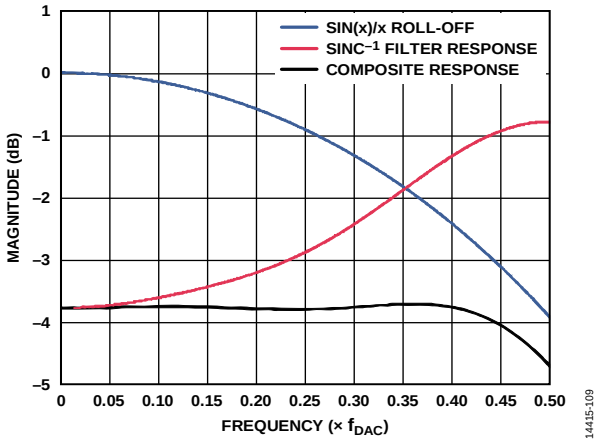


*Figure 119. Responses of Sin(x)/x Roll-Off, the $Sinc^{-1}$ Filter, and the Composite of the Two*

## DOWNSTREAM PROTECTION

The AD9163 has several features designed to protect the power amplifier (PA) of the system, as well as other downstream blocks. They consist of a control signal from the LMFC sync logic and a transmit enable function. The protection mechanism in each case is the blanking of data that is passed to the DAC decoder. The differences lie in the location in the datapath and slight variations of functionality.

The JESD204B serial link has several flags and quality measures to indicate the serial link is up and running error free. If any of these measures flags an issue, a signal from the LMFC sync logic is sent to a mux that stops data from flowing to the DAC decoder and replaces it with 0s.

There are several transmit enable features, including a TX_ENABLE register that can be used to squelch data at several points in the datapath or configure the TX_ENABLE pin to do likewise.

### *Transmit Enable*

The transmit enable feature can be configured either as a SPI controlled function or a pin controlled function. It can be used for several different purposes. The SPI controlled function has less accurate timing due to its reliance on a microcontroller to program it; therefore, it is typically used as a preventative measure at power-up or when configuring the device.

The SPI controlled TX_ENABLE function can be used to zero the input to the digital datapath or to zero the output from the digital datapath, as shown in Figure 120. If the input to the digital datapath is zeroed, any filtering that is selected filters the 0 signal, causing a gradual ramp-down of energy in the digital datapath. If the digital datapath is bypassed, as in 1× mode, the data at the input to the DAC immediately drops to zero.

The TX_ENABLE pin can be used for more accurate timing when enabling or disabling the DAC output. The effect of the TX_ENABLE pin can be configured by the same TX_ENABLE register (Register 0x03F) as is used for the SPI controlled functions, and it can be made to have the same effects as the SPI controlled function, namely to zero the input to the digital datapath or to zero the output from the digital datapath. In addition, the TX_ENABLE pin can also be configured to ramp down (or up) the full-scale current of the DAC. The ramp down reduces the output power of the DAC by about 20 dB from full scale to the minimum output current.

The TX_ENABLE pin can also be programmed to reset the NCO phase accumulator. See Table 37 for a description of the settings available for the TX_ENABLE function.

**Table 37. TX_ENABLE Settings**

| Register 0x03F | Setting | Description |
|---|---|---|
| Bit 7 | 0 | SPI control: zero data to the DAC |
| | 1 | SPI control: allow data to pass to the DAC |
| Bit 6 | 0 | SPI control: zero data at input to the datapath |
| | 1 | SPI control: allow data to enter the datapath |
| Bits[5:4] | N/A[1] | Reserved |
| Bit 3 | 0 | Use SPI writes to reset the NCO[2] |
| | 1 | Use TX_ENABLE to reset the NCO |
| Bit 2 | 0 | Use SPI control to zero data to the DAC |
| | 1 | Use TX_ENABLE pin to zero data to the DAC |
| Bit 1 | 0 | Use SPI control to zero data at the input to the datapath |
| | 1 | Use TX_ENABLE pin to zero data at input to the datapath |
| Bit 0 | 0 | Use SPI registers to control the full-scale current |
| | 1 | Use TX_ENABLE pin to control the full-scale current |

[1] N/A means not applicable.
[2] Use SPI writes to reset the NCO if resetting the NCO is desired. Register 0x800, Bits[7:6] determine whether the NCO is reset. See Table 36 for more details.

## DATAPATH PRBS

The datapath PRBS can verify the AD9163 datapath receives and correctly decodes data. The datapath PRBS verifies the JESD204B parameters of the transmitter and receiver match, the lanes of the receiver are mapped appropriately, the lanes are appropriately inverted, and, if necessary, the start-up routine is correctly implemented.

To run the datapath PRBS test, complete the following steps:

1. Set up the device in the desired operating mode using the start-up sequence.
2. Send PRBS7 or PRBS15 data.
3. Write Register 0x14, Bit 2 = 0 for PRBS7 or 1 for PRBS15.
4. Write Register 0x14B, Bits[1:0] = 0b11 to enable and reset the PRBS test.
5. Write Register 0x14B, Bits[1:0] = 0b01 to enable the PRBS test and release reset.
6. Wait 500 ms.
7. Check the status of the PRBS by checking the IRQ for the I and Q path PRBS as described in the Datapath PRBS IRQ section.
8. Read Register 0x14B, Bits [7:6]. Bit 6 is 0 if the I channel has any errors. Bit 7 is 0 if the Q channel has any errors.
9. Read Register 0x14C to read the error count for the I channel.
10. Read Register 0x14D to read the error count for the Q channel. The PRBS processes 32 bits at a time, and compares the 32 new bits to the previous set of 32 bits. It detects and reports only 1 error in every group of 32 bits; therefore, the error count partly depends on when the errors are seen.

For example, see the following sequence:

- Bits: 32 good, 31 good, 1 bad; 32 good [2 errors]
- Bits: 32 good, 22 good, 10 bad; 32 good [2 errors]
- Bits: 32 good, 31 good, 1 bad; 31 good, 1 bad; 32 good [3 errors]

## DATAPATH PRBS IRQ

The PRBS fail signals for the I and Q path are available as IRQ events. Use Register 0x020, Bits [1:0] to enable the fail signals, and then use Register 0x024, Bits [1:0] to read back the status and reset the IRQ signals. See the Interrupt Request Operation section for more information.

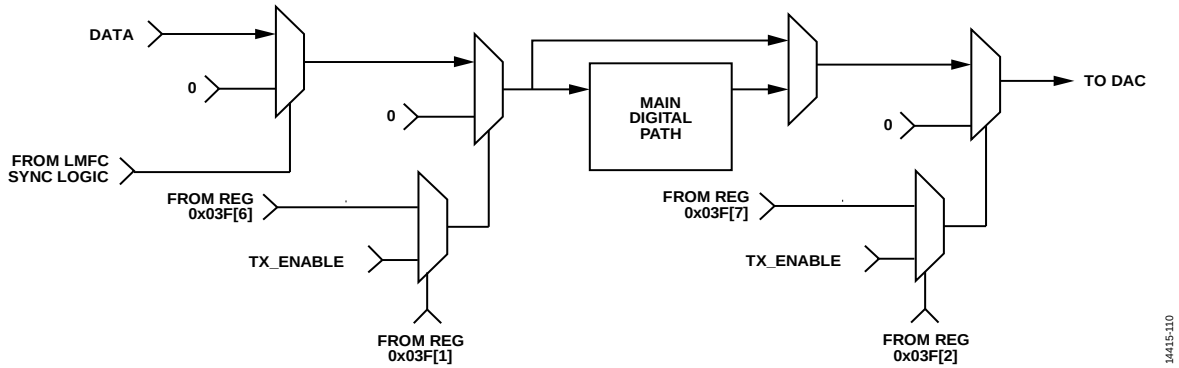


*Figure 120. Downstream Protection Block Diagram*

# INTERRUPT REQUEST OPERATION

The AD9163 provides an interrupt request output signal ($\overline{IRQ}$) on Ball G4 that can be used to notify an external host processor of significant device events. On assertion of the interrupt, query the device to determine the precise event that occurred. The $\overline{IRQ}$ pin is an open-drain, active low output. Pull the $\overline{IRQ}$ 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.

Figure 121 shows a simplified block diagram of how the IRQ blocks work. If IRQ_EN is low, the INTERRUPT_SOURCE signal is set to 0. If IRQ_EN is high, any rising edge of EVENT causes the INTERRUPT_SOURCE signal to be set high. If any INTERRUPT_SOURCE signal is high, the $\overline{IRQ}$ pin is pulled low. INTERRUPT_SOURCE can be reset to 0 by either an IRQ_RESET signal or a DEVICE_RESET signal.

Depending on the STATUS_MODE signal, the EVENT_STATUS bit reads back an event signal or INTERRUPT_SOURCE signal. The AD9163 has several interrupt register blocks (IRQ) that can monitor up to 75 events (depending on device configuration). Certain details vary by IRQ register block as described in Table 38. Table 39 shows the source registers of the IRQ_EN, IRQ_RESET, and STATUS_MODE signals in Figure 121, as well as the address where EVENT_STATUS is read back.

**Table 38. IRQ Register Block Details**

| Register Block | Event Reported | EVENT_STATUS |
|---|---|---|
| 0x020, 0x024 | Per chip | INTERRUPT_SOURCE if $\overline{IRQ}$ is enabled; if not, it is the event signal |
| 0x4B8 to 0x4BB; 0x470 to 0x473 | Per link and lane | INTERRUPT_SOURCE if $\overline{IRQ}$ is enabled; if not, 0 |

## 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 $\overline{IRQ}$ activation, run the following routine to clear an interrupt request:

1. Read the status of the event flag bits that are being monitored.
2. Disable the interrupt by writing 0 to IRQ_EN.
3. Read the event source.
4. Perform any actions that may be required to clear the cause of the event. In many cases, no specific actions may be required.
5. Verify that the event source is functioning as expected.
6. Clear the interrupt by writing 1 to IRQ_RESET.
7. Enable the interrupt by writing 1 to IRQ_EN.

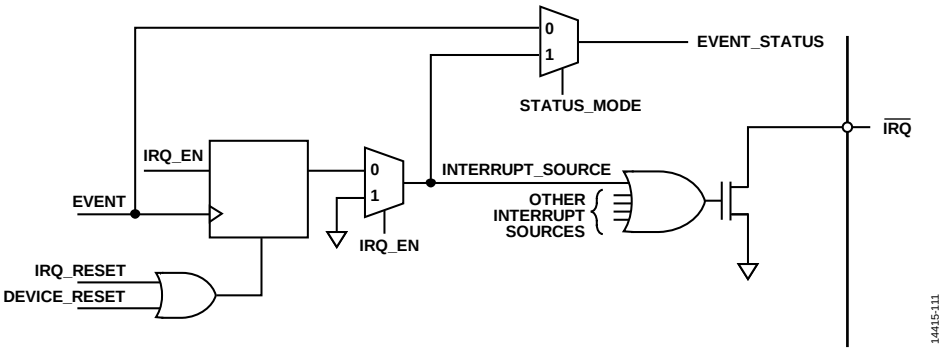


*Figure 121. Simplified Schematic of $\overline{IRQ}$ Circuitry*

**Table 39. IRQ Register Block Address of IRQ Signal Details**

| | Address of IRQ Signals[1] | | | |
|---|---|---|---|---|
| Register Block | IRQ_EN | IRQ_RESET | STATUS_MODE[2] | EVENT_STATUS |
| 0x020, 0x024 | 0x020; R/W per chip | 0x024; W per chip | STATUS_MODE = IRQ_EN | 0x024; R per chip |
| 0x4B8 to 0x4BB | 0x4B8, 0x4B9; W per error type | 0x4BA, 0x4BB; W per error type | N/A, STATUS_MODE = 1 | 0x4BA, 0x4BB; R per chip |
| 0x470 to 0x473 | 0x470 to 0x473; W per error type | 0x470 to 0x473; W per link | N/A, STATUS_MODE = 1 | 0x470 to 0x473; R per link |

[1] R is read; W is write; and R/W is read/write.
[2] N/A means not applicable.

# APPLICATIONS INFORMATION

## HARDWARE CONSIDERATIONS

### *Power Supply Recommendations*

All the AD9163 supply domains must remain as noise free as possible for the best operation. Power supply noise has a frequency component that affects performance, and is specified in volts rms terms.

An LC filter on the output of the power supply is recommended to attenuate the noise, and must be placed as close to the AD9163 as possible. The VDD12_CLK supply is the most noise sensitive supply on the device, followed by the VDD25_DAC and VNEG_N1P2 supplies, which are the DAC output rails. It is highly recommended that the VDD12_CLK be supplied by itself with an ultralow noise regulator such as the ADM7154 or ADP1761 to achieve the best phase noise performance possible. Noisier regulators impose phase noise onto the DAC output.

The VDD12A supply can be connected to the digital DVDD supply with a separate filter network. All of the SERDES 1.2 V supplies can be connected to one regulator with separate filter networks. The IOVDD supply can be connected to the VDD25_DAC supply with a separate filter network, or can be powered from a system controller (for example, a microcontroller), 1.8 V to 3.3 V supply. The power supply sequencing requirement must be met; therefore, a switch or other solution must be used when connected to the IOVDD supply with VDD25_DAC.

Take note of the maximum power consumption numbers given in Table 3 to ensure the power supply design can tolerate temperature and IC process variation extremes. The amount of current drawn is dependent on the chosen use cases, and specifications are provided for several use cases to illustrate examples and contributions from individual blocks, and to assist in calculating the maximum required current per supply.

Another consideration for the power supply design is peak current handling capability. The AD9163 draws more current in the main digital supply when synthesizing a signal with significant amplitude variations, such as a modulated signal, as compared to when in idle mode or synthesizing a dc signal. Therefore, the power supply must be able to supply current quickly to accommodate burst signals such as GSM, TDMA, or other signals that have an on/off time domain response. Because the amount of current variation depends on the signals used, it is best to perform lab testing first to establish ranges. A typical difference can be several hundred milliamperes.

#### Power Sequencing

The AD9163 requires power sequencing to avoid damage to the DAC. A board design with the AD9163 must include a power sequencer chip, such as the ADM1184, to ensure that the domains power up in the correct order. The ADM1184 monitors the level of power domains upon power-up. It sends an enable signal to the next grouping of power domains. When all power domains are powered up, a power-good signal is sent to the system controller to indicate all power supplies are powered up.

The IOVDD, VDD12A, VDD12_CLK, and DVDD domains must be powered up first. Then, the VNEG_N1P2, VDD_1P2, PLL_CLK_VDD12, DVDD_1P2, and SYNC_VDD_3P3 can be powered up. The VDD25_DAC domain must be powered up last. There is no requirement for a power-down sequence.

#### Power and Ground Planes

Solid ground planes are recommended to avoid ground loops and to provide a solid, uninterrupted ground reference for the high speed transmission lines that require controlled impedances. It is recommended that power planes be stacked between ground layers for high frequency filtering. Doing so adds extra filtering and isolation between power supply domains in addition to the decoupling capacitors.

Do not use segmented power planes as a reference for controlled impedances unless the entire length of the controlled impedance trace traverses across only a single segmented plane. These and additional guidelines for the topology of high speed transmission lines are described in the JESD204B Serial Interface Inputs (SERDIN0± to SERDIN7±) section.

For some applications, where highest performance and higher output frequencies are required, the choice of PCB materials significantly impacts results. For example, materials such as polyimide or materials from the Rogers Corporation can be used, for example, to improve tolerance to high temperatures and improve performance. Rogers 4350 material is used for the top three layers in some of the evaluation board designs: between the top signal layer and the ground layer below it, between the ground layer and an internal signal layer, and between that signal layer and another ground layer.

### *JESD204B Serial Interface Inputs (SERDIN0± to SERDIN7±)*

When considering the layout of the JESD204B serial interface transmission lines, there are many factors to consider to maintain optimal link performance. Among these factors are insertion loss, return loss, signal skew, and the topology of the differential traces.

### Insertion Loss

The JESD204B specification limits the amount of insertion loss allowed in the transmission channel (see Figure 93). The AD9163 equalization circuitry allows significantly more loss in the channel than is required by the JESD204B specification. It is still important that the designer of the PCB minimize the amount of insertion loss by adhering to the following guidelines:

- Keep the differential traces short by placing the AD9163 as near the transmitting logic device as possible and routing the trace as directly as possible between the devices.
- Route the differential pairs on a single plane using a solid ground plane as a reference. It is recommended to route the SERDES lanes on the same layer as the AD9163 to avoid vias being used in the SERDES lanes.
- Use a PCB material with a low dielectric constant (<4) to minimize loss, if possible.

When choosing between the stripline and microstrip techniques, keep in mind the following considerations: stripline has less loss (see Figure 95 and Figure 96) and emits less EMI, but requires the use of vias that can add complexity to the task of controlling the impedance; whereas microstrip is easier to implement (if the component placement and density allow routing on the top layer) and eases the task of controlling the impedance.

If using the top layer of the PCB is problematic or the advantages of stripline are desirable, follow these recommendations:

- Minimize the number of vias.
- If possible, use blind vias to eliminate via stub effects and use microvias to minimize via inductance.
- If using standard vias, use the maximum via length to minimize the stub size. For example, on an 8-layer board, use Layer 7 for the stripline pair (see Figure 122).
- For each via pair, place a pair of ground vias adjacent to them to minimize the impedance discontinuity (see Figure 122).




*Figure 122. Minimizing Stub Effect and Adding Ground Vias for Differential Stripline Traces*

### Return Loss

The JESD204B specification limits the amount of return loss allowed in a converter device and a logic device, but does not specify return loss for the channel. However, every effort must be made to maintain a continuous impedance on the transmission line between the transmitting logic device and the AD9163. Minimizing the use of vias, or eliminating them all together, reduces one of the primary sources for impedance mismatches on a transmission line (see the Insertion Loss section). Maintain a solid reference beneath (for microstrip) or above and below (for stripline) the differential traces to ensure continuity in the impedance of the transmission line. If the stripline technique is used, follow the guidelines listed in the Insertion Loss section to minimize impedance mismatches and stub effects.

Another primary source for impedance mismatch is at either end of the transmission line, where care must be taken to match the impedance of the termination to that of the transmission line. The AD9163 handles this internally with a calibrated termination scheme for the receiving end of the line. See the Interface Power-Up and Input Termination section for details on this circuit and the calibration routine.

### Signal Skew

There are many sources for signal skew, but the two sources to consider when laying out a PCB are interconnect skew within a single JESD204B link and skew between multiple JESD204B links. In each case, keeping the channel lengths matched to within 12.5 mm is adequate for operating the JESD204B link at speeds of up to 12.5 Gbps. This amount of channel length match is equivalent to about 85% UI on the AD9163 evaluation board. Managing the interconnect skew within a single link is fairly straightforward. Managing multiple links across multiple devices is more complex. However, follow the 12.5 mm guideline for length matching. The AD9163 can handle more skew than the 85% UI due to the 6 PCLK buffer in the JESD204B receiver, but matching the channel lengths as close as possible is still recommended.

### Topology

Structure the differential SERDINx± pairs to achieve 50 Ω to ground for each half of the pair. Stripline vs. microstrip trade-offs are described in the Insertion Loss section. In either case, it is important to keep these transmission lines separated from potential noise sources such as high speed digital signals and noisy supplies. If using stripline differential traces, route them using a coplanar method, with both traces on the same layer. Although this method does not offer more noise immunity than the broadside routing method (traces routed on adjacent layers), it is easier to route and manufacture so that the impedance continuity is maintained. An illustration of broadside vs. coplanar is shown in Figure 123.

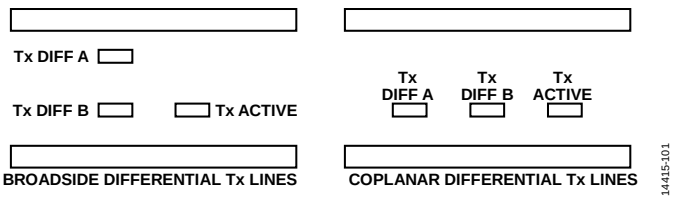


*Figure 123. Broadside vs. Coplanar Differential Stripline Routing Techniques*

When considering the trace width vs. copper weight and thickness, the speed of the interface must be considered. At multigigabit speeds, the skin effect of the conducting material confines the current flow to the surface. Maximize the surface area of the conductor by making the trace width made wider to reduce the losses. Additionally, loosely couple differential traces to accommodate the wider trace widths. This coupling helps reduce the crosstalk and minimize the impedance mismatch when the traces must separate to accommodate components, vias, connectors, or other routing obstacles. Tightly coupled vs. loosely coupled differential traces are shown in Figure 124.

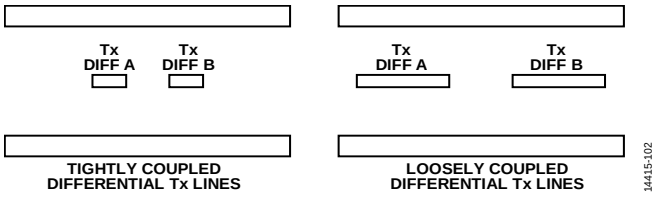


*Figure 124. Tightly Coupled vs. Loosely Coupled Differential Traces*

### AC Coupling Capacitors

The AD9163 requires that the JESD204B input signals be ac-coupled to the source. These capacitors must be 100 nF and placed as close as possible to the transmitting logic device.

To minimize the impedance mismatch at the pads, select the package size of the capacitor so that the pad size on the PCB matches the trace width as closely as possible.

### *$\overline{\text{SYNCOUT}}$±, SYSREF±, and CLK± Signals*

The $\overline{\text{SYNCOUT}}$± and SYSREF± signals on the AD9163 is low speed LVDS differential signals. Use controlled impedance traces routed with 100 Ω differential impedance and 50 Ω to ground when routing these signals. As with the SERDIN0± to SERDIN7± data pairs, it is important to keep these signals separated from potential noise sources such as high speed digital signals and noisy supplies.

Separate the $\overline{\text{SYNCOUT}}$± signal from other noisy signals, because noise on the $\overline{\text{SYNCOUT}}$± may be interpreted as a request for /K/ characters.

It is important to keep similar trace lengths for the CLK± and SYSREF± signals from the clock source to each of the devices on either end of the JESD204B links (see Figure 125). If using a clock chip that can tightly control the phase of CLK± and SYSREF±, the trace length matching requirements are greatly reduced.

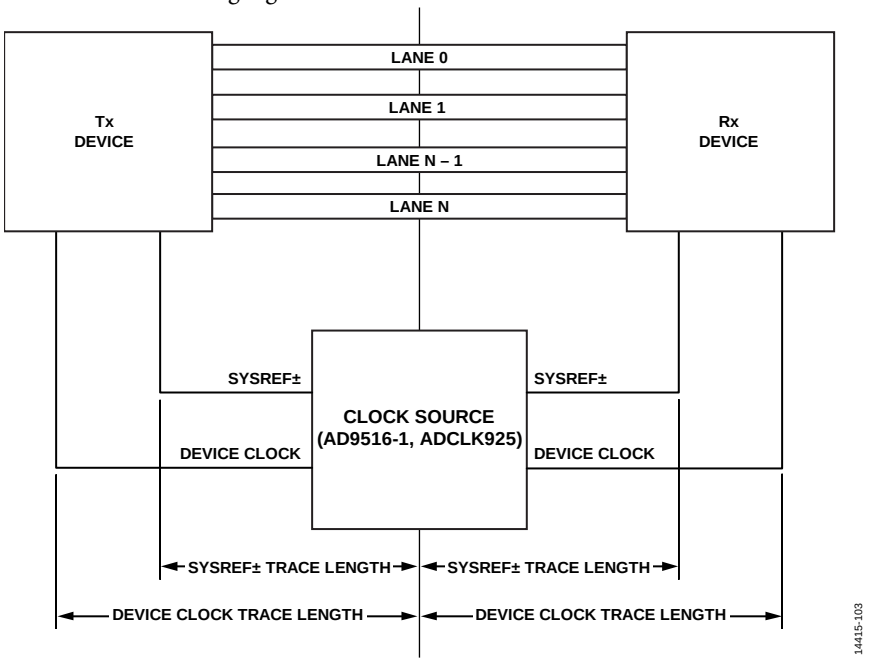


*Figure 125. SYSREF± Signal and Device Clock Trace Length*

# ANALOG INTERFACE CONSIDERATIONS

## ANALOG MODES OF OPERATION

The AD9163 uses the quad-switch architecture shown in Figure 126. Only one pair of switches is enabled during a half-clock cycle, thus requiring each pair to be clocked on alternative clock edges. A key benefit of the quad-switch architecture is that it masks the code dependent glitches that occur in the conventional two-switch DAC architecture.

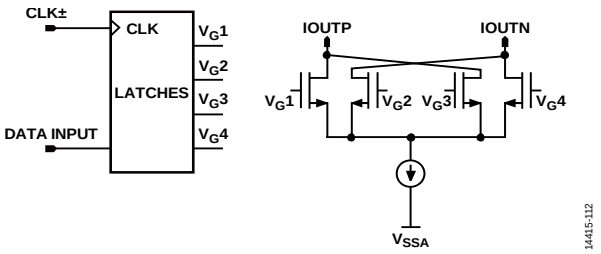


*Figure 126. Quad-Switch Architecture*

In two-switch architecture, when a switch transition occurs and $D_1$ and $D_2$ are in different states, a glitch occurs. However, if $D_1$ and $D_2$ happen to be at the same state, the switch transitions and no glitches occur. This code dependent glitching causes an increased amount of distortion in the DAC. In quad-switch architecture (no matter what the codes are), two switches are always transitioning at each half-clock cycle, thus eliminating the code-dependent glitches, but, in the process, creating a constant glitch at $2 \times f_{DAC}$. For this reason, a significant clock spur at $2 \times f_{DAC}$ is evident in the DAC output spectrum.

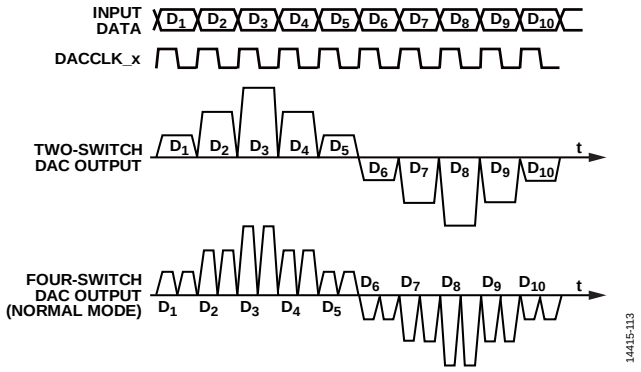


*Figure 127. Two-Switch and Quad-Switch DAC Waveforms*

As a consequence of the quad-switch architecture enabling updates on each half-clock cycle, it is possible to operate that DAC core at 2× the DAC clock rate if new data samples are latched into the DAC core on both the rising and falling edges of the DAC clock. This notion serves as the basis when operating the AD9163 in either Mix-Mode or return to zero (RZ) mode. In each case, the DAC core is presented with new data samples on each clock edge: in RZ mode, the rising edge clocks data and the falling edge clocks zero, whereas in Mix-Mode, the falling edge sample is simply the complement of the rising edge sample value.

When Mix-Mode is used, the output is effectively chopped at the DAC sample rate. This chopping has the effect of reducing the power of the fundamental signal while increasing the power of the images centered around the DAC sample rate, thus improving the dynamic range of these images.

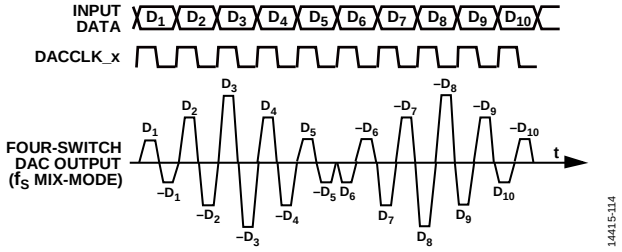


*Figure 128. Mix-Mode Waveform*

This ability to change modes provides the user the flexibility to place a carrier anywhere in the first three Nyquist zones, depending on the operating mode selected. Switching between baseband and Mix-Mode reshapes the sinc roll-off inherent at the DAC output. In baseband mode, the sinc null appears at $f_{DAC}$ because the same sample latched on the rising clock edge is also latched again on the falling clock edge, thus resulting in the same ubiquitous sinc response of a traditional DAC. In Mix-Mode, the complement sample of the rising edge is latched on the falling edge, therefore pushing the sinc null to $2 \times f_{DAC}$. Figure 129 shows the ideal frequency response of the three modes with the sinc roll-off included.

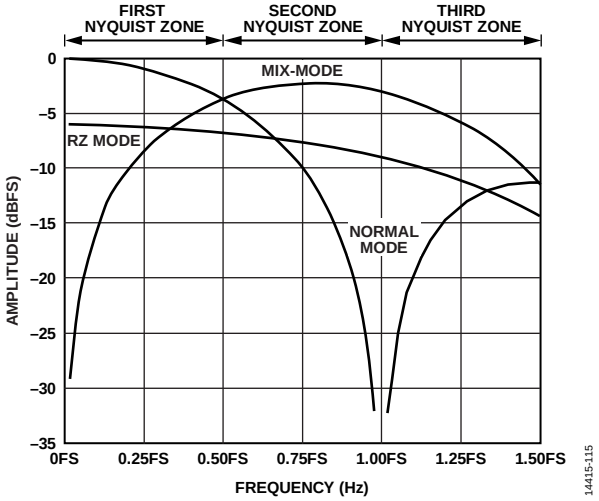


*Figure 129. Sinc Roll-Off for NRZ, RZ, and Mix-Mode Operation*

The quad-switch can be configured via the SPI (Register 0x152, Bits[1:0]) to operate in either NRZ mode (0b00), RZ mode (0b10), or Mix-Mode (0b01). The AD9163 has an additional frequency response characteristic due to the FIR85 filter. This filter samples data on both the rising and falling edges of the DAC clock, in essence doubling the input clock frequency. As a result, the NRZ (normal) mode roll-off in Figure 129 is extended to $2 \times f_{DAC}$ in Figure 129, and follows the Mix-Mode roll-off due to the zero-order hold at 2 × DAC clock (see Figure 130).

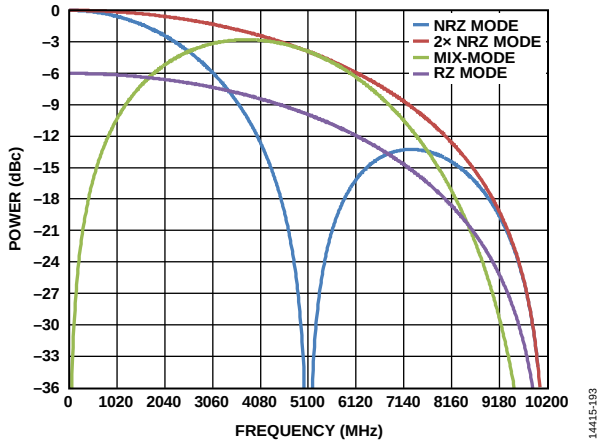


*Figure 130. Sinc Roll-Off with 2× NRZ Mode Added, $f_{DAC}$ = 5.1 GSPS*

## CLOCK INPUT

The AD9163 contains a low jitter, differential clock receiver that is capable of interfacing directly to a differential or single-ended clock source. Because the input is self biased with a nominal impedance of 90 Ω, it is recommended that the clock source be ac-coupled to the CLK± input pins. The nominal differential input is 1 V p-p, but the clock receiver can operate with a span that ranges from 250 mV p-p to 2.0 V p-p. Better phase noise performance is achieved with a higher clock input level.

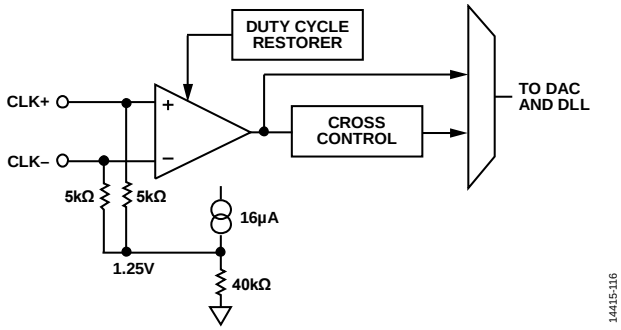


*Figure 131. Clock Input*

The quality of the clock source, as well as its interface to the AD9163 clock input, directly impacts ac performance. Select the phase noise and spur characteristics of the clock source to meet the target application requirements. Phase noise and spurs at a given frequency offset on the clock source are directly translated to the output signal. It can be shown that the phase noise characteristics of a reconstructed output sine wave are related to the clock source by 20 × $\log_{10}$ ($f_{OUT}/f_{CLK}$) when the DAC clock path contribution is negligible.

Figure 133 shows a clock source based on the ADF4355 low phase noise/jitter PLL. The ADF4355 can provide output frequencies from 54 MHz up to 6.8 GHz.

The clock control registers exist at Address 0x082 through Address 0x084. CLK_DUTY (Register 0x082) can be used to enable duty cycle correction (Bit 7), enable duty cycle offset control (Bit 6), and set the duty cycle offset (Bits[4:0]). The duty cycle offset word is a signed magnitude word, with Bit 4 being the sign bit (1 is negative) and Bits[3:0] the magnitude. The duty cycle adjusts across a range of approximately ±3%. Recommended settings for this register are listed in the Start-Up Sequence section.

The clock input has a register that adjusts the phase of the CLK+ and CLK– inputs. This register is located at Address 0x07F. The register has a signed magnitude (1 is negative) value that adds capacitance at ~20 fF per step to either the CLK+ or the CLK– input, according to Table 40. The CLK_PHASE_TUNE register can be used to adjust the clock input phase for better DAC image rejection.

**Table 40. CLK± Phase Adjust Values**

| Register 0x07F, Bits[5:0] | Capacitance at CLK+ | Capacitance at CLK– |
|---|---|---|
| 000000 | 0 | 0 |
| 000001 | 1 × 20 fF | 0 |
| 000010 | 2 × 20 fF | 0 |
| … | … | … |
| 011111 | 31 × 20 fF | 0 |
| 100000 | 0 | 0 |
| 100001 | 0 | 1 × 20 fF |
| 100010 | 0 | 2 × 20 fF |
| … | … | … |
| 111111 | 0 | 31 × 20 fF |

The improvement in performance from making these adjustments depends on the accuracy of the balance of the clock input balun and varies from unit to unit. Thus, if a high level of image rejection is required, it is likely that a per unit calibration is necessary. Performing this calibration can yield significant improvements, as much as 20 dB additional rejection of the image due to imbalance. Figure 132 shows the results of tuning clock phase, duty cycle (left at default in this case), and cross control. The improvement to performance, particularly at higher frequencies, can be as much as 20 dB.

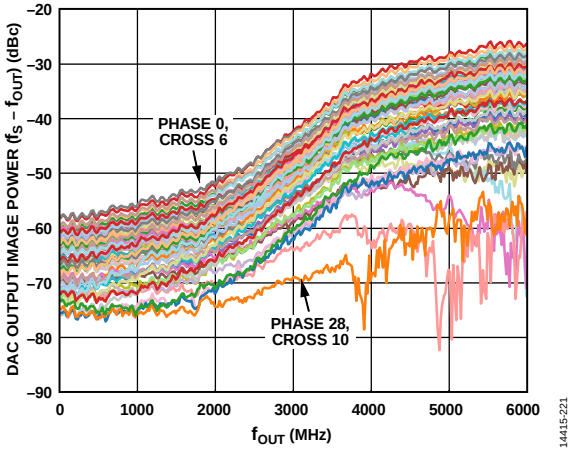


*Figure 132. Performance Improvement from Tuning the Clock Input*

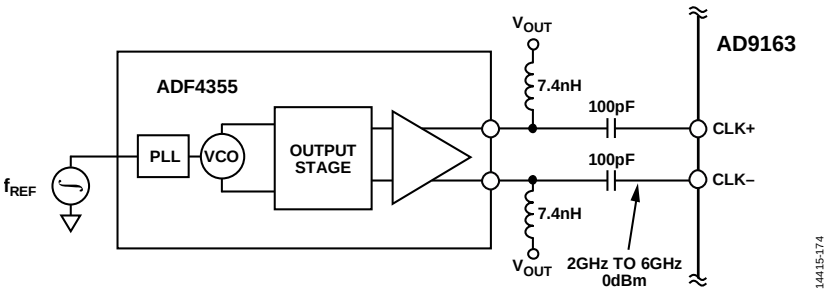


*Figure 133. Possible Signal Chain for CLK± Input*

## SHUFFLE MODE

The spurious performance of the AD9163 can be improved with a feature called shuffle mode. Shuffle mode uses proprietary technology to spread the energy of spurious signals across the DAC output as random noise. Shuffle mode is enabled by programming Register 0x151, Bit 2 = 0b1. Because shuffle is implemented with the MSBs, it is more effective when the DAC is operated with a small amount of digital backoff.

The amount of noise rise caused by shuffle mode is directly related to the power in the affected spurious signals. Because the AD9163 has good spurious performance without shuffle active, the penalty of shuffle mode to the noise spectral density is typically about 1 dB to 3 dB. Shuffle mode reduces spurious performance related to clock and foldback spurs, but does not affect real harmonics of the DAC output. Examples of the effects of shuffle mode are given in the Typical Performance Characteristics section (see Figure 47, Figure 48, Figure 62, Figure 63, and Figure 64).

## DLL

The CLK± input goes to a high frequency DLL to ensure robust locking of the DAC sample clock to the input clock. The DLL is configured and enabled as part of the recommended start-up sequence. The DLL control registers are located at Register 0x090 through Register 0x09B. The DLL settings are determined during product characterization and are given in the recommended start-up sequence (see the Start-Up Sequence section). It is not normally necessary to change these values, nor is the product characterization data valid on any settings other than the recommended ones.

## VOLTAGE REFERENCE

The AD9163 output current is set by a combination of digital control bits and the ISET reference current, as shown in Figure 134.

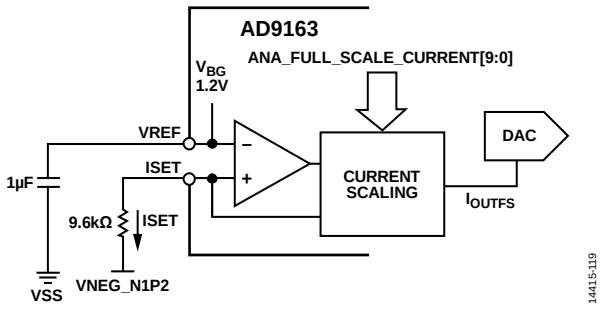


*Figure 134. Voltage Reference Circuit*

The reference current is obtained by forcing the band gap voltage across an external 9.6 kΩ resistor from ISET (Ball A12) to VNEG_N1P2. The 1.2 V nominal band gap voltage (VREF) generates a 125 µA reference current, ISET, in the 9.6 kΩ resistor, $R_{SET}$. The maximum full-scale current setting is related to the external resistor by the following equation:

$$I_{OUTFS} = 1.2\ V/R_{SET} \times 320\ (mA)$$

Note the following constraints when configuring the voltage reference circuit:

- Both the 9.6 kΩ resistor and 1 µF bypass capacitor are required for proper operation.
- Adjusting the DAC output full-scale current, $I_{OUTFS}$, from its default setting of 40 mA must be performed digitally.
- The AD9163 is not a multiplying DAC. Modulation of the reference current, ISET, with an ac signal is not supported.
- The band gap voltage appearing at the VREF pin must be buffered for use with an external circuitry because it has a high output impedance.
- An external reference can be used to overdrive the internal reference by connecting it to the VREF pin.

The $I_{OUTFS}$ value can be adjusted digitally over an 8 mA to 40 mA range by the ANA_FULL_SCALE_CURRENT[9:0] bits (Register 0x042, Bits[7:0] and Register 0x041, Bits[1:0]). The following equation relates $I_{OUTFS}$ to the ANA_FULL_SCALE_CURRENT[9:0] bits, which can be set from 0 to 1023.

$$I_{OUTFS} = 32\ mA \times (ANA_FULL_SCALE_CURRENT[9:0]/1023) + 8\ mA$$

Note that the default value of 0x3FF generates 40 mA full scale, and this value is used for most of the characterization presented in this data sheet, unless noted otherwise.

## TEMPERATURE SENSOR

The AD9163 has a band gap temperature sensor for monitoring the temperature changes of the AD9163. The temperature must be calibrated against a known temperature to remove the device to device variation on the band gap circuit that senses the temperature.

To calibrate the temperature, the user must take a reading at a known ambient temperature for a single point calibration of the AD9163 device. The slope for the formula is then calculated as

$$M = (T_{REF} + 190)/((CODE_REF)/1000)$$

where:
$T_{REF}$ is the calibrated temperature at which the temperature sensor is read.
*CODE_REF* is the readback code at the measured temperature, $T_{REF}$.

To monitor temperature change,

$$T_X = T_{REF} + M \times (CODE_X - CODE_REF)/1000$$

where:
*CODE_X* is the readback code at the unknown temperature, $T_X$.
*CODE_REF* is the readback code at the calibrated temperature, $T_{REF}$.

To use the temperature sensor, enable it by setting Register 0x135 to Register 0xA1. The user must write a 1 to Register 0x134, Bit 0 before reading back the die temperature from Register 0x132 (LSB) and Register 0x133 (MSB).

## ANALOG OUTPUTS

### *Equivalent DAC Output and Transfer Function*

The AD9163 provides complementary current outputs, OUTPUT+ and OUTPUT−, that sink current from an external load that is referenced to the 2.5 V VDD25_DAC supply. Figure 135 shows an equivalent output circuit for the DAC. Compared to most current output DACs of this type, the outputs of the AD9163 consists of a constant current ($I_{FIXED}$), and a peak differential ac current, $I_{CS}$ ($I_{CS} = I_{CSP} + I_{CSN}$). These two currents combine to form the $I_{INTx}$ currents shown in Figure 135. The internal currents, $I_{INTP}$ and $I_{INTN}$, are sent to the output pin and to an input termination resistance equivalent to 100 Ω pulled to the VDD25_DAC supply ($R_{INT}$). This termination serves to divide the output current based on the external termination resistors that are pulled to VDD25_DAC.

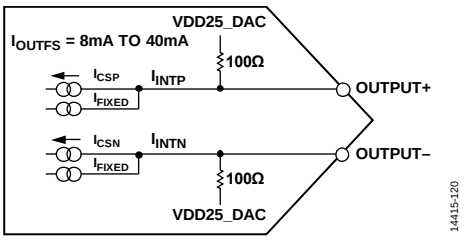


*Figure 135. Equivalent DAC Output Circuit*

The example shown in Figure 135 can be modeled as a pair of dc current sources that source a current of $I_{OUTFS}$ to each output. This differential ac current source is used to model the signal (that is, a digital code) dependent nature of the DAC output. The polarity and signal dependency of this ac current source are related to the digital code (F) by the following equation:

$$F\ (code) = (DACCODE - 32{,}768)/32{,}768 \quad (2)$$

where:
$-1 \le F\ (code) < +1$.
*DACCODE* = 0 to 65,535 (decimal).

The current that is measured at the OUTPUT+ and OUTPUT− outputs is as follows:

$$OUTPUT+ = (I_{FIXED}\ (\text{mA}) + (F \times I_{OUTFS})/F_{MAX}(\text{mA})) \times (R_{INT}/(R_{INT} + R_{LOAD})) \quad (3)$$

$$OUTPUT- = (I_{FIXED}\ (\text{mA}) + ((F_{MAX} - F) \times I_{OUTFS})/F_{MAX}(\text{mA})) \times (R_{INT}/(R_{INT} + R_{LOAD}))$$

The $I_{FIXED}$ value is about 3.8 mA. It is important to note that the AD9163 output cannot support dc coupling to the external load, and thus must be ac-coupled through appropriately sized capacitors for the chosen operating frequencies. Figure 136 shows the OUTPUT+ vs. DAC code transfer function when $I_{OUTFS}$ is set to 40 mA.

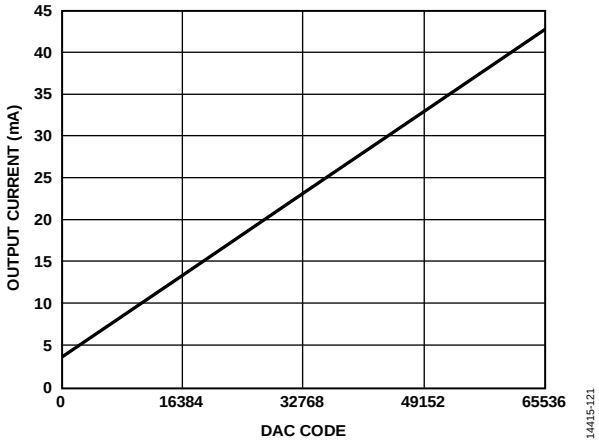


*Figure 136. Gain Curve for ANA_FULL_SCALE_CURRENT[9:0] = 1023, DAC Offset = 3.8 mA*

### *Peak DAC Output Power Capability*

The maximum peak power capability of a differential current output DAC is dependent on its peak differential ac current, $I_{PEAK}$, and the equivalent load resistance it sees. In the case of a 1:1 balun with 100 Ω differential source termination, the equivalent load that is seen by the DAC ac current source is 50 Ω. If the AD9163 is programmed for an $I_{OUTFS}$ = 40 mA, its ideal peak ac current is 20 mA and its maximum power, delivered to the equivalent load, is $10 \times (R_{INT}/(R_{INT} + R_{LOAD}) = 8$ mW (that is, $P = I^2R$). Because the source and load resistance seen by the 1:1 balun are equal, this power is shared equally. Therefore, the output load receives 4 mW, or 6 dBm maximum power.

To calculate the rms power delivered to the load, consider

- Peak to rms ratio of the digital waveform
- Any digital backoff from digital full scale
- DAC sinc response and nonideal losses in the external network
- DAC analog roll-off due to switch parasitic capacitance and load impedance

For example, a sine wave with no digital backoff ideally measures 6 dBm. If a typical balun loss of 1.2 dB is included, expect to measure 4.8 dBm of actual power in the region where the sinc response of the DAC has negligible influence and analog roll-off has not begun. Increasing the output power is best accomplished by increasing $I_{OUTFS}$. An example of DAC output characteristics for several balun and board types is shown in Figure 137.

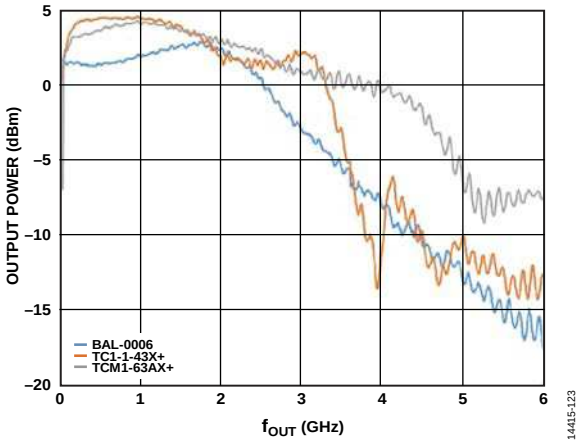


*Figure 137. Measured DAC Output Response; $f_{DAC}$ = 6 GSPS*

### Output Stage Configuration

The AD9163 is intended to serve high dynamic range applications that require wide signal reconstruction bandwidth (such as a DOCSIS cable modem termination system (CMTS)) and/or high IF/RF signal generation. Optimum ac performance can be realized only when the DAC output is configured for differential (that is, balanced) operation with its output common-mode voltage biased to a stable, low noise 2.5 V nominal analog supply (VDD25_DAC).

The output network used to interface to the DAC provides a near 0 Ω dc bias path to VDD25_DAC. Any imbalance in the output impedance over frequency between the OUTPUT+ and OUTPUT– pins degrades the distortion performance (mostly even order) and noise performance. Component selection and layout are critical in realizing the performance potential of the AD9163.

Most applications that require balanced to unbalanced conversion from 10 MHz to 3 GHz can take advantage of several available transformers that offer impedance ratios of both 2:1 and 1:1.

Figure 138 shows the AD9163 interfacing to the Mini-Circuits TCM1-63AX+ and the TC1-1-43X+ transformers.

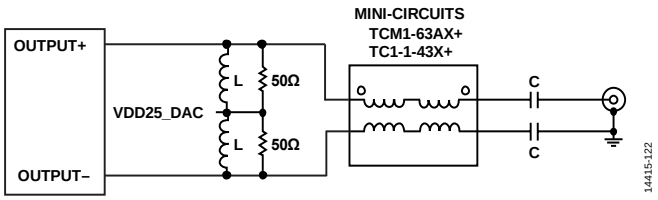


*Figure 138. Recommended Transformer for Wideband Applications with Upper Bandwidths of up to 5 GHz*

To assist in matching the AD9163 output, an equivalent model of the output was developed, and is shown in Figure 139. This equivalent model includes all effects from the ideal 40 mA current source in the die to the ball of the CSP_ BGA package, including parasitic capacitance, trace inductance and resistance, contact resistance of solder bumps, via inductance, and other effects.

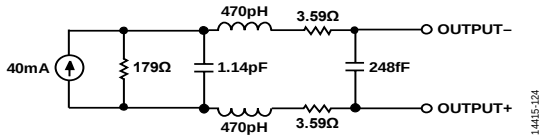


*Figure 139. Equivalent Circuit Model of the DAC Output*

A Smith chart is provided in Figure 140 showing the simulated S11 of the DAC output, using the model in Figure 139. The plot was taken using the circuit in Figure 139, with a 100 Ω differential load instead of the balun. For the measured response of the DAC output, see Figure 137.

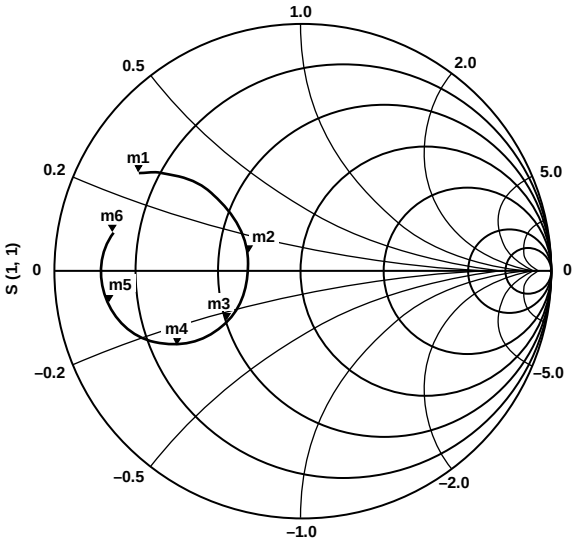


*Figure 140. Simulated Smith Chart Showing the DAC Output Impedance, $Z_O$ = 100 Ω*

# START-UP SEQUENCE

A number of steps is required to program the AD9163 to the proper operating state after the device is powered up. This sequence is divided into several steps, and is listed in Table 41, Table 42, and Table 43, along with an explanation of the purpose of each step. Private registers are reserved but must be written for proper operation. Blank cells in Table 41 to Table 43 mean that the value depends on the result as described in the description column.

The AD9163 is calibrated at the factory as part of the automatic test program. The configure DAC start-up sequence loads the factory calibration coefficients, as well as configures some parameters that optimize the performance of the DAC and the DAC clock DLL (see Table 41). Run this sequence whenever the DAC is powered down or reset.

The configure JESD204B sequence configures the SERDES block and then brings up the links (see Table 42). First, run the configure DAC start-up sequence, then run the configure JESD204B sequence.

Follow the configure NCO sequence if using the NCO (see Table 43). The configure DAC start-up sequence is run first, then the configure NCO sequence.

**Table 41. Configure DAC Start-Up Sequence After Power-Up**

| R/W | Register | Value | Description |
|---|---|---|---|
| W | 0x000 | 0x18 | Configure the device for 4-wire serial port operation (optional: leave at the default of 3-wire SPI) |
| W | 0x0D2 | 0x52 | Reset internal calibration registers (private) |
| W | 0x0D2 | 0xD2 | Clear the reset bit for the internal calibration registers (private) |
| W | 0x606 | 0x02 | Configure the nonvolatile random access memory (NVRAM) (private) |
| W | 0x607 | 0x00 | Configure the NVRAM (private) |
| W | 0x604 | 0x01 | Load the NVRAM; loads factory calibration factors from the NVRAM (private) |
| R | 0x003, 0x004, 0x005, 0x006 | N/A[1] | (Optional) read CHIP_TYPE, PROD_ID[15:0], PROD_GRADE, and DEV_REVISION from Register 0x003, Register 0x004, Register 0x005, and Register 0x006 |
| R | 0x604, Bit 1 | 0b1 | (Optional) read the boot loader pass bit in Register 0x604, Bit 1 = 0b1 to indicate a successful boot load (private) |
| W | 0x058 | 0x03 | Enable the band gap reference (private) |
| W | 0x090 | 0x1E | Power up the DAC clock DLL |
| W | 0x080 | 0x00 | Enable the clock receiver |
| W | 0x040 | 0x00 | Enable the DAC bias circuits |
| W | 0x020 | 0x0F | (Optional) enable the interrupts |
| W | 0x09E | 0x85 | Configure DAC analog parameters (private) |
| W | 0x091 | 0xE9 | Enable the DAC clock DLL |
| R | 0x092, Bit 0 | 0b1 | Check DLL_STATUS; set Register 0x092, Bit 0 = 1 to indicate the DAC clock DLL is locked to the DAC clock input |
| W | 0x0E8 | 0x20 | Enable calibration factors (private) |
| W | 0x152, Bits[1:0] | | Configure the DAC decode mode (0b00 = NRZ, 0b01 = Mix-Mode, or 0b10 = RZ) |

[1] N/A means not applicable.

**Table 42. Configure JESD204B Start-Up Sequence**

| R/W | Register | Value | Description |
|---|---|---|---|
| W | 0x300 | 0x00 | Ensure the SERDES links are disabled before configuring them. |
| W | 0x4B8 | 0xFF | Enable JESD204B interrupts. |
| W | 0x4B9 | 0x01 | Enable JESD204B interrupts. |
| W | 0x480 | 0x38 | Enable SERDES error counters. |
| W | 0x481 | 0x38 | Enable SERDES error counters. |
| W | 0x482 | 0x38 | Enable SERDES error counters. |
| W | 0x483 | 0x38 | Enable SERDES error counters. |
| W | 0x484 | 0x38 | Enable SERDES error counters. |
| W | 0x485 | 0x38 | Enable SERDES error counters. |
| W | 0x486 | 0x38 | Enable SERDES error counters. |
| W | 0x487 | 0x38 | Enable SERDES error counters. |
| W | 0x110 | | Configure number of lanes (Bits[7:4]) and interpolation rate (Bits[3:0]). |

| R/W | Register | Value | Description |
|---|---|---|---|
| W | 0x111 | | Configure the datapath options for Bit 7 (INVSINC_EN), Bit 6 (NCO_EN), Bit 4 (FILT_BW), Bit 2 (MODULUS_EN), Bit 1 (SEL_SIDEBAND), and Bit 0 (FIR85_FILT_EN). See the Register Summary section for details on the options. Set the reserved bits (Bit 5 and Bit 3) to 0b0. |
| W | 0x230 | | Configure the CDR block according to Table 18 for both half rate enable and the divider. |
| W | 0x289, Bits[1:0] | | Set up the SERDES PLL divider based on the conditions shown in Table 17. |
| W | 0x084, Bits[5:4] | | Set up the PLL reference clock rate based on the conditions shown in Table 17. |
| W | 0x200 | 0x00 | Enable JESD204B block (disable master SERDES power-down). |
| W | 0x475 | 0x09 | Soft reset of the JESD204B quad-byte deframer. |
| W | 0x453, Bit 7 | 0b1 | (Optional) enable scrambling on SERDES lanes. |
| W | 0x458, Bits[7:5] | | Set the subclass type: 0b000 = Subclass 0, 0b001 = Subclass 1. |
| W | 0x459, Bits[7:5] | 0b1 | Set the JESD204x version to JESD204B. |
| W | 0x45D | | Program the calculated checksum value for Lane 0 from values in Register 0x450 to Register 0x45C. |
| W | 0x475 | 0x01 | Bring the JESD204B quad-byte deframer out of reset. |
| W | 0x201, Bits[7:0] | | Set any bits to 1 to power down the appropriate physical lane. |
| W | 0x2A7 | 0x01 | (Optional) calibrate SERDES PHY Termination Block 1 (PHY 0, PHY 1, PHY 6, PHY 7). |
| W | 0x2AE | 0x01 | (Optional) calibrate SERDES PHY Termination Block 2 (PHY 2, PHY 3, PHY 4, PHY 5). |
| W | 0x29E | 0x1F | Override defaults in the SERDES PLL settings (private). |
| W | 0x206 | 0x00 | Reset the CDR. |
| W | 0x206 | 0x01 | Enable the CDR. |
| W | 0x280 | 0x03 | Enable the SERDES PLL. |
| R | 0x281, Bit 0 | 0b1 | Read back Register 0x281 until Bit 0 = 1 to indicate the SERDES PLL is locked. Prior to enabling the links, be sure that the JESD204B transmitter is enabled and ready to begin bringing up the link. |
| W | 0x300 | 0x01 | Enable SERDES links (begin bringing up the link). |
| R | 0x470 | 0xFF | Read the CGS status for all lanes. |
| R | 0x471 | 0xFF | Read the frame sync status for all lanes. |
| R | 0x472 | 0xFF | Read the good checksum status for all lanes. |
| R | 0x473 | 0xFF | Read the initial lane sync status for all lanes. |
| W | 0x024 | 0x1F | Clear the interrupts. |
| W | 0x4BA | 0xFF | Clear the SERDES interrupts. |
| W | 0x4BB | 0x01 | Clear the SERDES interrupt. |

**Table 43. Configure NCO Sequence**

| R/W | Register | Value | Description |
|---|---|---|---|
| W | 0x111, Bit 6 | 0b1 | Configure NCO_EN (Bit 6) = 0b1. Configure other datapath options for Bit 7 (INVSINC_EN), Bit 4 (FILT_BW), Bit 2 (MODULUS_EN), Bit 1 (SEL_SIDEBAND), and Bit 0 (FIR85_FILT_EN). See the Register Summary section for details on the options. Set the reserved bits (Bit 5 and Bit 3) to 0b0. |
| W | 0x113 | 0x00 | Ensure the frequency tuning word write request is low. |
| W | 0x119 | | Write FTW, Bits[47:40]. |
| W | 0x118 | | Write FTW, Bits[39:32]. |
| W | 0x117 | | Write FTW, Bits[31:24]. |
| W | 0x116 | | Write FTW, Bits[23:16]. |
| W | 0x115 | | Write FTW, Bits[15:8]. |
| W | 0x114 | | Write FTW, Bits[7:0]. |
| W | 0x113 | 0x01 | Load the FTW to the NCO. |

# REGISTER SUMMARY

**Table 44. Register Summary**

| Reg. | Name | Bits | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Reset | RW |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x000 | SPI_INTFCONFA | [7:0] | SOFTRESET_M | LSBFIRST_M | ADDRINC_M | SDOACTIVE_M | SDOACTIVE | ADDRINC | LSBFIRST | SOFTRESET | 0x00 | R/W |
| 0x001 | SPI_INTFCONFB | [7:0] | SINGLEINS | $\overline{CS}$STALL | RESERVED | | | SOFTRESET1 | SOFTRESET0 | RESERVED | 0x00 | R/W |
| 0x002 | SPI_DEVCONF | [7:0] | DEVSTATUS | | | | CUSTOPMODE | | SYSOPMODE | | 0x00 | R/W |
| 0x003 | SPI_CHIPTYPE | [7:0] | CHIP_TYPE | | | | | | | | 0x00 | R |
| 0x004 | SPI_PRODIDL | [7:0] | PROD_ID[7:0] | | | | | | | | 0x00 | R |
| 0x005 | SPI_PRODIDH | [7:0] | PROD_ID[15:8] | | | | | | | | 0x00 | R |
| 0x006 | SPI_CHIPGRADE | [7:0] | PROD_GRADE | | | | DEV_REVISION | | | | 0x00 | R |
| 0x020 | IRQ_ENABLE | [7:0] | RESERVED | | | EN_SYSREF_JITTER | EN_DATA_READY | EN_LANE_FIFO | EN_PRBSQ | EN_PRBSI | 0x00 | R/W |
| 0x024 | IRQ_STATUS | [7:0] | RESERVED | | | IRQ_SYSREF_JITTER | IRQ_DATA_READY | IRQ_LANE_FIFO | IRQ_PRBSQ | IRQ_PRBSI | 0x00 | R/W |
| 0x031 | SYNC_LMFC_DELAY_FRAME | [7:0] | RESERVED | | | SYNC_LMFC_DELAY_SET_FRM | | | | | 0x00 | R/W |
| 0x032 | SYNC_LMFC_DELAY0 | [7:0] | SYNC_LMFC_DELAY_SET[7:0] | | | | | | | | 0x00 | R/W |
| 0x033 | SYNC_LMFC_DELAY1 | [7:0] | RESERVED | | | | SYNC_LMFC_DELAY_SET[11:8] | | | | 0x00 | R/W |
| 0x034 | SYNC_LMFC_STAT0 | [7:0] | SYNC_LMFC_DELAY_STAT[7:0] | | | | | | | | 0x00 | R/W |
| 0x035 | SYNC_LMFC_STAT1 | [7:0] | RESERVED | | | | SYNC_LMFC_DELAY_STAT[11:8] | | | | 0x00 | R/W |
| 0x036 | SYSREF_COUNT | [7:0] | SYSREF_COUNT | | | | | | | | 0x00 | R/W |
| 0x037 | SYSREF_PHASE0 | [7:0] | SYSREF_PHASE[7:0] | | | | | | | | 0x00 | R/W |
| 0x038 | SYSREF_PHASE1 | [7:0] | RESERVED | | | | SYSREF_PHASE[11:8] | | | | 0x00 | R/W |
| 0x039 | SYSREF_JITTER_WINDOW | [7:0] | RESERVED | | SYSREF_JITTER_WINDOW | | | | | | 0x00 | R/W |
| 0x03A | SYNC_CTRL | [7:0] | RESERVED | | | | | | SYNC_MODE | | 0x00 | R/W |
| 0x03F | TX_ENABLE | [7:0] | SPI_DATAPATH_POST | SPI_DATAPATH_PRE | RESERVED | | TXEN_NCO_RESET | TXEN_DATAPATH_POST | TXEN_DATAPATH_PRE | TXEN_DAC_FSC | 0xC0 | R/W |
| 0x040 | ANA_DAC_BIAS_PD | [7:0] | RESERVED | | | | | | ANA_DAC_BIAS_PD1 | ANA_DAC_BIAS_PD0 | 0x03 | R/W |
| 0x041 | ANA_FSC0 | [7:0] | RESERVED | | | | | | ANA_FULL_SCALE_CURRENT[1:0] | | 0x03 | R/W |
| 0x042 | ANA_FSC1 | [7:0] | ANA_FULL_SCALE_CURRENT[9:2] | | | | | | | | 0xFF | R/W |
| 0x07F | CLK_PHASE_TUNE | [7:0] | RESERVED | | CLK_PHASE_TUNE | | | | | | 0x00 | R/W |
| 0x080 | CLK_PD | [7:0] | RESERVED | | | | | | | DACCLK_PD | 0x01 | R/W |
| 0x082 | CLK_DUTY | [7:0] | CLK_DUTY_EN | CLK_DUTY_OFFSET_EN | CLK_DUTY_BOOST_EN | CLK_DUTY_PRG | | | | | 0x80 | R/W |
| 0x083 | CLK_CRS_CTRL | [7:0] | CLK_CRS_EN | RESERVED | | | CLK_CRS_ADJ | | | | 0x80 | R/W |
| 0x084 | PLL_REF_CLK_PD | [7:0] | RESERVED | | PLL_REF_CLK_RATE | | RESERVED | | | PLL_REF_CLK_PD | 0x00 | R/W |
| 0x088 | SYSREF_CTRL0 | [7:0] | RESERVED | | | | HYS_ON | SYSREF_RISE | HYS_CNTRL[9:8] | | 0x00 | R/W |
| 0x089 | SYSREF_CTRL1 | [7:0] | HYS_CNTRL[7:0] | | | | | | | | 0x00 | R/W |
| 0x090 | DLL_PD | [7:0] | RESERVED | | | DLL_FINE_DC_EN | DLL_FINE_XC_EN | DLL_COARSE_DC_EN | DLL_COARSE_XC_EN | DLL_CLK_PD | 0x1F | R/W |
| 0x091 | DLL_CTRL | [7:0] | DLL_TRACK_ERR | DLL_SEARCH_ERR | DLL_SLOPE | DLL_SEARCH | | DLL_MODE | | DLL_ENABLE | 0xF0 | R/W |
| 0x092 | DLL_STATUS | [7:0] | RESERVED | | | | | DLL_FAIL | DLL_LOST | DLL_LOCKED | 0x00 | R/W |
| 0x093 | DLL_GB | [7:0] | RESERVED | | | | DLL_GUARD | | | | 0x00 | R/W |
| 0x094 | DLL_COARSE | [7:0] | RESERVED | | DLL_COARSE | | | | | | 0x00 | R/W |
| 0x095 | DLL_FINE | [7:0] | DLL_FINE | | | | | | | | 0x80 | R/W |

| Reg. | Name | Bits | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Reset | RW |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x096 | DLL_PHASE | [7:0] | RESERVED | | | DLL_PHS | | | | | 0x08 | R/W |
| 0x097 | DLL_BW | [7:0] | RESERVED | | | DLL_FILT_BW | | | DLL_WEIGHT | | 0x00 | R/W |
| 0x098 | DLL_READ | [7:0] | RESERVED | | | | | | | DLL_READ | 0x00 | R/W |
| 0x099 | DLL_COARSE_RB | [7:0] | RESERVED | | DLL_COARSE_RB | | | | | | 0x00 | R |
| 0x09A | DLL_FINE_RB | [7:0] | DLL_FINE_RB | | | | | | | | 0x00 | R |
| 0x09B | DLL_PHASE_RB | [7:0] | RESERVED | | | DLL_PHS_RB | | | | | 0x00 | R |
| 0x09D | DIG_CLK_INVERT | [7:0] | RESERVED | | | | | INV_DIG_CLK | DIG_CLK_DC_EN | DIG_CLK_XC_EN | 0x03 | R/W |
| 0x0A0 | DLL_CLK_DEBUG | [7:0] | DLL_TEST_EN | RESERVED | | | | | DLL_TEST_DIV | | 0x00 | R/W |
| 0x110 | INTERP_MODE | [7:0] | JESD_LANES | | | | INTERP_MODE | | | | 0x81 | R/W |
| 0x111 | DATAPATH_CFG | [7:0] | INVSINC_EN | NCO_EN | RESERVED | FILT_BW | RESERVED | MODULUS_EN | SEL_SIDEBAND | FIR85_FILT_EN | 0x00 | R/W |
| 0x113 | FTW_UPDATE | [7:0] | RESERVED | FTW_REQ_MODE | | | RESERVED | FTW_LOAD_SYSREF | FTW_LOAD_ACK | FTW_LOAD_REQ | 0x00 | R/W |
| 0x114 | FTW0 | [7:0] | FTW[7:0] | | | | | | | | 0x00 | R/W |
| 0x115 | FTW1 | [7:0] | FTW[15:8] | | | | | | | | 0x00 | R/W |
| 0x116 | FTW2 | [7:0] | FTW[23:16] | | | | | | | | 0x00 | R/W |
| 0x117 | FTW3 | [7:0] | FTW[31:24] | | | | | | | | 0x00 | R/W |
| 0x118 | FTW4 | [7:0] | FTW[39:32] | | | | | | | | 0x00 | R/W |
| 0x119 | FTW5 | [7:0] | FTW[47:40] | | | | | | | | 0x00 | R/W |
| 0x11C | PHASE_OFFSET0 | [7:0] | NCO_PHASE_OFFSET[7:0] | | | | | | | | 0x00 | R/W |
| 0x11D | PHASE_OFFSET1 | [7:0] | NCO_PHASE_OFFSET[15:8] | | | | | | | | 0x00 | R/W |
| 0x124 | ACC_MODULUS0 | [7:0] | ACC_MODULUS[7:0] | | | | | | | | 0x00 | R/W |
| 0x125 | ACC_MODULUS1 | [7:0] | ACC_MODULUS[15:8] | | | | | | | | 0x00 | R/W |
| 0x126 | ACC_MODULUS2 | [7:0] | ACC_MODULUS[23:16] | | | | | | | | 0x00 | R/W |
| 0x127 | ACC_MODULUS3 | [7:0] | ACC_MODULUS[31:24] | | | | | | | | 0x00 | R/W |
| 0x128 | ACC_MODULUS4 | [7:0] | ACC_MODULUS[39:32] | | | | | | | | 0x00 | R/W |
| 0x129 | ACC_MODULUS5 | [7:0] | ACC_MODULUS[47:40] | | | | | | | | 0x00 | R/W |
| 0x12A | ACC_DELTA0 | [7:0] | ACC_DELTA[7:0] | | | | | | | | 0x00 | R/W |
| 0x12B | ACC_DELTA1 | [7:0] | ACC_DELTA[15:8] | | | | | | | | 0x00 | R/W |
| 0x12C | ACC_DELTA2 | [7:0] | ACC_DELTA[23:16] | | | | | | | | 0x00 | R/W |
| 0x12D | ACC_DELTA3 | [7:0] | ACC_DELTA[31:24] | | | | | | | | 0x00 | R/W |
| 0x12E | ACC_DELTA4 | [7:0] | ACC_DELTA[39:32] | | | | | | | | 0x00 | R/W |
| 0x12F | ACC_DELTA5 | [7:0] | ACC_DELTA[47:40] | | | | | | | | 0x00 | R/W |
| 0x132 | TEMP_SENS_LSB | [7:0] | TEMP_SENS_OUT[7:0] | | | | | | | | 0x00 | R |
| 0x133 | TEMP_SENS_MSB | [7:0] | TEMP_SENS_OUT[15:8] | | | | | | | | 0x00 | R |
| 0x134 | TEMP_SENS_UPDATE | [7:0] | RESERVED | | | | | | | TEMP_SENS_UPDATE | 0x00 | R/W |
| 0x135 | TEMP_SENS_CTRL | [7:0] | TEMP_SENS_FAST | RESERVED | | | | | | TEMP_SENS_ENABLE | 0x20 | |
| 0x14B | PRBS | [7:0] | PRBS_GOOD_Q | PRBS_GOOD_I | RESERVED | PRBS_INV_Q | PRBS_INV_I | PRBS_MODE | PRBS_RESET | PRBS_EN | 0x10 | R/W |
| 0x14C | PRBS_ERROR_I | [7:0] | PRBS_COUNT_I | | | | | | | | 0x00 | R |
| 0x14D | PRBS_ERROR_Q | [7:0] | PRBS_COUNT_Q | | | | | | | | 0x00 | R |
| 0x151 | DECODE_CTRL | [7:0] | RESERVED | | | | | SHUFFLE | RESERVED | | 0x01 | R/W |
| 0x152 | DECODE_MODE | [7:0] | RESERVED | | | | | | DECODE_MODE | | 0x00 | R/W |
| 0x1DF | SPI_STRENGTH | [7:0] | RESERVED | | | | SPIDRV | | | | 0x0F | R/W |
| 0x200 | MASTER_PD | [7:0] | RESERVED | | | | | | | SPI_PD_MASTER | 0x01 | R/W |
| 0x201 | PHY_PD | [7:0] | SPI_PD_PHY | | | | | | | | 0x00 | R/W |
| 0x203 | GENERIC_PD | [7:0] | RESERVED | | | | | | SPI_SYNC1_PD | RESERVED | 0x00 | R/W |
| 0x206 | CDR_RESET | [7:0] | RESERVED | | | | | | | SPI_CDR_$\overline{\text{RESET}}$ | 0x01 | R/W |

| Reg. | Name | Bits | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Reset | RW |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x230 | CDR_OPERATING_MODE_REG_0 | [7:0] | RESERVED | | SPI_ENHALFRATE | RESERVED | | SPI_DIVISION_RATE | | RESERVED | 0x28 | R/W |
| 0x250 | EQ_CONFIG_PHY_0_1 | [7:0] | SPI_EQ_CONFIG1 | | | | | SPI_EQ_CONFIG0 | | | 0x88 | R/W |
| 0x251 | EQ_CONFIG_PHY_2_3 | [7:0] | SPI_EQ_CONFIG3 | | | | | SPI_EQ_CONFIG2 | | | 0x88 | R/W |
| 0x252 | EQ_CONFIG_PHY_4_5 | [7:0] | SPI_EQ_CONFIG5 | | | | | SPI_EQ_CONFIG4 | | | 0x88 | R/W |
| 0x253 | EQ_CONFIG_PHY_6_7 | [7:0] | SPI_EQ_CONFIG7 | | | | | SPI_EQ_CONFIG6 | | | 0x88 | R/W |
| 0x268 | EQ_BIAS_REG | [7:0] | EQ_POWER_MODE | | RESERVED | | | | | | 0x62 | R/W |
| 0x280 | SYNTH_ENABLE_CNTRL | [7:0] | RESERVED | | | | | SPI_RECAL_SYNTH | RESERVED | SPI_ENABLE_SYNTH | 0x00 | R/W |
| 0x281 | PLL_STATUS | [7:0] | RESERVED | | SPI_CP_OVER_RANGE_HIGH_RB | SPI_CP_OVER_RANGE_LOW_RB | SPI_CP_CAL_VALID_RB | RESERVED | | SPI_PLL_LOCK_RB | 0x00 | R |
| 0x289 | REF_CLK_DIVIDER_LDO | [7:0] | RESERVED | | | | | | SERDES_PLL_DIV_FACTOR | | 0x04 | R/W |
| 0x2A7 | TERM_BLK1_CTRLREG0 | [7:0] | RESERVED | | | | | | | SPI_I_TUNE_R_CAL_TERMBLK1 | 0x00 | R/W |
| 0x2A8 | TERM_BLK1_CTRLREG1 | [7:0] | SPI_I_SERIALIZER_RTRIM_TERMBLK1 | | | | | | | | 0x00 | R/W |
| 0x2AC | TERM_BLK1_RD_REG0 | [7:0] | RESERVED | | | | | SPI_O_RCAL_CODE_TERMBLK1 | | | 0x00 | R |
| 0x2AE | TERM_BLK2_CTRLREG0 | [7:0] | RESERVED | | | | | | | SPI_I_TUNE_R_CAL_TERMBLK2 | 0x00 | R/W |
| 0x2AF | TERM_BLK2_CTRLREG1 | [7:0] | SPI_I_SERIALIZER_RTRIM_TERMBLK2 | | | | | | | | 0x00 | R/W |
| 0x2B3 | TERM_BLK2_RD_REG0 | [7:0] | RESERVED | | | | SPI_O_RCAL_CODE_TERMBLK2 | | | | 0x00 | R |
| 0x2BB | TERM_OFFSET_0 | [7:0] | RESERVED | | | | TERM_OFFSET_0 | | | | 0x00 | R/W |
| 0x2BC | TERM_OFFSET_1 | [7:0] | RESERVED | | | | TERM_OFFSET_1 | | | | 0x00 | R/W |
| 0x2BD | TERM_OFFSET_2 | [7:0] | RESERVED | | | | TERM_OFFSET_2 | | | | 0x00 | R/W |
| 0x2BE | TERM_OFFSET_3 | [7:0] | RESERVED | | | | TERM_OFFSET_3 | | | | 0x00 | R/W |
| 0x2BF | TERM_OFFSET_4 | [7:0] | RESERVED | | | | TERM_OFFSET_4 | | | | 0x00 | R/W |
| 0x2C0 | TERM_OFFSET_5 | [7:0] | RESERVED | | | | TERM_OFFSET_5 | | | | 0x00 | R/W |
| 0x2C1 | TERM_OFFSET_6 | [7:0] | RESERVED | | | | TERM_OFFSET_6 | | | | 0x00 | R/W |
| 0x2C2 | TERM_OFFSET_7 | [7:0] | RESERVED | | | | TERM_OFFSET_7 | | | | 0x00 | R/W |
| 0x300 | GENERAL_JRX_CTRL_0 | [7:0] | RESERVED | CHECKSUM_MODE | RESERVED | | | | | LINK_EN | 0x00 | R/W |
| 0x302 | DYN_LINK_LATENCY_0 | [7:0] | RESERVED | | | DYN_LINK_LATENCY_0 | | | | | 0x00 | R |
| 0x304 | LMFC_DELAY_0 | [7:0] | RESERVED | | | LMFC_DELAY_0 | | | | | 0x00 | R/W |
| 0x306 | LMFC_VAR_0 | [7:0] | RESERVED | | | LMFC_VAR_0 | | | | | 0x1F | R/W |
| 0x308 | XBAR_LN_0_1 | [7:0] | RESERVED | | SRC_LANE1 | | | SRC_LANE0 | | | 0x08 | R/W |
| 0x309 | XBAR_LN_2_3 | [7:0] | RESERVED | | SRC_LANE3 | | | SRC_LANE2 | | | 0x1A | R/W |
| 0x30A | XBAR_LN_4_5 | [7:0] | RESERVED | | SRC_LANE5 | | | SRC_LANE4 | | | 0x2C | R/W |
| 0x30B | XBAR_LN_6_7 | [7:0] | RESERVED | | SRC_LANE7 | | | SRC_LANE6 | | | 0x3E | R/W |
| 0x30C | FIFO_STATUS_REG_0 | [7:0] | LANE_FIFO_FULL | | | | | | | | 0x00 | R |
| 0x30D | FIFO_STATUS_REG_1 | [7:0] | LANE_FIFO_EMPTY | | | | | | | | 0x00 | R |
| 0x311 | $\overline{\text{SYNC}}$_GEN_0 | [7:0] | RESERVED | | | | | EOMF_MASK_0 | RESERVED | EOF_MASK_0 | 0x00 | R/W |
| 0x312 | $\overline{\text{SYNC}}$_GEN_1 | [7:0] | $\overline{\text{SYNC}}$_ERR_DUR | | | | | $\overline{\text{SYNC}}$_SYNCREQ_DUR | | | 0x00 | R/W |
| 0x313 | $\overline{\text{SYNC}}$_GEN_3 | [7:0] | LMFC_PERIOD | | | | | | | | 0x00 | R |

| Reg. | Name | Bits | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Reset | RW |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x315 | PHY_PRBS_TEST_EN | [7:0] | PHY_TEST_EN | | | | | | | | 0x00 | R/W |
| 0x316 | PHY_PRBS_TEST_CTRL | [7:0] | RESERVED | PHY_SRC_ERR_CNT | | | PHY_PRBS_PAT_SEL | | PHY_TEST_START | PHY_TEST_RESET | 0x00 | R/W |
| 0x317 | PHY_PRBS_TEST_THRESHOLD_LOBITS | [7:0] | PHY_PRBS_THRESHOLD_LOBITS | | | | | | | | 0x00 | R/W |
| 0x318 | PHY_PRBS_TEST_THRESHOLD_MIDBITS | [7:0] | PHY_PRBS_THRESHOLD_MIDBITS | | | | | | | | 0x00 | R/W |
| 0x319 | PHY_PRBS_TEST_THRESHOLD_HIBITS | [7:0] | PHY_PRBS_THRESHOLD_HIBITS | | | | | | | | 0x00 | R/W |
| 0x31A | PHY_PRBS_TEST_ERRCNT_LOBITS | [7:0] | PHY_PRBS_ERR_CNT_LOBITS | | | | | | | | 0x00 | R |
| 0x31B | PHY_PRBS_TEST_ERRCNT_MIDBITS | [7:0] | PHY_PRBS_ERR_CNT_MIDBITS | | | | | | | | 0x00 | R |
| 0x31C | PHY_PRBS_TEST_ERRCNT_HIBITS | [7:0] | PHY_PRBS_ERR_CNT_HIBITS | | | | | | | | 0x00 | R |
| 0x31D | PHY_PRBS_TEST_STATUS | [7:0] | PHY_PRBS_PASS | | | | | | | | 0xFF | R |
| 0x31E | PHY_DATA_SNAPSHOT_CTRL | [7:0] | RESERVED | | | PHY_GRAB_LANE_SEL | | | PHY_GRAB_MODE | PHY_GRAB_DATA | 0x00 | R/W |
| 0x31F | PHY_SNAPSHOT_DATA_BYTE0 | [7:0] | PHY_SNAPSHOT_DATA_BYTE0 | | | | | | | | 0x00 | R |
| 0x320 | PHY_SNAPSHOT_DATA_BYTE1 | [7:0] | PHY_SNAPSHOT_DATA_BYTE1 | | | | | | | | 0x00 | R |
| 0x321 | PHY_SNAPSHOT_DATA_BYTE2 | [7:0] | PHY_SNAPSHOT_DATA_BYTE2 | | | | | | | | 0x00 | R |
| 0x322 | PHY_SNAPSHOT_DATA_BYTE3 | [7:0] | PHY_SNAPSHOT_DATA_BYTE3 | | | | | | | | 0x00 | R |
| 0x323 | PHY_SNAPSHOT_DATA_BYTE4 | [7:0] | PHY_SNAPSHOT_DATA_BYTE4 | | | | | | | | 0x00 | R |
| 0x32C | SHORT_TPL_TEST_0 | [7:0] | SHORT_TPL_SP_SEL | | | | SHORT_TPL_M_SEL | | SHORT_TPL_TEST_RESET | SHORT_TPL_TEST_EN | 0x00 | R/W |
| 0x32D | SHORT_TPL_TEST_1 | [7:0] | SHORT_TPL_REF_SP_LSB | | | | | | | | 0x00 | R/W |
| 0x32E | SHORT_TPL_TEST_2 | [7:0] | SHORT_TPL_REF_SP_MSB | | | | | | | | 0x00 | R/W |
| 0x32F | SHORT_TPL_TEST_3 | [7:0] | RESERVED | | | | | | | SHORT_TPL_FAIL | 0x00 | R |
| 0x334 | JESD_BIT_INVERSE_CTRL | [7:0] | JESD_BIT_INVERSE | | | | | | | | 0x00 | R/W |
| 0x400 | DID_REG | [7:0] | DID_RD | | | | | | | | 0x00 | R |
| 0x401 | BID_REG | [7:0] | BID_RD | | | | | | | | 0x00 | R |
| 0x402 | LID0_REG | [7:0] | RESERVED | ADJDIR_RD | PHADJ_RD | LL_LID0 | | | | | 0x00 | R |
| 0x403 | SCR_L_REG | [7:0] | SCR_RD | RESERVED | | L_RD | | | | | 0x00 | R |
| 0x404 | F_REG | [7:0] | F_RD | | | | | | | | 0x00 | R |
| 0x405 | K_REG | [7:0] | RESERVED | | | K_RD | | | | | 0x00 | R |
| 0x406 | M_REG | [7:0] | M_RD | | | | | | | | 0x00 | R |
| 0x407 | CS_N_REG | [7:0] | CS_RD | | RESERVED | N_RD | | | | | 0x00 | R |
| 0x408 | NP_REG | [7:0] | SUBCLASSV_RD | | | NP_RD | | | | | 0x00 | R |
| 0x409 | S_REG | [7:0] | JESDV_RD | | | S_RD | | | | | 0x00 | R |
| 0x40A | HD_CF_REG | [7:0] | HD_RD | RESERVED | | CF_RD | | | | | 0x00 | R |
| 0x40B | RES1_REG | [7:0] | RES1_RD | | | | | | | | 0x00 | R |
| 0x40C | RES2_REG | [7:0] | RES2_RD | | | | | | | | 0x00 | R |
| 0x40D | CHECKSUM0_REG | [7:0] | LL_FCHK0 | | | | | | | | 0x00 | R |
| 0x40E | COMPSUM0_REG | [7:0] | LL_FCMP0 | | | | | | | | 0x00 | R |

| Reg. | Name | Bits | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Reset | RW |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x412 | LID1_REG | [7:0] | RESERVED | | | LL_LID1 | | | | | 0x00 | R |
| 0x415 | CHECKSUM1_REG | [7:0] | LL_FCHK1 | | | | | | | | 0x00 | R |
| 0x416 | COMPSUM1_REG | [7:0] | LL_FCMP1 | | | | | | | | 0x00 | R |
| 0x41A | LID2_REG | [7:0] | RESERVED | | | LL_LID2 | | | | | 0x00 | R |
| 0x41D | CHECKSUM2_REG | [7:0] | LL_FCHK2 | | | | | | | | 0x00 | R |
| 0x41E | COMPSUM2_REG | [7:0] | LL_FCMP2 | | | | | | | | 0x00 | R |
| 0x422 | LID3_REG | [7:0] | RESERVED | | | LL_LID3 | | | | | 0x00 | R |
| 0x425 | CHECKSUM3_REG | [7:0] | LL_FCHK3 | | | | | | | | 0x00 | R |
| 0x426 | COMPSUM3_REG | [7:0] | LL_FCMP3 | | | | | | | | 0x00 | R |
| 0x42A | LID4_REG | [7:0] | RESERVED | | | LL_LID4 | | | | | 0x00 | R |
| 0x42D | CHECKSUM4_REG | [7:0] | LL_FCHK4 | | | | | | | | 0x00 | R |
| 0x42E | COMPSUM4_REG | [7:0] | LL_FCMP4 | | | | | | | | 0x00 | R |
| 0x432 | LID5_REG | [7:0] | RESERVED | | | LL_LID5 | | | | | 0x00 | R |
| 0x435 | CHECKSUM5_REG | [7:0] | LL_FCHK5 | | | | | | | | 0x00 | R |
| 0x436 | COMPSUM5_REG | [7:0] | LL_FCMP5 | | | | | | | | 0x00 | R |
| 0x43A | LID6_REG | [7:0] | RESERVED | | | LL_LID6 | | | | | 0x00 | R |
| 0x43D | CHECKSUM6_REG | [7:0] | LL_FCHK6 | | | | | | | | 0x00 | R |
| 0x43E | COMPSUM6_REG | [7:0] | LL_FCMP6 | | | | | | | | 0x00 | R |
| 0x442 | LID7_REG | [7:0] | RESERVED | | | LL_LID7 | | | | | 0x00 | R |
| 0x445 | CHECKSUM7_REG | [7:0] | LL_FCHK7 | | | | | | | | 0x00 | R |
| 0x446 | COMPSUM7_REG | [7:0] | LL_FCMP7 | | | | | | | | 0x00 | R |
| 0x450 | ILS_DID | [7:0] | DID | | | | | | | | 0x00 | R/W |
| 0x451 | ILS_BID | [7:0] | BID | | | | | | | | 0x00 | R/W |
| 0x452 | ILS_LID0 | [7:0] | RESERVED | ADJDIR | PHADJ | LID0 | | | | | 0x00 | R/W |
| 0x453 | ILS_SCR_L | [7:0] | SCR | RESERVED | | L | | | | | 0x87 | R/W |
| 0x454 | ILS_F | [7:0] | F | | | | | | | | 0x00 | R |
| 0x455 | ILS_K | [7:0] | RESERVED | | | K | | | | | 0x1F | R/W |
| 0x456 | ILS_M | [7:0] | M | | | | | | | | 0x01 | R |
| 0x457 | ILS_CS_N | [7:0] | CS | | RESERVED | N | | | | | 0x0F | R |
| 0x458 | ILS_NP | [7:0] | SUBCLASSV | | | NP | | | | | 0x0F | R/W |
| 0x459 | ILS_S | [7:0] | JESDV | | | S | | | | | 0x01 | R/W |
| 0x45A | ILS_HD_CF | [7:0] | HD | RESERVED | | CF | | | | | 0x80 | R |
| 0x45B | ILS_RES1 | [7:0] | RES1 | | | | | | | | 0x00 | R/W |
| 0x45C | ILS_RES2 | [7:0] | RES2 | | | | | | | | 0x00 | R/W |
| 0x45D | ILS_CHECKSUM | [7:0] | FCHK0 | | | | | | | | 0x00 | R/W |
| 0x46C | LANE_DESKEW | [7:0] | ILD7 | ILS6 | ILD5 | ILD4 | ILD3 | ILD2 | ILD1 | ILD0 | 0x00 | R |
| 0x46D | BAD_DISPARITY | [7:0] | BDE7 | BDE6 | BDE5 | BDE4 | BDE3 | BDE2 | BDE1 | BDE0 | 0x00 | R |
| 0x46E | NOT_IN_TABLE | [7:0] | NIT7 | NIT6 | NIT5 | NIT4 | NIT3 | NIT2 | NIT1 | NIT0 | 0x00 | R |
| 0x46F | UNEXPECTED_ KCHAR | [7:0] | UEK7 | UEK6 | UEK5 | UEK4 | UEK3 | UEK2 | UEK1 | UEK0 | 0x00 | R |
| 0x470 | CODE_GRP_SYNC | [7:0] | CGS7 | CGS6 | CGS5 | CGS4 | CGS3 | CGS2 | CGS1 | CGS0 | 0x00 | R |
| 0x471 | FRAME_SYNC | [7:0] | FS7 | FS6 | FS5 | FS4 | FS3 | FS2 | FS1 | FS0 | 0x00 | R |
| 0x472 | GOOD_ CHECKSUM | [7:0] | CKS7 | CKS6 | CKS5 | CKS4 | CKS3 | CKS2 | CKS1 | CKS0 | 0x00 | R |
| 0x473 | INIT_LANE_SYNC | [7:0] | ILS7 | ILS6 | ILS5 | ILS4 | ILS3 | ILS2 | ILS1 | ILS0 | 0x00 | R |
| 0x475 | CTRLREG0 | [7:0] | RX_DIS | CHAR_REPL_ DIS | RESERVED | | SOFTRST | FORCESYNCREQ | RESERVED | REPL_FRM_ENA | 0x01 | R/W |
| 0x476 | CTRLREG1 | [7:0] | RESERVED | | | QUAL_RDERR | DEL_SCR | CGS_SEL | NO_ILAS | FCHK_N | 0x14 | R/W |
| 0x477 | CTRLREG2 | [7:0] | ILS_MODE | RESERVED | REPDATATEST | QUETESTERR | AR_ECNTR | RESERVED | | | 0x00 | R/W |
| 0x478 | KVAL | [7:0] | KSYNC | | | | | | | | 0x01 | R/W |

| Reg. | Name | Bits | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Reset | RW |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x47C | ERRORTHRES | [7:0] | ETH | | | | | | | | 0xFF | R/W |
| 0x47D | SYNC_ASSERT_MASK | [7:0] | RESERVED | | | | | SYNC_ASSERT_MASK | | | 0x07 | R/W |
| 0x480 | ECNT_CTRL0 | [7:0] | RESERVED | | ECNT_ENA0 | | | ECNT_RST0 | | | 0x3F | R/W |
| 0x481 | ECNT_CTRL1 | [7:0] | RESERVED | | ECNT_ENA1 | | | ECNT_RST1 | | | 0x3F | R/W |
| 0x482 | ECNT_CTRL2 | [7:0] | RESERVED | | ECNT_ENA2 | | | ECNT_RST2 | | | 0x3F | R/W |
| 0x483 | ECNT_CTRL3 | [7:0] | RESERVED | | ECNT_ENA3 | | | ECNT_RST3 | | | 0x3F | R/W |
| 0x484 | ECNT_CTRL4 | [7:0] | RESERVED | | ECNT_ENA4 | | | ECNT_RST4 | | | 0x3F | R/W |
| 0x485 | ECNT_CTRL5 | [7:0] | RESERVED | | ECNT_ENA5 | | | ECNT_RST5 | | | 0x3F | R/W |
| 0x486 | ECNT_CTRL6 | [7:0] | RESERVED | | ECNT_ENA6 | | | ECNT_RST6 | | | 0x3F | R/W |
| 0x487 | ECNT_CTRL7 | [7:0] | RESERVED | | ECNT_ENA7 | | | ECNT_RST7 | | | 0x3F | R/W |
| 0x488 | ECNT_TCH0 | [7:0] | RESERVED | | | | | ECNT_TCH0 | | | 0x07 | R/W |
| 0x489 | ECNT_TCH1 | [7:0] | RESERVED | | | | | ECNT_TCH1 | | | 0x07 | R/W |
| 0x48A | ECNT_TCH2 | [7:0] | RESERVED | | | | | ECNT_TCH2 | | | 0x07 | R/W |
| 0x48B | ECNT_TCH3 | [7:0] | RESERVED | | | | | ECNT_TCH3 | | | 0x07 | R/W |
| 0x48C | ECNT_TCH4 | [7:0] | RESERVED | | | | | ECNT_TCH4 | | | 0x07 | R/W |
| 0x48D | ECNT_TCH5 | [7:0] | RESERVED | | | | | ECNT_TCH5 | | | 0x07 | R/W |
| 0x48E | ECNT_TCH6 | [7:0] | RESERVED | | | | | ECNT_TCH6 | | | 0x07 | R/W |
| 0x48F | ECNT_TCH7 | [7:0] | RESERVED | | | | | ECNT_TCH7 | | | 0x07 | R/W |
| 0x490 | ECNT_STAT0 | [7:0] | RESERVED | | | | LANE_ENA0 | ECNT_TCR0 | | | 0x00 | R |
| 0x491 | ECNT_STAT1 | [7:0] | RESERVED | | | | LANE_ENA1 | ECNT_TCR1 | | | 0x00 | R |
| 0x492 | ECNT_STAT2 | [7:0] | RESERVED | | | | LANE_ENA2 | ECNT_TCR2 | | | 0x00 | R |
| 0x493 | ECNT_STAT3 | [7:0] | RESERVED | | | | LANE_ENA3 | ECNT_TCR3 | | | 0x00 | R |
| 0x494 | ECNT_STAT4 | [7:0] | RESERVED | | | | LANE_ENA4 | ECNT_TCR4 | | | 0x00 | R |
| 0x495 | ECNT_STAT5 | [7:0] | RESERVED | | | | LANE_ENA5 | ECNT_TCR5 | | | 0x00 | R |
| 0x496 | ECNT_STAT6 | [7:0] | RESERVED | | | | LANE_ENA6 | ECNT_TCR6 | | | 0x00 | R |
| 0x497 | ECNT_STAT7 | [7:0] | RESERVED | | | | LANE_ENA7 | ECNT_TCR7 | | | 0x00 | R |
| 0x498 | BD_CNT0 | [7:0] | BD_CNT0 | | | | | | | | 0x00 | R |
| 0x499 | BD_CNT1 | [7:0] | BD_CNT1 | | | | | | | | 0x00 | R |
| 0x49A | BD_CNT2 | [7:0] | BD_CNT2 | | | | | | | | 0x00 | R |
| 0x49B | BD_CNT3 | [7:0] | BD_CNT3 | | | | | | | | 0x00 | R |
| 0x49C | BD_CNT4 | [7:0] | BD_CNT4 | | | | | | | | 0x00 | R |
| 0x49D | BD_CNT5 | [7:0] | BD_CNT5 | | | | | | | | 0x00 | R |
| 0x49E | BD_CNT6 | [7:0] | BD_CNT6 | | | | | | | | 0x00 | R |
| 0x49F | BD_CNT7 | [7:0] | BD_CNT7 | | | | | | | | 0x00 | R |
| 0x4A0 | NIT_CNT0 | [7:0] | NIT_CNT0 | | | | | | | | 0x00 | R |
| 0x4A1 | NIT_CNT1 | [7:0] | NIT_CNT1 | | | | | | | | 0x00 | R |
| 0x4A2 | NIT_CNT2 | [7:0] | NIT_CNT2 | | | | | | | | 0x00 | R |
| 0x4A3 | NIT_CNT3 | [7:0] | NIT_CNT3 | | | | | | | | 0x00 | R |
| 0x4A4 | NIT_CNT4 | [7:0] | NIT_CNT4 | | | | | | | | 0x00 | R |
| 0x4A5 | NIT_CNT5 | [7:0] | NIT_CNT5 | | | | | | | | 0x00 | R |
| 0x4A6 | NIT_CNT6 | [7:0] | NIT_CNT6 | | | | | | | | 0x00 | R |
| 0x4A7 | NIT_CNT7 | [7:0] | NIT_CNT7 | | | | | | | | 0x00 | R |
| 0x4A8 | UEK_CNT0 | [7:0] | UEK_CNT0 | | | | | | | | 0x00 | R |
| 0x4A9 | UEK_CNT1 | [7:0] | UEK_CNT1 | | | | | | | | 0x00 | R |
| 0x4AA | UEK_CNT2 | [7:0] | UEK_CNT2 | | | | | | | | 0x00 | R |
| 0x4AB | UEK_CNT3 | [7:0] | UEK_CNT3 | | | | | | | | 0x00 | R |
| 0x4AC | UEK_CNT4 | [7:0] | UEK_CNT4 | | | | | | | | 0x00 | R |
| 0x4AD | UEK_CNT5 | [7:0] | UEK_CNT5 | | | | | | | | 0x00 | R |

| Reg. | Name | Bits | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Reset | RW |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x4AE | UEK_CNT6 | [7:0] | UEK_CNT6 | | | | | | | | 0x00 | R |
| 0x4AF | UEK_CNT7 | [7:0] | UEK_CNT7 | | | | | | | | 0x00 | R |
| 0x4B0 | LINK_STATUS0 | [7:0] | BDE0 | NIT0 | UEK0 | ILD0 | ILS0 | CKS0 | FS0 | CGS0 | 0x00 | R |
| 0x4B1 | LINK_STATUS1 | [7:0] | BDE1 | NIT1 | UEK1 | ILD1 | ILS1 | CKS1 | FS1 | CGS1 | 0x00 | R |
| 0x4B2 | LINK_STATUS2 | [7:0] | BDE2 | NIT2 | UEK2 | ILD2 | ILS2 | CKS2 | FS2 | CGS2 | 0x00 | R |
| 0x4B3 | LINK_STATUS3 | [7:0] | BDE3 | NIT3 | UEK3 | ILD3 | ILS3 | CKS3 | FS3 | CGS3 | 0x00 | R |
| 0x4B4 | LINK_STATUS4 | [7:0] | BDE4 | NIT4 | UEK4 | ILD4 | ILS4 | CKS4 | FS4 | CGS4 | 0x00 | R |
| 0x4B5 | LINK_STATUS5 | [7:0] | BDE5 | NIT5 | UEK5 | ILD5 | ILS5 | CKS5 | FS5 | CGS5 | 0x00 | R |
| 0x4B6 | LINK_STATUS6 | [7:0] | BDE6 | NIT6 | UEK6 | ILD6 | ILS6 | CKS6 | FS6 | CGS6 | 0x00 | R |
| 0x4B7 | LINK_STATUS7 | [7:0] | BDE7 | NIT7 | UEK7 | ILD7 | ILS7 | CKS7 | FS7 | CGS7 | 0x00 | R |
| 0x4B8 | JESD_IRQ_ ENABLEA | [7:0] | EN_BDE | EN_NIT | EN_UEK | EN_ILD | EN_ILS | EN_CKS | EN_FS | EN_CGS | 0x00 | R/W |
| 0x4B9 | JESD_IRQ_ ENABLEB | [7:0] | RESERVED | | | | | | | EN_ILAS | 0x00 | R/W |
| 0x4BA | JESD_IRQ_ STATUSA | [7:0] | IRQ_BDE | IRQ_NIT | IRQ_UEK | IRQ_ILD | IRQ_ILS | IRQ_CKS | IRQ_FS | IRQ_CGS | 0x00 | R/W |
| 0x4BB | JESD_IRQ_ STATUSB | [7:0] | RESERVED | | | | | | | IRQ_ILAS | 0x00 | R/W |
| 0x800 | HOPF_CTRL | [7:0] | HOPF_MODE | | RESERVED | | | | | | 0x00 | R/W |

# REGISTER DETAILS

**Table 45. Register Details**

| Hex. Addr. | Name | Bits | Bit Name | Settings | Description | Reset | Access |
|---|---|---|---|---|---|---|---|
| 0x000 | SPI_INTFCONFA | 7 | SOFTRESET_M | | Soft reset (mirror). Set this to mirror Bit 0. | 0x0 | R |
| | | 6 | LSBFIRST_M | | LSB first (mirror). Set this to mirror Bit 1. | 0x0 | R |
| | | 5 | ADDRINC_M | | Address increment (mirror). Set this to mirror Bit 2. | 0x0 | R |
| | | 4 | SDOACTIVE_M | | SDO active (mirror). Set this to mirror Bit 3. | 0x0 | R |
| | | 3 | SDOACTIVE | | SDO active. Enables 4-wire SPI bus mode. | 0x0 | R/W |
| | | 2 | ADDRINC | | Address increment. When set, causes incrementing streaming addresses; otherwise, descending addresses are generated. | 0x0 | R/W |
| | | | | 1 | Streaming addresses are incremented. | | |
| | | | | 0 | Streaming addresses are decremented. | | |
| | | 1 | LSBFIRST | | LSB first. When set, causes input and output data to be oriented as LSB first. If this bit is clear, data is oriented as MSB first. | 0x0 | R/W |
| | | | | 1 | Shift LSB in first. | | |
| | | | | 0 | Shift MSB in first. | | |
| | | 0 | SOFTRESET | | Soft reset. This bit automatically clears to 0 after performing a reset operation. Setting this bit initiates a reset. This bit is autoclearing after the soft reset is complete. | 0x0 | R/W |
| | | | | 1 | Pulse the soft reset line. | | |
| | | | | 0 | Reset the soft reset line. | | |
| 0x001 | SPI_INTFCONFB | 7 | SINGLEINS | | Single instruction. | 0x0 | R/W |
| | | | | 1 | Perform single transfers. | | |
| | | | | 0 | Perform multiple transfers. | | |
| | | 6 | $\overline{\text{CS}}$STALL | | $\overline{\text{CS}}$ stalling. | 0x0 | R/W |
| | | | | 0 | Disable $\overline{\text{CS}}$ stalling. | | |
| | | | | 1 | Enable $\overline{\text{CS}}$ stalling. | | |
| | | [5:3] | RESERVED | | Reserved. | 0x0 | R/W |
| | | 2 | SOFTRESET1 | | Soft Reset 1. This bit automatically clears to 0 after performing a reset operation. | 0x0 | R/W |
| | | | | 1 | Pulse the Soft Reset 1 line. | | |
| | | | | 0 | Pulse the Soft Reset 1 line. | | |
| | | 1 | SOFTRESET0 | | Soft Reset 0. This bit automatically clears to 0 after performing a reset operation. | 0x0 | R/W |
| | | | | 1 | Pulse the Soft Reset 0 line. | | |
| | | | | 0 | Pulse the Soft Reset 0 line. | | |
| | | 0 | RESERVED | | Reserved. | 0x0 | R |
| 0x002 | SPI_DEVCONF | [7:4] | DEVSTATUS | | Device status. | 0x0 | R/W |
| | | [3:2] | CUSTOPMODE | | Customer operating mode. | 0x0 | R/W |
| | | [1:0] | SYSOPMODE | | System operating mode. | 0x0 | R/W |
| | | | | 0 | Normal operation. | | |
| | | | | 1 | Low power operation. | | |

| Hex. Addr. | Name | Bits | Bit Name | Settings | Description | Reset | Access |
|---|---|---|---|---|---|---|---|
| | | | | 2 | Medium power standby. | | |
| | | | | 3 | Low power sleep. | | |
| 0x003 | SPI_CHIPTYPE | [7:0] | CHIP_TYPE | | Chip type. | 0x0 | R |
| 0x004 | SPI_PRODIDL | [7:0] | PROD_ID[7:0] | | Product ID. | 0x0 | R |
| 0x005 | SPI_PRODIDH | [7:0] | PROD_ID[15:8] | | Product ID. | 0x0 | R |
| 0x006 | SPI_CHIPGRADE | [7:4] | PROD_GRADE | | Product grade. | 0x0 | R |
| | | [3:0] | DEV_REVISION | | Device revision. | 0x0 | R |
| 0x020 | IRQ_ENABLE | [7:5] | RESERVED | | Reserved. | 0x0 | R |
| | | 4 | EN_SYSREF_JITTER | | Enable SYSREF± jitter interrupt. | 0x0 | R/W |
| | | | | 0 | Disable interrupt. | | |
| | | | | 1 | Enable interrupt. | | |
| | | 3 | EN_DATA_READY | | Enable JESD204x receiver ready (JRX_DATA_READY) low interrupt. | 0x0 | R/W |
| | | | | 0 | Disable interrupt. | | |
| | | | | 1 | Enable interrupt. | | |
| | | 2 | EN_LANE_FIFO | | Enable lane FIFO overflow/ underflow interrupt. | 0x0 | R/W |
| | | | | 0 | Disable interrupt. | | |
| | | | | 1 | Enable interrupt. | | |
| | | 1 | EN_PRBSQ | | Enable PRBS imaginary error interrupt. | 0x0 | R/W |
| | | | | 0 | Disable interrupt. | | |
| | | | | 1 | Enable interrupt. | | |
| | | 0 | EN_PRBSI | | Enable PRBS real error interrupt. | 0x0 | R/W |
| | | | | 0 | Disable interrupt. | | |
| | | | | 1 | Enable interrupt. | | |
| 0x024 | IRQ_STATUS | [7:5] | RESERVED | | Reserved. | 0x0 | R |
| | | 4 | IRQ_SYSREF_JITTER | | SYSREF± jitter is too big. Writing 1 clears the status. | 0x0 | R/W |
| | | 3 | IRQ_DATA_READY | | JRX_DATA_READY is low. Writing 1 clears the status. | 0x0 | R/W |
| | | | | 0 | No warning. | | |
| | | | | 1 | Warning detected. | | |
| | | 2 | IRQ_LANE_FIFO | | Lane FIFO overflow/underflow. Writing 1 clears the status. | 0x0 | R/W |
| | | | | 0 | No warning. | | |
| | | | | 1 | Warning detected. | | |
| | | 1 | IRQ_PRBSQ | | PRBS imaginary error. Writing 1 clears the status. | 0x0 | R/W |
| | | | | 0 | No warning. | | |
| | | | | 1 | Warning detected. | | |
| | | 0 | IRQ_PRBSI | | PRBS real error. Writing 1 clears the status. | 0x0 | R/W |
| | | | | 0 | No warning. | | |
| | | | | 1 | Warning detected. | | |
| 0x031 | SYNC_LMFC_DELAY_FRAME | [7:5] | RESERVED | | Reserved. | 0x0 | R |
| | | [4:0] | SYNC_LMFC_DELAY_SET_FRM | | Desired delay from rising edge of SYSREF± input to rising edge of LMFC in frames. | 0x0 | R/W |
| 0x032 | SYNC_LMFC_DELAY0 | [7:0] | SYNC_LMFC_DELAY_SET[7:0] | | Desired delay from rising edge of SYSREF± input to rising edge of LMFC in DAC clock units. | 0x0 | R/W |
| 0x033 | SYNC_LMFC_DELAY1 | [7:4] | RESERVED | | Reserved. | 0x0 | R |
| | | [3:0] | SYNC_LMFC_DELAY_SET[11:8] | | Desired delay from rising edge of SYSREF± input to rising edge of LMFC in DAC clock units. | 0x0 | R/W |

| Hex. Addr. | Name | Bits | Bit Name | Settings | Description | Reset | Access |
|---|---|---|---|---|---|---|---|
| 0x034 | SYNC_LMFC_STAT0 | [7:0] | SYNC_LMFC_DELAY_STAT[7:0] | | Measured delay from rising edge of SYSREF± input to rising edge of LMFC in DAC clock units (note: 2 LSBs are always zero). A write to SYNC_LMFC_STATx or SYSREF_PHASEx saves the data for readback. | 0x0 | R/W |
| 0x035 | SYNC_LMFC_STAT1 | [7:4] | RESERVED | | Reserved. | 0x0 | R |
| | | [3:0] | SYNC_LMFC_DELAY_STAT[11:8] | | Measured delay from rising edge of SYSREF± input to rising edge of LMFC in DAC clock units (note: 2 LSBs are always zero). A write to SYNC_LMFC_STATx or SYSREF_PHASEx saves the data for readback. | 0x0 | R/W |
| 0x036 | SYSREF_COUNT | [7:0] | SYSREF_COUNT | | Count of SYSREF± signals received. A write resets the count. A write to SYNC_LMFC_STATx or SYSREF_PHASEx saves the data for readback. | 0x0 | R/W |
| 0x037 | SYSREF_PHASE0 | [7:0] | SYSREF_PHASE[7:0] | | Phase of measured SYSREF± event. Thermometer encoded. A write to SYNC_LMFC_STATx or SYSREF_PHASEx saves the data for readback. | 0x0 | R/W |
| 0x038 | SYSREF_PHASE1 | [7:4] | RESERVED | | Reserved. | 0x0 | R |
| | | [3:0] | SYSREF_PHASE[11:8] | | Phase of measured SYSREF± event. Thermometer encoded. A write to SYNC_LMFC_STATx or SYSREF_PHASEx saves the data for readback. | 0x0 | R/W |
| 0x039 | SYSREF_JITTER_WINDOW | [7:6] | RESERVED | | Reserved. | 0x0 | R |
| | | [5:0] | SYSREF_JITTER_WINDOW | | Amount of jitter allowed on the SYSREF± input. SYSREF± jitter variations bigger than this triggers an interrupt. Units are in DAC clocks. The bottom two bits are ignored. | 0x0 | R/W |
| 0x03A | SYNC_CTRL | [7:2] | RESERVED | | Reserved. | 0x0 | R |
| | | [1:0] | SYNC_MODE | | Synchronization mode. | 0x0 | R/W |
| | | | | 00 | Do not perform synchronization, monitor SYSREF± to LMFC delay only. | | |
| | | | | 01 | Perform continuous synchronization of LMFC on every SYSREF±. | | |
| | | | | 10 | Perform a single synchronization on the next SYSREF±, then switch to monitor mode. | | |
| 0x03F | TX_ENABLE | 7 | SPI_DATAPATH_POST | | SPI control of the data at the output of the datapath. | 0x1 | R/W |
| | | | | 0 | Disable or zero the data from the datapath into the DAC. | | |
| | | | | 1 | Use the data from the datapath to drive the DAC. | | |
| | | 6 | SPI_DATAPATH_PRE | | SPI control of the data at the input of the datapath. | 0x1 | R/W |
| | | | | 0 | Disable or zero the data feeding into the datapath. | | |
| | | | | 1 | Use the data from the JESD204B lanes to drive into the datapath. | | |

| Hex. Addr. | Name | Bits | Bit Name | Settings | Description | Reset | Access |
|---|---|---|---|---|---|---|---|
| | | [5:4] | RESERVED | | Reserved. | 0x0 | R |
| | | 3 | TXEN_NCO_RESET | | Allows TX_ENABLE to control the DDS NCO reset. | 0x0 | R/W |
| | | | | 0 | Use the SPI (HOPF_MODE SPI bits) to control the DDS NCO reset. | | |
| | | | | 1 | Use the TX_ENABLE pin to control the DDS NCO reset. | | |
| | | 2 | TXEN_DATAPATH_POST | | Allows TX_ENABLE to control the data at the output of the datapath. | 0x0 | R/W |
| | | | | 0 | Use the SPI (Bit SPI_DATAPATH_POST) for control. | | |
| | | | | 1 | Use the TX_ENABLE pin for control. | | |
| | | 1 | TXEN_DATAPATH_PRE | | Allows TX_ENABLE to control the data at the input of the datapath. | 0x0 | R/W |
| | | | | 0 | Use the SPI (Bit SPI_DATAPATH_PRE) for control. | | |
| | | | | 1 | Use the TX_ENABLE pin for control. | | |
| | | 0 | TXEN_DAC_FSC | | Allows TX_ENABLE to control the DAC full-scale current. | 0x0 | R/W |
| | | | | 0 | Use the SPI registers, ANA_FSC0 and ANA_FSC1, for control. | | |
| | | | | 1 | Use the TX_ENABLE pin for control. | | |
| 0x040 | ANA_DAC_BIAS_PD | [7:2] | RESERVED | | Reserved. | 0x0 | R |
| | | 1 | ANA_DAC_BIAS_PD1 | | Powers down the DAC core bias circuits. A 1 powers down the DAC core bias circuits. | 0x1 | R/W |
| | | 0 | ANA_DAC_BIAS_PD0 | | Powers down the DAC core bias circuits. A 1 powers down the DAC core bias circuits. | 0x1 | R/W |
| 0x041 | ANA_FSC0 | [7:2] | RESERVED | | Reserved. | 0x0 | R |
| | | [1:0] | ANA_FULL_SCALE_CURRENT[1:0] | | DAC full-scale current. Analog full-scale current adjustment. | 0x3 | R/W |
| 0x042 | ANA_FSC1 | [7:0] | ANA_FULL_SCALE_CURRENT[9:2] | | DAC full-scale current. Analog full-scale current adjustment. | 0xFF | R/W |
| 0x07F | CLK_PHASE_TUNE | [7:6] | RESERVED | | Reserved. | 0x0 | R |
| | | [5:0] | CLK_PHASE_TUNE | | Fine tuning of the clock input phase balance. Adds small capacitors to the CLK+/CLK– inputs, ~20 fF per step, signed magnitude. | 0x0 | R/W |

| | Capacitance | |
|---|---|---|
| Bits[5:0] | At CLK+ | At CLK– |
| 000000 | 0 | 0 |
| 000001 | 1 | 0 |
| 000010 | 2 | 0 |
| … | … | … |
| 011111 | 31 | 0 |
| 100000 | 0 | 0 |
| 100001 | 0 | 1 |
| 100010 | 0 | 2 |
| 111111 | 0 | 31 |

| Hex. Addr. | Name | Bits | Bit Name | Settings | Description | Reset | Access |
|---|---|---|---|---|---|---|---|
| 0x080 | CLK_PD | [7:1] | RESERVED | | Reserved. | 0x0 | R |
| | | 0 | DACCLK_PD | | DAC clock power-down. Powers down the DAC clock circuitry. | 0x1 | R/W |
| | | | | 0 | Power up. | | |
| | | | | 1 | Power down. | | |
| 0x082 | CLK_DUTY | 7 | CLK_DUTY_EN | | Enable duty cycle control. | 0x1 | R/W |
| | | 6 | CLK_DUTY_OFFSET_EN | | Enable duty cycle offset. | 0x0 | R/W |
| | | 5 | CLK_DUTY_BOOST_EN | | Enable duty cycle range boost. Extends range to ±5% at cost of 1 dB to 2 dB worse phase noise. | 0x0 | R/W |
| | | [4:0] | CLK_DUTY_PRG | | Program the duty cycle offset. 5-bit signed magnitude field, with the MSB as the sign bit and the four LSBs as the magnitude from 0 to 15. A larger magnitude skews duty cycle to a greater amount. Range is ±3%. | 0x0 | R/W |
| 0x083 | CLK_CRS_CTRL | 7 | CLK_CRS_EN | | Enable clock cross control adjustment. | 0x1 | R/W |
| | | [6:4] | RESERVED | | Reserved. | 0x0 | R |
| | | [3:0] | CLK_CRS_ADJ | | Program the clock crossing point. | 0x0 | R/W |
| 0x084 | PLL_REF_CLK_PD | [7:6] | RESERVED | | Reserved. | 0x0 | R |
| | | [5:4] | PLL_REF_CLK_RATE | | PLL reference clock rate multiplier. | 0x0 | R/W |
| | | | | 00 | Normal rate (1×) PLL reference clock. | | |
| | | | | 01 | Double rate (2×) PLL reference clock. | | |
| | | | | 10 | Quadruple rate (4×) PLL reference clock. | | |
| | | | | 11 | Disable the PLL reference clock. | | |
| | | [3:1] | RESERVED | | Reserved. | 0x0 | R |
| | | 0 | PLL_REF_CLK_PD | | PLL reference clock power-down. | 0x0 | R/W |
| | | | | 0 | Enable the PLL reference clock. | | |
| | | | | 1 | Power down the PLL reference clock. | | |
| 0x088 | SYSREF_CTRL0 | [7:4] | RESERVED | | Reserved. | 0x0 | R |
| | | 3 | HYS_ON | | SYSREF± hysteresis enable. This bit enables the programmable hysteresis control for the SYSREF± receiver. | 0x0 | R/W |
| | | 2 | SYSREF_RISE | | Use SYSREF± rising edge. | 0x0 | R/W |
| | | [1:0] | HYS_CNTRL[9:8] | | Controls the amount of hysteresis in the SYSREF± receiver. Each of the 10 bits adds 10 mV of differential hysteresis to the receiver input. | 0x0 | R/W |
| 0x089 | SYSREF_CTRL1 | [7:0] | HYS_CNTRL[7:0] | | Controls the amount of hysteresis in the SYSREF± receiver. Each of the 10 bits adds 10 mV of differential hysteresis to the receiver input. | 0x0 | R/W |
| 0x090 | DLL_PD | [7:5] | RESERVED | | Reserved. | 0x0 | R |
| | | 4 | DLL_FINE_DC_EN | | Fine delay line duty cycle correction enable. | 0x1 | R/W |
| | | 3 | DLL_FINE_XC_EN | | Fine delay line cross control enable. | 0x1 | R/W |
| | | 2 | DLL_COARSE_DC_EN | | Coarse delay line duty cycle correction enable. | 0x1 | R/W |
| | | 1 | DLL_COARSE_XC_EN | | Coarse delay line cross control enable. | 0x1 | R/W |

| Hex. Addr. | Name | Bits | Bit Name | Settings | Description | Reset | Access |
|---|---|---|---|---|---|---|---|
| | | 0 | DLL_CLK_PD | | Power down DLL and digital clock generator. | 0x1 | R/W |
| | | | | 0 | Power up DLL controller. | | |
| | | | | 1 | Power down DLL controller. | | |
| 0x091 | DLL_CTRL | 7 | DLL_TRACK_ERR | | Track error behavior. | 0x1 | R/W |
| | | | | 0 | Continue on error. | | |
| | | | | 1 | Restart on error. | | |
| | | 6 | DLL_SEARCH_ERR | | Search error behavior. | 0x1 | R/W |
| | | | | 0 | Stop on error. | | |
| | | | | 1 | Retry on error. | | |
| | | 5 | DLL_SLOPE | | Desired slope. | 0x1 | R/W |
| | | | | 0 | Negative slope. | | |
| | | | | 1 | Positive slope. | | |
| | | [4:3] | DLL_SEARCH | | Search direction. | 0x2 | R/W |
| | | | | 00 | Search down from initial point only. | | |
| | | | | 01 | Search up from initial point only. | | |
| | | | | 10 | Search up and down from initial point. | | |
| | | [2:1] | DLL_MODE | | Controller mode. | 0x0 | R/W |
| | | | | 00 | Search then track. | | |
| | | | | 01 | Track only. | | |
| | | | | 10 | Search only. | | |
| | | 0 | DLL_ENABLE | | Controller enable. | 0x0 | R/W |
| | | | | 0 | Disable DLL controller: use static SPI settings. | | |
| | | | | 1 | Enable DLL controller: use controller with feedback loop. | | |
| 0x092 | DLL_STATUS | [7:3] | RESERVED | | Reserved. | 0x0 | R |
| | | 2 | DLL_FAIL | | The DAC clock DLL failed to lock. | 0x0 | R |
| | | 1 | DLL_LOST | | The DAC clock DLL has lost lock. | 0x0 | R/W |
| | | 0 | DLL_LOCKED | | The DAC clock DLL has achieved lock. | 0x0 | R |
| 0x093 | DLL_GB | [7:4] | RESERVED | | Reserved. | 0x0 | R |
| | | [3:0] | DLL_GUARD | | Search guard band. | 0x0 | R/W |
| 0x094 | DLL_COARSE | [7:6] | RESERVED | | Reserved. | 0x0 | R |
| | | [5:0] | DLL_COARSE | | Coarse delay line setpoint. | 0x0 | R/W |
| 0x095 | DLL_FINE | [7:0] | DLL_FINE | | Fine delay line setpoint. | 0x80 | R/W |
| 0x096 | DLL_PHASE | [7:5] | RESERVED | | Reserved. | 0x0 | R |
| | | [4:0] | DLL_PHS | | Desired phase. | 0x8 | R/W |
| | | | | 0 | Minimum allowed phase. | | |
| | | | | 16 | Maximum allowed phase. | | |
| 0x097 | DLL_BW | [7:5] | RESERVED | | Reserved. | 0x0 | R |
| | | [4:2] | DLL_FILT_BW | | Phase measurement filter bandwidth. | 0x0 | R/W |
| | | [1:0] | DLL_WEIGHT | | Tracking speed. | 0x0 | R/W |
| 0x098 | DLL_READ | [7:1] | RESERVED | | Reserved. | 0x0 | R |
| | | 0 | DLL_READ | | Read request: 0 to 1 transition updates the coarse, fine, and phase readback values. | 0x0 | R/W |
| 0x099 | DLL_COARSE_RB | [7:6] | RESERVED | | Reserved. | 0x0 | R |
| | | [5:0] | DLL_COARSE_RB | | Coarse delay line readback. | 0x0 | R |
| 0x09A | DLL_FINE_RB | [7:0] | DLL_FINE_RB | | Fine delay line readback. | 0x0 | R |
| 0x09B | DLL_PHASE_RB | [7:5] | RESERVED | | Reserved. | 0x0 | R |
| | | [4:0] | DLL_PHS_RB | | Phase readback. | 0x0 | R |

| Hex. Addr. | Name | Bits | Bit Name | Settings | Description | Reset | Access |
|---|---|---|---|---|---|---|---|
| 0x09D | DIG_CLK_INVERT | [7:3] | RESERVED | | Reserved. | 0x0 | R |
| | | 2 | INV_DIG_CLK | | Invert digital clock from DLL. | 0x0 | R/W |
| | | | | 0 | Normal polarity. | | |
| | | | | 1 | Inverted polarity. | | |
| | | 1 | DIG_CLK_DC_EN | | Digital clock duty cycle correction enable. | 0x1 | R/W |
| | | 0 | DIG_CLK_XC_EN | | Digital clock cross control enable. | 0x1 | R/W |
| 0x0A0 | DLL_CLK_DEBUG | 7 | DLL_TEST_EN | | DLL clock output test enable. | 0x0 | R/W |
| | | [6:2] | RESERVED | | Reserved. | 0x0 | R |
| | | [1:0] | DLL_TEST_DIV | | DLL clock output divide. | 0x0 | R/W |
| 0x110 | INTERP_MODE | [7:4] | JESD_LANES | | Number of JESD204B lanes. For proper operation of the JESD204B data link, this signal must only be programmed while the QBD is held in soft reset (Register 0x475, Bit 3), and must not be changed during normal operation. | 0x8 | R/W |
| | | [3:0] | INTERP_MODE | | Interpolation mode. For proper operation of the JESD204B data link, this signal must only be programmed while the QBD is held in soft reset (Register 0x475, Bit 3), and must not be changed during normal operation. | 0x1 | R/W |
| | | | | 0000 | Reserved. | | |
| | | | | 0001 | Reserved. | | |
| | | | | 0010 | Reserved. | | |
| | | | | 0011 | Reserved. | | |
| | | | | 0100 | 6×. | | |
| | | | | 0101 | 8×. | | |
| | | | | 0110 | 12×. | | |
| | | | | 0111 | 16×. | | |
| | | | | 1000 | 24×. | | |
| 0x111 | DATAPATH_CFG | 7 | INVSINC_EN | | Inverse sinc filter enable. | 0x0 | R/W |
| | | | | 0 | Disable inverse sinc filter. | | |
| | | | | 1 | Enable inverse sinc filter. | | |
| | | 6 | NCO_EN | | Modulation enable. | 0x0 | R/W |
| | | | | 0 | Disable NCO. | | |
| | | | | 1 | Enable NCO. | | |
| | | 5 | RESERVED | | Reserved. | 0x0 | R |
| | | 4 | FILT_BW | | Datapath filter bandwidth. | 0x0 | R/W |
| | | | | 0 | Filter bandwidth is 80%. | | |
| | | | | 1 | Filter bandwidth is 90%. | | |
| | | 3 | RESERVED | | Reserved. | 0x0 | R |
| | | 2 | MODULUS_EN | | Modulus DDS enable | 0x0 | R/W |
| | | | | 0 | Disable modulus DDS. | | |
| | | | | 1 | Enable modulus DDS. | | |
| | | 1 | SEL_SIDEBAND | | Selects upper or lower sideband from modulation result. | 0x0 | R/W |
| | | | | 0 | Use upper sideband. | | |
| | | | | 1 | Use lower sideband = spectral flip. | | |
| | | 0 | FIR85_FILT_EN | | FIR85 filter enable. | 0x0 | R/W |

| Hex. Addr. | Name | Bits | Bit Name | Settings | Description | Reset | Access |
|---|---|---|---|---|---|---|---|
| 0x113 | FTW_UPDATE | 7 | RESERVED | | Reserved. | 0x0 | R |
| | | [6:4] | FTW_REQ_MODE | | Frequency tuning word automatic update mode. | 0x0 | R/W |
| | | | | 000 | No automatic requests are generated when the FTW registers are written. | | |
| | | | | 001 | Automatically generate FTW_LOAD_REQ after FTW0 is written. | | |
| | | | | 010 | Automatically generate FTW_LOAD_REQ after FTW1 is written. | | |
| | | | | 011 | Automatically generate FTW_LOAD_REQ after FTW2 is written. | | |
| | | | | 100 | Automatically generate FTW_LOAD_REQ after FTW3 is written. | | |
| | | | | 101 | Automatically generate FTW_LOAD_REQ after FTW4 is written. | | |
| | | | | 110 | Automatically generate FTW_LOAD_REQ after FTW5 is written. | | |
| | | 3 | RESERVED | | Reserved. | 0x0 | R |
| | | 2 | FTW_LOAD_SYSREF | | FTW load and reset from rising edge of SYSREF±. | 0x0 | R/W |
| | | 1 | FTW_LOAD_ACK | | Frequency tuning word update acknowledge. | 0x0 | R |
| | | | | 0 | FTW is not loaded. | | |
| | | | | 1 | FTW is loaded. | | |
| | | 0 | FTW_LOAD_REQ | | Frequency tuning word update request from SPI. | 0x0 | R/W |
| | | | | 0 | Clear FTW_LOAD_ACK. | | |
| | | | | 1 | 0 to 1 transition loads the FTW. | | |
| 0x114 | FTW0 | [7:0] | FTW[7:0] | | NCO frequency tuning word. This is X in the equation $f_{OUT} = f_{DAC} \times (M/N) = f_{DAC} \times ((X + A/B)/2^{48})$. | 0x0 | R/W |
| 0x115 | FTW1 | [7:0] | FTW[15:8] | | NCO frequency tuning word. This is X in the equation $f_{OUT} = f_{DAC} \times (M/N) = f_{DAC} \times ((X + A/B)/2^{48})$. | 0x0 | R/W |
| 0x116 | FTW2 | [7:0] | FTW[23:16] | | NCO frequency tuning word. This is X in the equation $f_{OUT} = f_{DAC} \times (M/N) = f_{DAC} \times ((X + A/B)/2^{48})$. | 0x0 | R/W |
| 0x117 | FTW3 | [7:0] | FTW[31:24] | | NCO frequency tuning word. This is X in the equation $f_{OUT} = f_{DAC} \times (M/N) = f_{DAC} \times ((X + A/B)/2^{48})$. | 0x0 | R/W |
| 0x118 | FTW4 | [7:0] | FTW[39:32] | | NCO frequency tuning word. This is X in the equation $f_{OUT} = f_{DAC} \times (M/N) = f_{DAC} \times ((X + A/B)/2^{48})$. | 0x0 | R/W |
| 0x119 | FTW5 | [7:0] | FTW[47:40] | | NCO frequency tuning word. This is X in the equation $f_{OUT} = f_{DAC} \times (M/N) = f_{DAC} \times ((X + A/B)/2^{48})$. | 0x0 | R/W |
| 0x11C | PHASE_OFFSET0 | [7:0] | NCO_PHASE_OFFSET[7:0] | | NCO phase offset. | 0x0 | R/W |
| 0x11D | PHASE_OFFSET1 | [7:0] | NCO_PHASE_OFFSET[15:8] | | NCO phase offset. | 0x0 | R/W |
| 0x124 | ACC_MODULUS0 | [7:0] | ACC_MODULUS[7:0] | | DDS modulus. This is B in the equation $f_{OUT} = f_{DAC} \times (M/N) = f_{DAC} \times ((X + A/B)/2^{48})$. Note this modulus value is used for all NCO FTWs. | 0x0 | R/W |

| Hex. Addr. | Name | Bits | Bit Name | Settings | Description | Reset | Access |
|---|---|---|---|---|---|---|---|
| 0x125 | ACC_MODULUS1 | [7:0] | ACC_MODULUS[15:8] | | DDS modulus. This is B in the equation $f_{OUT} = f_{DAC} \times (M/N) = f_{DAC} \times ((X + A/B)/2^{48})$. Note this modulus value is used for all NCO FTWs. | 0x0 | R/W |
| 0x126 | ACC_MODULUS2 | [7:0] | ACC_MODULUS[23:16] | | DDS modulus. This is B in the equation $f_{OUT} = f_{DAC} \times (M/N) = f_{DAC} \times ((X + A/B)/2^{48})$. Note this modulus value is used for all NCO FTWs. | 0x0 | R/W |
| 0x127 | ACC_MODULUS3 | [7:0] | ACC_MODULUS[31:24] | | DDS modulus. This is B in the equation $f_{OUT} = f_{DAC} \times (M/N) = f_{DAC} \times ((X + A/B)/2^{48})$. Note this modulus value is used for all NCO FTWs. | 0x0 | R/W |
| 0x128 | ACC_MODULUS4 | [7:0] | ACC_MODULUS[39:32] | | DDS modulus. This is B in the equation $f_{OUT} = f_{DAC} \times (M/N) = f_{DAC} \times ((X + A/B)/2^{48})$. Note this modulus value is used for all NCO FTWs. | 0x0 | R/W |
| 0x129 | ACC_MODULUS5 | [7:0] | ACC_MODULUS[47:40] | | DDS modulus. This is B in the equation $f_{OUT} = f_{DAC} \times (M/N) = f_{DAC} \times ((X + A/B)/2^{48})$. Note this modulus value is used for all NCO FTWs. | 0x0 | R/W |
| 0x12A | ACC_DELTA0 | [7:0] | ACC_DELTA[7:0] | | DDS delta. This is A in the equation $f_{OUT} = f_{DAC} \times (M/N) = f_{DAC} \times ((X + A/B)/2^{48})$. Note this modulus value is used for all NCO FTWs. Note this delta value is used for all NCO FTWs. | 0x0 | R/W |
| 0x12B | ACC_DELTA1 | [7:0] | ACC_DELTA[15:8] | | DDS delta. This is A in the equation $f_{OUT} = f_{DAC} \times (M/N) = f_{DAC} \times ((X + A/B)/2^{48})$. Note this modulus value is used for all NCO FTWs. Note this delta value is used for all NCO FTWs. | 0x0 | R/W |
| 0x12C | ACC_DELTA2 | [7:0] | ACC_DELTA[23:16] | | DDS delta. This is A in the equation $f_{OUT} = f_{DAC} \times (M/N) = f_{DAC} \times ((X + A/B)/2^{48})$. Note this modulus value is used for all NCO FTWs. Note this delta value is used for all NCO FTWs. | 0x0 | R/W |
| 0x12D | ACC_DELTA3 | [7:0] | ACC_DELTA[31:24] | | DDS delta. This is A in the equation $f_{OUT} = f_{DAC} \times (M/N) = f_{DAC} \times ((X + A/B)/2^{48})$. Note this delta value is used for all NCO FTWs. | 0x0 | R/W |
| 0x12E | ACC_DELTA4 | [7:0] | ACC_DELTA[39:32] | | DDS delta. This is A in the equation $f_{OUT} = f_{DAC} \times (M/N) = f_{DAC} \times ((X + A/B)/2^{48})$. Note this modulus value is used for all NCO FTWs. Note this delta value is used for all NCO FTWs. | 0x0 | R/W |
| 0x12F | ACC_DELTA5 | [7:0] | ACC_DELTA[47:40] | | DDS delta. This is A in the equation $f_{OUT} = f_{DAC} \times (M/N) = f_{DAC} \times ((X + A/B)/2^{48})$. Note this modulus value is used for all NCO FTWs. Note this delta value is used for all NCO FTWs. | 0x0 | R/W |
| 0x132 | TEMP_SENS_LSB | [7:0] | TEMP_SENS_OUT[7:0] | | Output of the temperature sensor ADC. | 0x0 | R |
| 0x133 | TEMP_SENS_MSB | [7:0] | TEMP_SENS_OUT[15:8] | | Output of the temperature sensor ADC. | 0x0 | R |
| 0x134 | TEMP_SENS_UPDATE | [7:1] | RESERVED | | Reserved. | 0x0 | R |
| | | 0 | TEMP_SENS_UPDATE | | Set to 1 to update the temperature sensor reading with a new value. | 0x0 | R/W |
| 0x135 | TEMP_SENS_CTRL | 7 | TEMP_SENS_FAST | | A 1 sets the temperature sensor digital filter bandwidth wider for faster settling time. | 0x0 | R/W |
| | | [6:1] | RESERVED | | Reserved. | 0x10 | R/W |
| | | 0 | TEMP_SENS_ENABLE | | Set to 1 to enable the temperature sensor. | 0x0 | R/W |

| Hex. Addr. | Name | Bits | Bit Name | Settings | Description | Reset | Access |
|---|---|---|---|---|---|---|---|
| 0x14B | PRBS | 7 | PRBS_GOOD_Q | | Good data indicator imaginary channel. | 0x0 | R |
| | | | | 0 | Incorrect sequence detected. | | |
| | | | | 1 | Correct PRBS sequence detected. | | |
| | | 6 | PRBS_GOOD_I | | Good data indicator real channel. | 0x0 | R |
| | | | | 0 | Incorrect sequence detected. | | |
| | | | | 1 | Correct PRBS sequence detected. | | |
| | | 5 | RESERVED | | Reserved. | 0x0 | R |
| | | 4 | PRBS_INV_Q | | Data inversion imaginary channel. | 0x1 | R/W |
| | | | | 0 | Expect normal data. | | |
| | | | | 1 | Expect inverted data. | | |
| | | 3 | PRBS_INV_I | | Data inversion real channel. | 0x0 | R/W |
| | | | | 0 | Expect normal data. | | |
| | | | | 1 | Expect inverted data. | | |
| | | 2 | PRBS_MODE | | Polynomial select. | 0x0 | R/W |
| | | | | 0 | 7-bit: $x^7 + x^6 + 1$. | | |
| | | | | 1 | 15-bit: $x^{15} + x^{14} + 1$. | | |
| | | 1 | PRBS_RESET | | Reset error counters. | 0x0 | R/W |
| | | | | 0 | Normal operation. | | |
| | | | | 1 | Reset counters. | | |
| | | 0 | PRBS_EN | | Enable PRBS checker. | 0x0 | R/W |
| | | | | 0 | Disable. | | |
| | | | | 1 | Enable. | | |
| 0x14C | PRBS_ERROR_I | [7:0] | PRBS_COUNT_I | | Error count value real channel. | 0x0 | R |
| 0x14D | PRBS_ERROR_Q | [7:0] | PRBS_COUNT_Q | | Error count value imaginary channel. | 0x0 | R |
| 0x151 | DECODE_CTRL | [7:3] | RESERVED | | Reserved. | 0x0 | R/W |
| | | 2 | SHUFFLE | | Shuffle mode. Enables shuffle mode for better spurious performance. | 0x0 | R/W |
| | | | | 0 | Disable MSB shuffling (use thermometer encoding). | | |
| | | | | 1 | Enable MSB shuffling. | | |
| | | [1:0] | RESERVED | | Reserved. | 0x0 | R/W |
| 0x152 | DECODE_MODE | [7:2] | RESERVED | | Reserved. | 0x0 | R |
| | | [1:0] | DECODE_MODE | | Decode mode. | 0x0 | R/W |
| | | | | 00 | Nonreturn-to-zero mode (first Nyquist). | | |
| | | | | 01 | Mix-Mode (second Nyquist). | | |
| | | | | 10 | Return to zero. | | |
| | | | | 11 | Reserved. | | |
| 0x1DF | SPI_STRENGTH | [7:4] | RESERVED | | Reserved. | 0x0 | R |
| | | [3:0] | SPIDRV | | Slew and drive strength for CMOS SPI outputs. Slew = Bits[1:0], drive = Bits[3:2]. | 0xF | R/W |
| 0x200 | MASTER_PD | [7:1] | RESERVED | | Reserved. | 0x0 | R |
| | | 0 | SPI_PD_MASTER | | Power down the entire JESD204B Rx analog (all eight channels and bias). | 0x1 | R/W |
| 0x201 | PHY_PD | [7:0] | SPI_PD_PHY | | SPI override to power down the individual PHYs. | 0x0 | R/W |
| | | | | | Bit 0 controls the SERDIN0± PHY. | | |
| | | | | | Bit 1 controls the SERDIN1± PHY. | | |
| | | | | | Bit 2 controls the SERDIN2± PHY. | | |

| Hex. Addr. | Name | Bits | Bit Name | Settings | Description | Reset | Access |
|---|---|---|---|---|---|---|---|
| | | | | | Bit 3 controls the SERDIN3± PHY. | | |
| | | | | | Bit 4 controls the SERDIN4± PHY. | | |
| | | | | | Bit 5 controls the SERDIN5± PHY. | | |
| | | | | | Bit 6 controls the SERDIN6± PHY. | | |
| | | | | | Bit 7 controls the SERDIN7± PHY. | | |
| 0x203 | GENERIC_PD | [7:2] | RESERVED | | Reserved. | 0x0 | R |
| | | 1 | SPI_SYNC1_PD | | Power down LVDS buffer for the sync request signal, $\overline{\text{SYNCOUT}}$. | 0x0 | R/W |
| | | 0 | RESERVED | | Reserved. | 0x0 | R/W |
| 0x206 | CDR_RESET | [7:1] | RESERVED | | Reserved. | 0x0 | R |
| | | 0 | SPI_CDR_$\overline{\text{RESET}}$ | | Resets the digital control logic for all PHYs. | 0x1 | R/W |
| | | | | 0 | CDR logic is reset. | | |
| | | | | 1 | CDR logic is operational. | | |
| 0x230 | CDR_OPERATING_MODE_REG_0 | [7:6] | RESERVED | | Reserved. | 0x1 | R/W |
| | | 5 | SPI_ENHALFRATE | | Enables half rate CDR operation, must be enabled for data rates above 6 Gbps. | 0x1 | R/W |
| | | | | 0 | Disables CDR half rate operation, data rate ≤ 6 Gbps. | | |
| | | | | 1 | Enables CDR half rate operation, data rate > 6 Gbps. | | |
| | | [4:3] | RESERVED | | Reserved. | 0x0 | R/W |
| | | [2:1] | SPI_DIVISION_RATE | | Enables oversampling of the input data. | 0x0 | R/W |
| | | | | 00 | No division. Data rate > 3 Gbps. | | |
| | | | | 01 | Division by 2. 1.5 Gbps < data rate ≤ 3 Gbps. | | |
| | | | | 10 | Division by 4. 750 Mbps < data rate ≤ 1.5 Gbps. | | |
| | | 0 | RESERVED | | Reserved. | 0x0 | R/W |
| 0x250 | EQ_CONFIG_PHY_0_1 | [7:4] | SPI_EQ_CONFIG1 | | | 0x8 | R/W |
| | | | | 0000 | Manual mode (SPI configured values used). | | |
| | | | | 0001 | Boost level = 1. | | |
| | | | | 0010 | Boost level = 2. | | |
| | | | | 0011 | Boost level = 3. | | |
| | | | | 0100 | Boost level = 4. | | |
| | | | | 0101 | Boost level = 5. | | |
| | | | | 0110 | Boost level = 6. | | |
| | | | | 0111 | Boost level = 7. | | |
| | | | | 1000 | Boost level = 8. | | |
| | | | | 1001 | Boost level = 9. | | |
| | | | | 1010 | Boost level = 10. | | |
| | | | | 1011 | Boost level = 11. | | |
| | | | | 1100 | Boost level = 12. | | |
| | | | | 1101 | Boost level = 13. | | |
| | | | | 1110 | Boost level = 14. | | |
| | | | | 1111 | Boost level = 15. | | |
| | | [3:0] | SPI_EQ_CONFIG0 | | | 0x8 | R/W |
| | | | | 0000 | Manual mode (SPI configured values used). | | |
| | | | | 0001 | Boost level = 1. | | |
| | | | | 0010 | Boost level = 2. | | |
| | | | | 0011 | Boost level = 3. | | |

| Hex. Addr. | Name | Bits | Bit Name | Settings | Description | Reset | Access |
|---|---|---|---|---|---|---|---|
| | | | | 0100 | Boost level = 4. | | |
| | | | | 0101 | Boost level = 5. | | |
| | | | | 0110 | Boost level = 6. | | |
| | | | | 0111 | Boost level = 7. | | |
| | | | | 1000 | Boost level = 8. | | |
| | | | | 1001 | Boost level = 9. | | |
| | | | | 1010 | Boost level = 10. | | |
| | | | | 1011 | Boost level = 11. | | |
| | | | | 1100 | Boost level = 12. | | |
| | | | | 1101 | Boost level = 13. | | |
| | | | | 1110 | Boost level = 14. | | |
| | | | | 1111 | Boost level = 15. | | |
| 0x251 | EQ_CONFIG_PHY_2_3 | [7:4] | SPI_EQ_CONFIG3 | | | 0x8 | R/W |
| | | | | 0000 | Manual mode (SPI configured values used). | | |
| | | | | 0001 | Boost level = 1. | | |
| | | | | 0010 | Boost level = 2. | | |
| | | | | 0011 | Boost level = 3. | | |
| | | | | 0100 | Boost level = 4. | | |
| | | | | 0101 | Boost level = 5. | | |
| | | | | 0110 | Boost level = 6. | | |
| | | | | 0111 | Boost level = 7. | | |
| | | | | 1000 | Boost level = 8. | | |
| | | | | 1001 | Boost level = 9. | | |
| | | | | 1010 | Boost level = 10. | | |
| | | | | 1011 | Boost level = 11. | | |
| | | | | 1100 | Boost level = 12. | | |
| | | | | 1101 | Boost level = 13. | | |
| | | | | 1110 | Boost level = 14. | | |
| | | | | 1111 | Boost level = 15. | | |
| | | [3:0] | SPI_EQ_CONFIG2 | | | 0x8 | R/W |
| | | | | 0000 | Manual mode (SPI configured values used). | | |
| | | | | 0001 | Boost level = 1. | | |
| | | | | 0010 | Boost level = 2. | | |
| | | | | 0011 | Boost level = 3. | | |
| | | | | 0100 | Boost level = 4. | | |
| | | | | 0101 | Boost level = 5. | | |
| | | | | 0110 | Boost level = 6. | | |
| | | | | 0111 | Boost level = 7. | | |
| | | | | 1000 | Boost level = 8. | | |
| | | | | 1001 | Boost level = 9. | | |
| | | | | 1010 | Boost level = 10. | | |
| | | | | 1011 | Boost level = 11. | | |
| | | | | 1100 | Boost level = 12. | | |
| | | | | 1101 | Boost Level = 13. | | |
| | | | | 1110 | Boost level = 14. | | |
| | | | | 1111 | Boost level = 15. | | |
| 0x252 | EQ_CONFIG_PHY_4_5 | [7:4] | SPI_EQ_CONFIG5 | | | 0x8 | R/W |
| | | | | 0000 | Manual mode (SPI configured values used). | | |
| | | | | 0001 | Boost level = 1. | | |
| | | | | 0010 | Boost level = 2. | | |
| | | | | 0011 | Boost level = 3. | | |

| Hex. Addr. | Name | Bits | Bit Name | Settings | Description | Reset | Access |
|---|---|---|---|---|---|---|---|
| | | | | 0100 | Boost level = 4. | | |
| | | | | 0101 | Boost level = 5. | | |
| | | | | 0110 | Boost level = 6. | | |
| | | | | 0111 | Boost level = 7. | | |
| | | | | 1000 | Boost level = 8. | | |
| | | | | 1001 | Boost level = 9. | | |
| | | | | 1010 | Boost level = 10. | | |
| | | | | 1011 | Boost level = 11. | | |
| | | | | 1100 | Boost level = 12. | | |
| | | | | 1101 | Boost level = 13. | | |
| | | | | 1110 | Boost level = 14. | | |
| | | | | 1111 | Boost level = 15. | | |
| | | [3:0] | SPI_EQ_CONFIG4 | | | 0x8 | R/W |
| | | | | 0000 | Manual mode (SPI configured values used). | | |
| | | | | 0001 | Boost level = 1. | | |
| | | | | 0010 | Boost level = 2. | | |
| | | | | 0011 | Boost level = 3. | | |
| | | | | 0100 | Boost level = 4. | | |
| | | | | 0101 | Boost level = 5. | | |
| | | | | 0110 | Boost level = 6. | | |
| | | | | 0111 | Boost level = 7. | | |
| | | | | 1000 | Boost level = 8. | | |
| | | | | 1001 | Boost level = 9. | | |
| | | | | 1010 | Boost level = 10. | | |
| | | | | 1011 | Boost level = 11. | | |
| | | | | 1100 | Boost level = 12. | | |
| | | | | 1101 | Boost level = 13. | | |
| | | | | 1110 | Boost level = 14. | | |
| | | | | 1111 | Boost level = 15. | | |
| 0x253 | EQ_CONFIG_PHY_6_7 | [7:4] | SPI_EQ_CONFIG7 | | | 0x8 | R/W |
| | | | | 0000 | Manual mode (SPI configured values used). | | |
| | | | | 0001 | Boost level = 1. | | |
| | | | | 0010 | Boost level = 2. | | |
| | | | | 0011 | Boost level = 3. | | |
| | | | | 0100 | Boost level = 4. | | |
| | | | | 0101 | Boost level = 5. | | |
| | | | | 0110 | Boost level = 6. | | |
| | | | | 0111 | Boost level = 7. | | |
| | | | | 1000 | Boost level = 8. | | |
| | | | | 1001 | Boost level = 9. | | |
| | | | | 1010 | Boost level = 10. | | |
| | | | | 1011 | Boost level = 11. | | |
| | | | | 1100 | Boost level = 12. | | |
| | | | | 1101 | Boost level = 13. | | |
| | | | | 1110 | Boost level = 14. | | |
| | | | | 1111 | Boost level = 15. | | |
| | | [3:0] | SPI_EQ_CONFIG6 | | | 0x8 | R/W |
| | | | | 0000 | Manual mode (SPI configured values used). | | |
| | | | | 0001 | Boost level = 1. | | |
| | | | | 0010 | Boost level = 2. | | |
| | | | | 0011 | Boost level = 3. | | |

| Hex. Addr. | Name | Bits | Bit Name | Settings | Description | Reset | Access |
|---|---|---|---|---|---|---|---|
| | | | | 0100 | Boost level = 4. | | |
| | | | | 0101 | Boost level = 5. | | |
| | | | | 0110 | Boost level = 6. | | |
| | | | | 0111 | Boost level = 7. | | |
| | | | | 1000 | Boost level = 8. | | |
| | | | | 1001 | Boost level = 9. | | |
| | | | | 1010 | Boost level = 10. | | |
| | | | | 1011 | Boost level = 11. | | |
| | | | | 1100 | Boost level = 12. | | |
| | | | | 1101 | Boost level = 13. | | |
| | | | | 1110 | Boost level = 14. | | |
| | | | | 1111 | Boost level = 15. | | |
| 0x268 | EQ_BIAS_REG | [7:6] | EQ_POWER_MODE | | Controls the equalizer power mode/insertion loss capability. | 0x1 | R/W |
| | | | | 00 | Normal mode. | | |
| | | | | 01 | Low power mode. | | |
| | | [5:0] | RESERVED | | Reserved. | 0x4 | R/W |
| 0x280 | SYNTH_ENABLE_CNTRL | [7:3] | RESERVED | | Reserved. | 0x0 | R |
| | | 2 | SPI_RECAL_SYNTH | | Set this bit high to rerun all of the SERDES PLL calibration routines. Set this bit low again to allow additional recalibrations. Rising edge causes the calibration. | 0x0 | R/W |
| | | 1 | RESERVED | | Reserved. | 0x0 | R/W |
| | | 0 | SPI_ENABLE_SYNTH | | Enable the SERDES PLL. Setting this bit turns on all currents and proceeds to calibrate the PLL. Make sure reference clock and division ratios are correct before enabling this bit. | 0x0 | R/W |
| 0x281 | PLL_STATUS | [7:6] | RESERVED | | Reserved. | 0x0 | R |
| | | 5 | SPI_CP_OVER_RANGE_HIGH_RB | | If set, the SERDES PLL CP output is above valid operating range. | 0x0 | R |
| | | | | 0 | Charge pump output is within operating range. | | |
| | | | | 1 | Charge pump output is above operating range. | | |
| | | 4 | SPI_CP_OVER_RANGE_LOW_RB | | If set, the SERDES PLL CP output is below valid operating range. | 0x0 | R |
| | | | | 0 | Charge pump output is within operating range. | | |
| | | | | 1 | Charge pump output is below operating range. | | |
| | | 3 | SPI_CP_CAL_VALID_RB | | This bit tells the user if the charge pump calibration has completed and is valid. | 0x0 | R |
| | | | | 0 | Charge pump calibration is not valid. | | |
| | | | | 1 | Charge pump calibration is valid. | | |
| | | [2:1] | RESERVED | | Reserved. | 0x0 | R |
| | | 0 | SPI_PLL_LOCK_RB | | If set, the SERDES synthesizer locked. | 0x0 | R |
| | | | | 0 | PLL is not locked. | | |
| | | | | 1 | PLL is locked. | | |

| Hex. Addr. | Name | Bits | Bit Name | Settings | Description | Reset | Access |
|---|---|---|---|---|---|---|---|
| 0x289 | REF_CLK_DIVIDER_LDO | [7:2] | RESERVED | | Reserved. | 0x0 | R |
| | | [1:0] | SERDES_PLL_DIV_FACTOR | | SERDES PLL reference clock division factor. This field controls the division of the SERDES PLL reference clock before it is fed into the SERDES PLL PFD. It must be set so that $f_{REF}$/DivFactor is between 35 MHz and 80 MHz. | 0x0 | R/W |
| | | | | 00 | Divide by 4 for lane rate between 6 Gbps and 12.5 Gbps. | | |
| | | | | 01 | Divide by 2 for lane rate between 3 Gbps and 6 Gbps. | | |
| | | | | 10 | Divide by 1 for lane rate between 1.5 Gbps and 3 Gbps. | | |
| 0x2A7 | TERM_BLK1_CTRLREG0 | [7:1] | RESERVED | | Reserved. | 0x0 | R |
| | | 0 | SPI_I_TUNE_R_CAL_TERMBLK1 | | Rising edge of this bit starts a termination calibration routine. | 0x0 | R/W |
| 0x2A8 | TERM_BLK1_CTRLREG1 | [7:0] | SPI_I_SERIALIZER_RTRIM_TERMBLK1 | | SPI override for termination value for PHY 0, PHY 1, PHY 6, and PHY 7. Value options are as follows: | 0x0 | R/W |
| | | | | XXX0XXXX | Automatically calibrate termination value. | | |
| | | | | XXX1000X | Force 000 as termination value. | | |
| | | | | XXX1001X | Force 001 as termination value. | | |
| | | | | XXX1010X | Force 010 as termination value. | | |
| | | | | XXX1011X | Force 011 as termination value. | | |
| | | | | XXX1100X | Force 100 as termination value. | | |
| | | | | XXX1101X | Force 101 as termination value. | | |
| | | | | XXX1110X | Force 110 as termination value. | | |
| | | | | XXX1111X | Force 111 as termination value. | | |
| | | | | XXX1000X | Force 000 as termination value. | | |
| 0x2AC | TERM_BLK1_RD_REG0 | [7:4] | RESERVED | | Reserved. | 0x0 | R |
| | | [3:0] | SPI_O_RCAL_CODE_TERMBLK1 | | Readback of calibration code for PHY 0, PHY 1, PHY 6, and PHY 7. | 0x0 | R |
| 0x2AE | TERM_BLK2_CTRLREG0 | [7:1] | RESERVED | | Reserved. | 0x0 | R |
| | | 0 | SPI_I_TUNE_R_CAL_TERMBLK2 | | Rising edge of this bit starts a termination calibration routine. | 0x0 | R/W |
| 0x2AF | TERM_BLK2_CTRLREG1 | [7:0] | SPI_I_SERIALIZER_RTRIM_TERMBLK2 | | SPI override for termination value for PHY 2, PHY 3, PHY 4, and PHY 5. Value options are as follows: | 0x0 | R/W |
| | | | | XXX0XXXX | Automatically calibrate termination value. | | |
| | | | | XXX1000X | Force 000 as termination value. | | |
| | | | | XXX1001X | Force 001 as termination value. | | |
| | | | | XXX1010X | Force 010 as termination value. | | |
| | | | | XXX1011X | Force 011 as termination value. | | |
| | | | | XXX1100X | Force 100 as termination value. | | |
| | | | | XXX1101X | Force 101 as termination value. | | |
| | | | | XXX1110X | Force 110 as termination value. | | |
| | | | | XXX1111X | Force 111 as termination value. | | |
| | | | | XXX1000X | Force 000 as termination value. | | |
| 0x2B3 | TERM_BLK2_RD_REG0 | [7:4] | RESERVED | | Reserved. | 0x0 | R |
| | | [3:0] | SPI_O_RCAL_CODE_TERMBLK2 | | Readback of calibration code for PHY 2, PHY 3, PHY 4, and PHY 5. | 0x0 | R |

| Hex. Addr. | Name | Bits | Bit Name | Settings | Description | Reset | Access |
|---|---|---|---|---|---|---|---|
| 0x2BB | TERM_OFFSET_0 | [7:4] | RESERVED | | Reserved. | 0x0 | R |
| | | [3:0] | TERM_OFFSET_0 | | Add or subtract from the termination calibration value of Physical Lane 0. 4-bit signed magnitude value that adds to or subtracts from the termination value. Bit 3 is the sign bit, and Bits[2:0] are the magnitude bits. | 0x0 | R/W |
| 0x2BC | TERM_OFFSET_1 | [7:4] | RESERVED | | Reserved. | 0x0 | R |
| | | [3:0] | TERM_OFFSET_1 | | Add or subtract from the termination calibration value of Physical Lane 1. 4-bit signed magnitude value that adds to or subtracts from the termination value. Bit 3 is the sign bit, and Bits[2:0] are the magnitude bits. | 0x0 | R/W |
| 0x2BD | TERM_OFFSET_2 | [7:4] | RESERVED | | Reserved. | 0x0 | R |
| | | [3:0] | TERM_OFFSET_2 | | Add or subtract from the termination calibration value of Physical Lane 2. 4-bit signed magnitude value that adds to or subtracts from the termination value. Bit 3 is the sign bit, and Bits[2:0] are the magnitude bits. | 0x0 | R/W |
| 0x2BE | TERM_OFFSET_3 | [7:4] | RESERVED | | Reserved. | 0x0 | R |
| | | [3:0] | TERM_OFFSET_3 | | Add or subtract from the termination calibration value of Physical Lane 3. 4-bit signed magnitude value that adds to or subtracts from the termination value. Bit 3 is the sign bit, and Bits[2:0] are the magnitude bits. | 0x0 | R/W |
| 0x2BF | TERM_OFFSET_4 | [7:4] | RESERVED | | Reserved. | 0x0 | R |
| | | [3:0] | TERM_OFFSET_4 | | Add or subtract from the termination calibration value of Physical Lane 4. 4-bit signed magnitude value that adds to or subtracts from the termination value. Bit 3 is the sign bit, and Bits[2:0] are the magnitude bits. | 0x0 | R/W |
| 0x2C0 | TERM_OFFSET_5 | [7:4] | RESERVED | | Reserved. | 0x0 | R |
| | | [3:0] | TERM_OFFSET_5 | | Add or subtract from the termination calibration value of Physical Lane 5. 4-bit signed magnitude value that adds to or subtracts from the termination value. Bit 3 is the sign bit, and Bits[2:0] are the magnitude bits. | 0x0 | R/W |
| 0x2C1 | TERM_OFFSET_6 | [7:4] | RESERVED | | Reserved. | 0x0 | R |
| | | [3:0] | TERM_OFFSET_6 | | Add or subtract from the termination calibration value of Physical Lane 6. 4-bit signed magnitude value that adds to or subtracts from the termination value. Bit 3 is the sign bit, and Bits[2:0] are the magnitude bits. | 0x0 | R/W |
| 0x2C2 | TERM_OFFSET_7 | [7:4] | RESERVED | | Reserved. | 0x0 | R |
| | | [3:0] | TERM_OFFSET_7 | | Add or subtract from the termination calibration value of Physical Lane 7. 4-bit signed magnitude value that adds to or subtracts from the termination value. Bit 3 is the sign bit, and Bits[2:0] are the magnitude bits. | 0x0 | R/W |

| Hex. Addr. | Name | Bits | Bit Name | Settings | Description | Reset | Access |
|---|---|---|---|---|---|---|---|
| 0x300 | GENERAL_JRX_CTRL_0 | 7 | RESERVED | | Reserved. | 0x0 | R |
| | | 6 | CHECKSUM_MODE | | JESD204B link parameter checksum calculation method. | 0x0 | R/W |
| | | | | 0 | Checksum is sum of fields. | | |
| | | | | 1 | Checksum is sum of octets. | | |
| | | [5:1] | RESERVED | | Reserved. | 0x0 | R |
| | | 0 | LINK_EN | | This bit brings up the JESD204B receiver when all link parameters are programmed and all clocks are ready. | 0x0 | R/W |
| 0x302 | DYN_LINK_LATENCY_0 | [7:5] | RESERVED | | Reserved. | 0x0 | R |
| | | [4:0] | DYN_LINK_LATENCY_0 | | Measurement of the JESD204B link delay (in PCLK units). Link 0 dynamic link latency. Latency between current deframer LMFC and the global LMFC. | 0x0 | R |
| 0x304 | LMFC_DELAY_0 | [7:5] | RESERVED | | Reserved. | 0x0 | R |
| | | [4:0] | LMFC_DELAY_0 | | Fixed part of the JESD204B link delay (in PCLK units). Delay in frame clock cycles for global LMFC for Link 0. | 0x0 | R/W |
| 0x306 | LMFC_VAR_0 | [7:5] | RESERVED | | Reserved. | 0x0 | R |
| | | [4:0] | LMFC_VAR_0 | | Variable part of the JESD204B link delay (in PCLK units). Location in Rx LMFC where JESD204B words are read out from buffer. This setting must not be more than 10 PCLKs. | 0x1F | R/W |
| 0x308 | XBAR_LN_0_1 | [7:6] | RESERVED | | Reserved. | 0x0 | R |
| | | [5:3] | SRC_LANE1 | | Select data from SERDIN0±, SERDIN1±, …, or SERDIN7± for Logic Lane 1. | 0x1 | R/W |
| | | | | 000 | Data is from SERDIN0±. | | |
| | | | | 001 | Data is from SERDIN1±. | | |
| | | | | 010 | Data is from SERDIN2±. | | |
| | | | | 011 | Data is from SERDIN3±. | | |
| | | | | 100 | Data is from SERDIN4±. | | |
| | | | | 101 | Data is from SERDIN5±. | | |
| | | | | 110 | Data is from SERDIN6±. | | |
| | | | | 111 | Data is from SERDIN7±. | | |
| | | [2:0] | SRC_LANE0 | | Select data from SERDIN0±, SERDIN1±, …, or SERDIN7± for Logic Lane 0. | 0x0 | R/W |
| | | | | 000 | Data is from SERDIN0±. | | |
| | | | | 001 | Data is from SERDIN1±. | | |
| | | | | 010 | Data is from SERDIN2±. | | |
| | | | | 011 | Data is from SERDIN3±. | | |
| | | | | 100 | Data is from SERDIN4±. | | |
| | | | | 101 | Data is from SERDIN5±. | | |
| | | | | 110 | Data is from SERDIN6±. | | |
| | | | | 111 | Data is from SERDIN7±. | | |
| 0x309 | XBAR_LN_2_3 | [7:6] | RESERVED | | Reserved. | 0x0 | R |
| | | [5:3] | SRC_LANE3 | | Select data from SERDIN0±, SERDIN1±, …, or SERDIN7± for Logic Lane 3. | 0x3 | R/W |
| | | | | 000 | Data is from SERDIN0±. | | |
| | | | | 001 | Data is from SERDIN1±. | | |
| | | | | 010 | Data is from SERDIN2±. | | |
| | | | | 011 | Data is from SERDIN3±. | | |

| Hex. Addr. | Name | Bits | Bit Name | Settings | Description | Reset | Access |
|---|---|---|---|---|---|---|---|
| | | | | 100 | Data is from SERDIN4±. | | |
| | | | | 101 | Data is from SERDIN5±. | | |
| | | | | 110 | Data is from SERDIN6±. | | |
| | | | | 111 | Data is from SERDIN7±. | | |
| | | [2:0] | SRC_LANE2 | | Select data from SERDIN0±, SERDIN1±, …, or SERDIN7± for Logic Lane 2. | 0x2 | R/W |
| | | | | 000 | Data is from SERDIN0±. | | |
| | | | | 001 | Data is from SERDIN1±. | | |
| | | | | 010 | Data is from SERDIN2±. | | |
| | | | | 011 | Data is from SERDIN3±. | | |
| | | | | 100 | Data is from SERDIN4±. | | |
| | | | | 101 | Data is from SERDIN5±. | | |
| | | | | 110 | Data is from SERDIN6±. | | |
| | | | | 111 | Data is from SERDIN7±. | | |
| 0x30A | XBAR_LN_4_5 | [7:6] | RESERVED | | Reserved. | 0x0 | R |
| | | [5:3] | SRC_LANE5 | | Select data from SERDIN0±, SERDIN1±, …, or SERDIN7± for Logic Lane 5. | 0x5 | R/W |
| | | | | 000 | Data is from SERDIN0±. | | |
| | | | | 001 | Data is from SERDIN1±. | | |
| | | | | 010 | Data is from SERDIN2±. | | |
| | | | | 011 | Data is from SERDIN3±. | | |
| | | | | 100 | Data is from SERDIN4±. | | |
| | | | | 101 | Data is from SERDIN5±. | | |
| | | | | 110 | Data is from SERDIN6±. | | |
| | | | | 111 | Data is from SERDIN7±. | | |
| | | [2:0] | SRC_LANE4 | | Select data from SERDIN0±, SERDIN1±, …, or SERDIN7± for Logic Lane 4. | 0x4 | R/W |
| | | | | 000 | Data is from SERDIN0±. | | |
| | | | | 001 | Data is from SERDIN1±. | | |
| | | | | 010 | Data is from SERDIN2±. | | |
| | | | | 011 | Data is from SERDIN3±. | | |
| | | | | 100 | Data is from SERDIN4±. | | |
| | | | | 101 | Data is from SERDIN5±. | | |
| | | | | 110 | Data is from SERDIN6±. | | |
| | | | | 111 | Data is from SERDIN7±. | | |
| 0x30B | XBAR_LN_6_7 | [7:6] | RESERVED | | Reserved. | 0x0 | R |
| | | [5:3] | SRC_LANE7 | | Select data from SERDIN0±, SERDIN1±, …, or SERDIN7± for Logic Lane 7. | 0x7 | R/W |
| | | | | 000 | Data is from SERDIN0±. | | |
| | | | | 001 | Data is from SERDIN1±. | | |
| | | | | 010 | Data is from SERDIN2±. | | |
| | | | | 011 | Data is from SERDIN3±. | | |
| | | | | 100 | Data is from SERDIN4±. | | |
| | | | | 101 | Data is from SERDIN5±. | | |
| | | | | 110 | Data is from SERDIN6±. | | |
| | | | | 111 | Data is from SERDIN7±. | | |
| | | [2:0] | SRC_LANE6 | | Select data from SERDIN0±, SERDIN1±, …, or SERDIN7± for Logic Lane 6. | 0x6 | R/W |
| | | | | 000 | Data is from SERDIN0±. | | |
| | | | | 001 | Data is from SERDIN1±. | | |

| Hex. Addr. | Name | Bits | Bit Name | Settings | Description | Reset | Access |
|---|---|---|---|---|---|---|---|
| | | | | 010 | Data is from SERDIN2±. | | |
| | | | | 011 | Data is from SERDIN3±. | | |
| | | | | 100 | Data is from SERDIN4±. | | |
| | | | | 101 | Data is from SERDIN5±. | | |
| | | | | 110 | Data is from SERDIN6±. | | |
| | | | | 111 | Data is from SERDIN7±. | | |
| 0x30C | FIFO_STATUS_REG_0 | [7:0] | LANE_FIFO_FULL | | Bit 0 corresponds to FIFO full flag for data from SERDIN0±.<br>Bit 1 corresponds to FIFO full flag for data from SERDIN1±.<br>Bit 2 corresponds to FIFO full flag for data from SERDIN2±.<br>Bit 3 corresponds to FIFO full flag for data from SERDIN3±.<br>Bit 4 corresponds to FIFO full flag for data from SERDIN4±.<br>Bit 5 corresponds to FIFO full flag for data from SERDIN5±.<br>Bit 6 corresponds to FIFO full flag for data from SERDIN6±.<br>Bit 7 corresponds to FIFO full flag for data from SERDIN7±. | 0x0 | R |
| 0x30D | FIFO_STATUS_REG_1 | [7:0] | LANE_FIFO_EMPTY | | Bit 0 corresponds to FIFO empty flag for data from SERDIN0±.<br>Bit 1 corresponds to FIFO empty flag for data from SERDIN1±.<br>Bit 2 corresponds to FIFO empty flag for data from SERDIN2±.<br>Bit 3 corresponds to FIFO empty flag for data from SERDIN3±.<br>Bit 4 corresponds to FIFO empty flag for data from SERDIN4±.<br>Bit 5 corresponds to FIFO empty flag for data from SERDIN5±.<br>Bit 6 corresponds to FIFO empty flag for data from SERDIN6±.<br>Bit 7 corresponds to FIFO empty flag for data from SERDIN7±. | 0x0 | R |
| 0x311 | $\overline{\text{SYNC}}$_GEN_0 | [7:3] | RESERVED | | Reserved. | 0x0 | R |
| | | 2 | EOMF_MASK_0 | | Mask $\overline{\text{EOM}}$F from QBD_0. Assert $\overline{\text{SYNCOUT}}$ based on loss of multiframe sync. | 0x0 | R/W |
| | | | | 0 | Do not assert $\overline{\text{SYNCOUT}}$ on loss of multiframe. | | |
| | | | | 1 | Assert $\overline{\text{SYNCOUT}}$ on loss of multiframe. | | |
| | | 1 | RESERVED | | Reserved. | 0x0 | R/W |
| | | 0 | EOF_MASK_0 | | Mask $\overline{\text{EOF}}$ from QBD_0. Assert $\overline{\text{SYNCOUT}}$ based on loss of frame sync. | 0x0 | R/W |
| | | | | 0 | Do not assert $\overline{\text{SYNCOUT}}$ on loss of frame. | | |
| | | | | 1 | Assert $\overline{\text{SYNCOUT}}$ on loss of frame. | | |

| Hex. Addr. | Name | Bits | Bit Name | Settings | Description | Reset | Access |
|---|---|---|---|---|---|---|---|
| 0x312 | $\overline{\text{SYNC}}$_GEN_1 | [7:4] | $\overline{\text{SYNC}}$_ERR_DUR | | Duration of $\overline{\text{SYNCOUT}}$ signal low for purpose of sync error report. 0 means half PCLK cycle. Add an additional PCLK = 4 octets for each increment of the value. | 0x0 | R/W |
| | | [3:0] | $\overline{\text{SYNC}}$_SYNCREQ_DUR | | Duration of $\overline{\text{SYNCOUT}}$ signal low for purpose of sync request. 0 means 5 frame + 9 octets. Add an additional PCLK = 4 octets for each increment of the value. | 0x0 | R/W |
| 0x313 | $\overline{\text{SYNC}}$_GEN_3 | [7:0] | LMFC_PERIOD | | LMFC period in PCLK cycle. This is to report the global LMFC period based on PCLK. | 0x0 | R |
| 0x315 | PHY_PRBS_TEST_EN | [7:0] | PHY_TEST_EN | | Enable PHY BER by ungating the clocks. | 0x0 | R/W |
| | | | | 1 | PHY test enable. | | |
| | | | | 0 | PHY test disable. | | |
| 0x316 | PHY_PRBS_TEST_CTRL | 7 | RESERVED | | Reserved. | 0x0 | R |
| | | [6:4] | PHY_SRC_ERR_CNT | | | 0x0 | R/W |
| | | | | 000 | Report Lane 0 error count. | | |
| | | | | 001 | Report Lane 1 error count. | | |
| | | | | 010 | Report Lane 2 error count. | | |
| | | | | 011 | Report Lane 3 error count. | | |
| | | | | 100 | Report Lane 4 error count. | | |
| | | | | 101 | Report Lane 5 error count. | | |
| | | | | 110 | Report Lane 6 error count. | | |
| | | | | 111 | Report Lane 7 error count. | | |
| | | [3:2] | PHY_PRBS_PAT_SEL | | Select PRBS pattern for PHY BER test. | 0x0 | R/W |
| | | | | 00 | PRBS7. | | |
| | | | | 01 | PRBS15. | | |
| | | | | 10 | PRBS31. | | |
| | | | | 11 | Not used. | | |
| | | 1 | PHY_TEST_START | | Start and stop the PHY PRBS test. | 0x0 | R/W |
| | | | | 0 | Test not started. | | |
| | | | | 1 | Test started. | | |
| | | 0 | PHY_TEST_RESET | | Reset PHY PRBS test state machine and error counters. | 0x0 | R/W |
| | | | | 0 | Not reset. | | |
| | | | | 1 | Reset. | | |
| 0x317 | PHY_PRBS_TEST_THRESHOLD_LOBITS | [7:0] | PHY_PRBS_THRESHOLD_LOBITS | | Bits[7:0] of the 24-bit threshold value set the error flag for PHY PRBS test. | 0x0 | R/W |
| 0x318 | PHY_PRBS_TEST_THRESHOLD_MIDBITS | [7:0] | PHY_PRBS_THRESHOLD_MIDBITS | | Bits[15:8] of the 24-bit threshold value set the error flag for PHY PRBS test. | 0x0 | R/W |
| 0x319 | PHY_PRBS_TEST_THRESHOLD_HIBITS | [7:0] | PHY_PRBS_THRESHOLD_HIBITS | | Bits[23:16] of the 24-bit threshold value set the error flag for PHY PRBS test. | 0x0 | R/W |
| 0x31A | PHY_PRBS_TEST_ERRCNT_LOBITS | [7:0] | PHY_PRBS_ERR_CNT_LOBITS | | Bits[7:0] of the 24-bit reported PHY BER test error count from selected lane. | 0x0 | R |
| 0x31B | PHY_PRBS_TEST_ERRCNT_MIDBITS | [7:0] | PHY_PRBS_ERR_CNT_MIDBITS | | Bits[15:8] of the 24-bit reported PHY BER test error count from selected lane. | 0x0 | R |
| 0x31C | PHY_PRBS_TEST_ERRCNT_HIBITS | [7:0] | PHY_PRBS_ERR_CNT_HIBITS | | Bits[23:16] of the 24-bit reported PHY BER test error count from selected lane. | 0x0 | R |

| Hex. Addr. | Name | Bits | Bit Name | Settings | Description | Reset | Access |
|---|---|---|---|---|---|---|---|
| 0x31D | PHY_PRBS_TEST_STATUS | [7:0] | PHY_PRBS_PASS | | Each bit is for the corresponding lane. Report PHY BER test pass/fail for each lane. | 0xFF | R |
| 0x31E | PHY_DATA_SNAPSHOT_CTRL | [7:5] | RESERVED | | Reserved. | 0x0 | R |
| | | [4:2] | PHY_GRAB_LANE_SEL | | Select which lane to grab data. | 0x0 | R/W |
| | | | | 000 | Grab data from Lane 0. | | |
| | | | | 001 | Grab data from Lane 1. | | |
| | | | | 010 | Grab data from Lane 2. | | |
| | | | | 011 | Grab data from Lane 3. | | |
| | | | | 100 | Grab data from Lane 4. | | |
| | | | | 101 | Grab data from Lane 5. | | |
| | | | | 110 | Grab data from Lane 6. | | |
| | | | | 111 | Grab data from Lane 7. | | |
| | | 1 | PHY_GRAB_MODE | | Use error trigger to grab data. | 0x0 | R/W |
| | | | | 0 | Grab data when PHY_GRAB_DATA is set. | | |
| | | | | 1 | Grab data upon bit error. | | |
| | | 0 | PHY_GRAB_DATA | | Transition from 0 to 1 causes logic to store current receive data from one lane. | 0x0 | R/W |
| 0x31F | PHY_SNAPSHOT_DATA_BYTE0 | [7:0] | PHY_SNAPSHOT_DATA_BYTE0 | | Current data received represents PHY_SNAPSHOT_DATA[7:0]. | 0x0 | R |
| 0x320 | PHY_SNAPSHOT_DATA_BYTE1 | [7:0] | PHY_SNAPSHOT_DATA_BYTE1 | | Current data received represents PHY_SNAPSHOT_DATA[15:8]. | 0x0 | R |
| 0x321 | PHY_SNAPSHOT_DATA_BYTE2 | [7:0] | PHY_SNAPSHOT_DATA_BYTE2 | | Current data received represents PHY_SNAPSHOT_DATA[23:16]. | 0x0 | R |
| 0x322 | PHY_SNAPSHOT_DATA_BYTE3 | [7:0] | PHY_SNAPSHOT_DATA_BYTE3 | | Current data received represents PHY_SNAPSHOT_DATA[31:24]. | 0x0 | R |
| 0x323 | PHY_SNAPSHOT_DATA_BYTE4 | [7:0] | PHY_SNAPSHOT_DATA_BYTE4 | | Current data received represents PHY_SNAPSHOT_DATA[39:32]. | 0x0 | R |
| 0x32C | SHORT_TPL_TEST_0 | [7:4] | SHORT_TPL_SP_SEL | | Short transport layer sample selection. Select which sample to check from a specific DAC. | 0x0 | R/W |
| | | | | 0000 | Sample 0. | | |
| | | | | 0001 | Sample 1. | | |
| | | | | 0010 | Sample 2. | | |
| | | | | 0011 | Sample 3. | | |
| | | | | 0100 | Sample 4. | | |
| | | | | 0101 | Sample 5. | | |
| | | | | 0110 | Sample 6. | | |
| | | | | 0111 | Sample 7. | | |
| | | | | 1000 | Sample 8. | | |
| | | | | 1001 | Sample 9. | | |
| | | | | 1010 | Sample 10. | | |
| | | | | 1011 | Sample 11. | | |
| | | | | 1100 | Sample 12. | | |
| | | | | 1101 | Sample 13. | | |
| | | | | 1110 | Sample 14. | | |
| | | | | 1111 | Sample 15. | | |
| | | [3:2] | SHORT_TPL_M_SEL | | Short transport layer test DAC selection. Select which DAC to check. | 0x0 | R/W |
| | | | | 00 | DAC 0. | | |
| | | | | 01 | DAC 1. | | |
| | | | | 10 | DAC 2. | | |
| | | | | 11 | DAC 3. | | |

| Hex. Addr. | Name | Bits | Bit Name | Settings | Description | Reset | Access |
|---|---|---|---|---|---|---|---|
| | | 1 | SHORT_TPL_TEST_RESET | | Short transport layer test reset. Resets the result of short transport layer test. | 0x0 | R/W |
| | | | | 0 | Not reset. | | |
| | | | | 1 | Reset. | | |
| | | 0 | SHORT_TPL_TEST_EN | | Short transport layer test enable. Enable short transport layer test. | 0x0 | R/W |
| | | | | 0 | Disable. | | |
| | | | | 1 | Enable. | | |
| 0x32D | SHORT_TPL_TEST_1 | [7:0] | SHORT_TPL_REF_SP_LSB | | Short transport layer reference sample LSB. This is the lower eight bits of expected DAC sample. It is used to compare with the received DAC sample at the output of JESD204B Rx. | 0x0 | R/W |
| 0x32E | SHORT_TPL_TEST_2 | [7:0] | SHORT_TPL_REF_SP_MSB | | Short transport layer test reference sample MSB. This is the upper eight bits of expected DAC sample. It is used to compare with the received sample at JESD204B Rx output. | 0x0 | R/W |
| 0x32F | SHORT_TPL_TEST_3 | [7:1] | RESERVED | | Reserved. | 0x0 | R |
| | | 0 | SHORT_TPL_FAIL | | Short transport layer test fail. This bit shows if the selected DAC sample matches the reference sample. If they match, the test passes; otherwise, the test fails. | 0x0 | R |
| | | | | 0 | Test pass. | | |
| | | | | 1 | Test fail. | | |
| 0x334 | JESD_BIT_INVERSE_CTRL | [7:0] | JESD_BIT_INVERSE | | Each bit of this byte inverses the JESD204B deserialized data from one specific JESD204B Rx PHY. The bit order matches the logical lane order. For example, Bit 0 controls Lane 0, Bit 1 controls Lane 1. | 0x0 | R/W |
| 0x400 | DID_REG | [7:0] | DID_RD | | Received ILAS configuration on Lane 0. DID is the device ID number. Link information received on Lane 0 as specified in Section 8.3 of JESD204B. | 0x0 | R |
| 0x401 | BID_REG | [7:0] | BID_RD | | Received ILAS configuration on Lane 0. BID is the bank ID, extension to DID. Link information received on Lane 0 as specified in Section 8.3 of JESD204B. | 0x0 | R |
| 0x402 | LID0_REG | 7 | RESERVED | | Reserved. | 0x0 | R |
| | | 6 | ADJDIR_RD | | Received ILAS configuration on Lane 0. ADJDIR is the direction to adjust the DAC LMFC. Link information received on Lane 0 as specified in Section 8.3 of JESD204B. | 0x0 | R |
| | | 5 | PHADJ_RD | | Received ILAS configuration on Lane 0. PHADJ is the phase adjustment request to DAC. Link information received on Lane 0 as specified in Section 8.3 of JESD204B. | 0x0 | R |
| | | [4:0] | LL_LID0 | | Received ILAS LID configuration on Lane 0. LID0 is the lane identification for Lane 0. Link information received on Lane 0 as specified in Section 8.3 of JESD204B. | 0x0 | R |

| Hex. Addr. | Name | Bits | Bit Name | Settings | Description | Reset | Access |
|---|---|---|---|---|---|---|---|
| 0x403 | SCR_L_REG | 7 | SCR_RD | | Received ILAS configuration on Lane 0. SCR is the Tx scrambling status. Link information received on Lane 0 as specified in Section 8.3 of JESD204B. | 0x0 | R |
| | | | | 0 | Scrambling is disabled. | | |
| | | | | 1 | Scrambling is enabled. | | |
| | | [6:5] | RESERVED | | Reserved. | 0x0 | R |
| | | [4:0] | L_RD | | Received ILAS configuration on Lane 0. L is the number of lanes per converter device. Link information received on Lane 0 as specified in Section 8.3 of JESD204B. | 0x0 | R |
| | | | | 00000 | 1 lane per converter device. | | |
| | | | | 00001 | 2 lanes per converter device. | | |
| | | | | 00011 | 4 lanes per converter device. | | |
| | | | | 00111 | 8 lanes per converter device. | | |
| 0x404 | F_REG | [7:0] | F_RD | | Received ILAS configuration on Lane 0. F is the number of octets per frame. Settings of 1, 2, and 4 are valid (value in register is F − 1). Link information received on Lane 0 as specified in Section 8.3 of JESD204B. | 0x0 | R |
| | | | | 0 | 1 octet per frame. | | |
| | | | | 1 | 2 octets per frame. | | |
| | | | | 11 | 4 octets per frame. | | |
| 0x405 | K_REG | [7:5] | RESERVED | | Reserved. | 0x0 | R |
| | | [4:0] | K_RD | | Received ILAS configuration on Lane 0. K is the number of frames per multiframe. Settings of 16 or 32 are valid. On this device, all modes use K = 32 (value in register is K − 1). Link information received on Lane 0 as specified in Section 8.3 of JESD204B. | 0x0 | R |
| | | | | 01111 | 16 frames per multiframe. | | |
| | | | | 11111 | 32 frames per multiframe. | | |
| 0x406 | M_REG | [7:0] | M_RD | | Received ILAS configuration on Lane 0. M is the number of converters per device. Link information received on Lane 0 as specified in Section 8.3 of JESD204B. M is 1 for real interface and 2 for complex interface (value in register is M − 1). | 0x0 | R |
| 0x407 | CS_N_REG | [7:6] | CS_RD | | Received ILAS configuration on Lane 0. CS is the number of control bits per sample. Link information received on Lane 0 as specified in Section 8.3 of JESD204B. CS is always 0 on this device. | 0x0 | R |
| | | 5 | RESERVED | | Reserved. | 0x0 | R |
| | | [4:0] | N_RD | | Received ILAS configuration on Lane 0. N is the converter resolution. Value in register is N − 1 (for example, 16 bits = 0b01111). | 0x0 | R |

| Hex. Addr. | Name | Bits | Bit Name | Settings | Description | Reset | Access |
|---|---|---|---|---|---|---|---|
| 0x408 | NP_REG | [7:5] | SUBCLASSV_RD | | Received ILAS configuration on Lane 0. SUBCLASSV is the device subclass version. Link information received on Lane 0 as specified in Section 8.3 of JESD204B. | 0x0 | R |
| | | | | 000 | Subclass 0. | | |
| | | | | 001 | Subclass 1. | | |
| | | [4:0] | NP_RD | | Received ILAS configuration on Lane 0. NP is the total number of bits per sample. Link information received on Lane 0 as specified in Section 8.3 of JESD204B. Value in register is NP − 1, for example, 16 bits per sample = 0b01111. | 0x0 | R |
| 0x409 | S_REG | [7:5] | JESDV_RD | | Received ILAS configuration on Lane 0. JESDV is the JESD204x version. Link information received on Lane 0 as specified in Section 8.3 of JESD204B. | 0x0 | R |
| | | | | 000 | JESD204A. | | |
| | | | | 001 | JESD204B. | | |
| | | [4:0] | S_RD | | Received ILAS configuration on Lane 0. S is the number of samples per converter per frame cycle. Link information received on Lane 0 as specified in Section 8.3 of JESD204B. Value in register is S − 1. | 0x0 | R |
| 0x40A | HD_CF_REG | 7 | HD_RD | | Received ILAS configuration on Lane 0. HD is the high density format. Refer to Section 5.1.3 of JESD204B standard. Link information received on Lane 0 as specified in Section 8.3 of JESD204B. | 0x0 | R |
| | | | | 0 | Low density mode. | | |
| | | | | 1 | High density mode. | | |
| | | [6:5] | RESERVED | | Reserved. | 0x0 | R |
| | | [4:0] | CF_RD | | Received ILAS configuration on Lane 0. CF is the number of control words per frame clock period per link. Link information received on Lane 0 as specified in Section 8.3 of JESD204B. CF is always 0 on this device. | 0x0 | R |
| 0x40B | RES1_REG | [7:0] | RES1_RD | | Received ILAS configuration on Lane 0. Reserved Field 1. Link information received on Lane 0 as specified in Section 8.3 of JESD204B. | 0x0 | R |
| 0x40C | RES2_REG | [7:0] | RES2_RD | | Received ILAS configuration on Lane 0. Reserved Field 2. Link information received on Lane 0 as specified in Section 8.3 of JESD204B. | 0x0 | R |
| 0x40D | CHECKSUM0_REG | [7:0] | LL_FCHK0 | | Received checksum during ILAS on Lane 0. Checksum for Lane 0. Link information received on Lane 0 as specified in Section 8.3 of JESD204B. | 0x0 | R |

| Hex. Addr. | Name | Bits | Bit Name | Settings | Description | Reset | Access |
|---|---|---|---|---|---|---|---|
| 0x40E | COMPSUM0_REG | [7:0] | LL_FCMP0 | | Computed checksum on Lane 0. Computed checksum for Lane 0. The JESD204B Rx computes the checksum of the link information received on Lane 0 as specified in Section 8.3 of JESD204B. The computation method is set by the CHECKSUM_MODE bit (Register 0x300, Bit 6) and must match the likewise calculated checksum in Register 0x40D. | 0x0 | R |
| 0x412 | LID1_REG | [7:5] | RESERVED | | Reserved. | 0x0 | R |
| | | [4:0] | LL_LID1 | | Received ILAS LID configuration on Lane 1. Lane identification for Lane 1. Link information received on Lane 0 as specified in Section 8.3 of JESD204B. | 0x0 | R |
| 0x415 | CHECKSUM1_REG | [7:0] | LL_FCHK1 | | Received checksum during ILAS on lane 1. Checksum for Lane 1. Link information received on Lane 0 as specified in Section 8.3 of JESD204B. | 0x0 | R |
| 0x416 | COMPSUM1_REG | [7:0] | LL_FCMP1 | | Computed checksum on Lane 1. Computed checksum for Lane 1 (see description for Register 0x40E). | 0x0 | R |
| 0x41A | LID2_REG | [7:5] | RESERVED | | Reserved. | 0x0 | R |
| | | [4:0] | LL_LID2 | | Received ILAS LID configuration on Lane 2. Lane identification for Lane 2. | 0x0 | R |
| 0x41D | CHECKSUM2_REG | [7:0] | LL_FCHK2 | | Received checksum during ILAS on Lane 2. Checksum for Lane 2. | 0x0 | R |
| 0x41E | COMPSUM2_REG | [7:0] | LL_FCMP2 | | Computed checksum on Lane 2. Computed checksum for Lane 2 (see description for Register 0x40E). | 0x0 | R |
| 0x422 | LID3_REG | [7:5] | RESERVED | | Reserved. | 0x0 | R |
| | | [4:0] | LL_LID3 | | Received ILAS LID configuration on Lane 3. Lane identification for Lane 3. | 0x0 | R |
| 0x425 | CHECKSUM3_REG | [7:0] | LL_FCHK3 | | Received checksum during ILAS on Lane 3. Checksum for Lane 3. | 0x0 | R |
| 0x426 | COMPSUM3_REG | [7:0] | LL_FCMP3 | | Computed checksum on Lane 3. Computed checksum for Lane 3 (see description for Register 0x40E). | 0x0 | R |
| 0x42A | LID4_REG | [7:5] | RESERVED | | Reserved. | 0x0 | R |
| | | [4:0] | LL_LID4 | | Received ILAS LID configuration on Lane 4. Lane identification for Lane 4. | 0x0 | R |
| 0x42D | CHECKSUM4_REG | [7:0] | LL_FCHK4 | | Received checksum during ILAS on Lane 4. Checksum for Lane 4. | 0x0 | R |
| 0x42E | COMPSUM4_REG | [7:0] | LL_FCMP4 | | Computed checksum on Lane 4. Computed checksum for Lane 4 (see description for Register 0x40E). | 0x0 | R |
| 0x432 | LID5_REG | [7:5] | RESERVED | | Reserved. | 0x0 | R |
| | | [4:0] | LL_LID5 | | Received ILAS LID configuration on Lane 5. Lane identification for Lane 5. | 0x0 | R |
| 0x435 | CHECKSUM5_REG | [7:0] | LL_FCHK5 | | Received checksum during ILAS on Lane 5. Checksum for Lane 5. | 0x0 | R |
| 0x436 | COMPSUM5_REG | [7:0] | LL_FCMP5 | | Computed checksum on Lane 5. Computed checksum for Lane 5 (see description for Register 0x40E). | 0x0 | R |

| Hex. Addr. | Name | Bits | Bit Name | Settings | Description | Reset | Access |
|---|---|---|---|---|---|---|---|
| 0x43A | LID6_REG | [7:5] | RESERVED | | Reserved. | 0x0 | R |
| | | [4:0] | LL_LID6 | | Received ILAS LID configuration on Lane 6. Lane identification for Lane 6. | 0x0 | R |
| 0x43D | CHECKSUM6_REG | [7:0] | LL_FCHK6 | | Received checksum during ILAS on Lane 6. Checksum for Lane 6. | 0x0 | R |
| 0x43E | COMPSUM6_REG | [7:0] | LL_FCMP6 | | Computed checksum on Lane 6. Computed checksum for Lane 6 (see description for Register 0x40E). | 0x0 | R |
| 0x442 | LID7_REG | [7:5] | RESERVED | | Reserved. | 0x0 | R |
| | | [4:0] | LL_LID7 | | Received ILAS LID configuration on Lane 7. Lane identification for Lane 7. | 0x0 | R |
| 0x445 | CHECKSUM7_REG | [7:0] | LL_FCHK7 | | Received checksum during ILAS on Lane 7. Checksum for Lane 7. | 0x0 | R |
| 0x446 | COMPSUM7_REG | [7:0] | LL_FCMP7 | | Computed checksum on Lane 7. Computed checksum for Lane 7 (see description for Register 0x40E). | 0x0 | R |
| 0x450 | ILS_DID | [7:0] | DID | | Device (= link) identification number. DID is the device ID number. Link information received on Lane 0 as specified in Section 8.3 of JESD204B. Must be set to the value read in Register 0x400. This signal must only be programmed while the QBD is held in soft reset (Register 0x475, Bit 3), and must not be changed during normal operation. | 0x0 | R/W |
| 0x451 | ILS_BID | [7:0] | BID | | Bank ID, extension to DID. This signal must only be programmed while the QBD is held in soft reset (Register 0x475, Bit 3), and must not be changed during normal operation. | 0x0 | R/W |
| 0x452 | ILS_LID0 | 7 | RESERVED | | Reserved. | 0x0 | R |
| | | 6 | ADJDIR | | Direction to adjust DAC LMFC (Subclass 2 only). ADJDIR is the direction to adjust DAC LMFC. Link information received on Lane 0 as specified in Section 8.3 of JESD204B. This signal must only be programmed while the QBD is held in soft reset (Register 0x475, Bit 3), and must not be changed during normal operation. | 0x0 | R/W |
| | | 5 | PHADJ | | Phase adjustment to DAC (Subclass 2 only). PHADJ is the phase adjustment request to the DAC. Link information received on Lane 0 as specified in Section 8.3 of JESD204B. This signal must only be programmed while the QBD is held in soft reset (Register 0x475, Bit 3), and must not be changed during normal operation. | 0x0 | R/W |
| | | [4:0] | LID0 | | Lane identification number (within link). LID0 is the lane identification for Lane 0. Link information received on Lane 0 as specified in Section 8.3 of JESD204B. This signal must only be programmed while the QBD is held in soft reset (Register 0x475, Bit 3), and must not be changed during normal operation. | 0x0 | R/W |

| Hex. Addr. | Name | Bits | Bit Name | Settings | Description | Reset | Access |
|---|---|---|---|---|---|---|---|
| 0x453 | ILS_SCR_L | 7 | SCR | | Scramble enable. SCR is the Rx descrambling enable. This signal must only be programmed while the QBD is held in soft reset (Register 0x475, Bit 3), and must not be changed during normal operation. | 0x1 | R/W |
| | | | | 0 | Descrambling is disabled. | | |
| | | | | 1 | Descrambling is enabled. | | |
| | | [6:5] | RESERVED | | Reserved. | 0x0 | R |
| | | [4:0] | L | | Number of lanes per converter (minus 1). L is the number of lanes per converter device. Settings of 1, 2, 3, 4, 6, and 8 are valid. Refer to Table 14 and Table 15. | 0x7 | R |
| 0x454 | ILS_F | [7:0] | F | | Number of octets per frame (minus 1). This value of F is not used to soft configure the QBD. Register CTRLREG1 is used to soft-configure the QBD. | 0x0 | R |
| 0x455 | ILS_K | [7:5] | RESERVED | | Reserved. | 0x0 | R |
| | | [4:0] | K | | Number of frames per multiframe (minus 1). K is the number of frames per multiframe. On this device, all modes use K = 32 (value in register is K − 1). This signal must only be programmed while the QBD is held in soft reset (Register 0x475, Bit 3), and must not be changed during normal operation. | 0x1F | R/W |
| | | | | 01111 | 16 frames per multiframe. | | |
| | | | | 11111 | 32 frames per multiframe. | | |
| 0x456 | ILS_M | [7:0] | M | | Number of converters per device (minus 1). M is the number of converters/device. Settings of 1 and 2 are valid. Refer to Table 14 and Table 15. | 0x1 | R |
| 0x457 | ILS_CS_N | [7:6] | CS | | Number of control bits per sample. CS is the number of control bits per sample. Must be set to 0. Control bits are not supported. | 0x0 | R |
| | | 5 | RESERVED | | Reserved. | 0x0 | R |
| | | [4:0] | N | | Converter resolution (minus 1). N is the converter resolution. Must be set to 16 (0x0F). | 0xF | R |
| 0x458 | ILS_NP | [7:5] | SUBCLASSV | | Device subclass version. SUBCLASSV is the device subclass version. This signal must only be programmed while the QBD is held in soft reset (Register 0x475, Bit 3), and must not be changed during normal operation. | 0x0 | R/W |
| | | | | 000 | Subclass 0. | | |
| | | | | 001 | Subclass 1. | | |
| | | | | 010 | Subclass 2 (not supported). | | |
| | | [4:0] | NP | | Total number of bits per sample (minus 1). NP is the total number of bits per sample. Must be set to 16 (0x0F). Refer to Table 14 and Table 15. | 0xF | R |

| Hex. Addr. | Name | Bits | Bit Name | Settings | Description | Reset | Access |
|---|---|---|---|---|---|---|---|
| 0x459 | ILS_S | [7:5] | JESDV | | JESD204x version. JESDV is the JESD204x version. This signal must only be programmed while the QBD is held in soft reset (Register 0x475, Bit 3), and must not be changed during normal operation. | 0x0 | R/W |
| | | | | 000 | JESD204A. | | |
| | | | | 001 | JESD204B. | | |
| | | [4:0] | S | | Number of samples per converter per frame cycle (minus 1). S is the number of samples per converter per frame cycle. Settings of 1 and 2 are valid. Refer to Table 14 and Table 15. | 0x1 | R |
| 0x45A | ILS_HD_CF | 7 | HD | | High density format. HD is the high density mode. Refer to Section 5.1.3 of JESD204B standard. | 0x1 | R |
| | | | | 0 | Low density mode. | | |
| | | | | 1 | High density mode. | | |
| | | [6:5] | RESERVED | | Reserved. | 0x0 | R |
| | | [4:0] | CF | | Number of control bits per sample. CF is the number of control words per frame clock period per link. Must be set to 0. Control bits are not supported. | 0x0 | R |
| 0x45B | ILS_RES1 | [7:0] | RES1 | | Reserved. Reserved Field 1. This signal must only be programmed while the QBD is held in soft reset (Register 0x475, Bit 3), and must not be changed during normal operation. | 0x0 | R/W |
| 0x45C | ILS_RES2 | [7:0] | RES2 | | Reserved. Reserved Field 2. This signal must only be programmed while the QBD is held in soft reset (Register 0x475, Bit 3), and must not be changed during normal operation. | 0x0 | R/W |
| 0x45D | ILS_CHECKSUM | [7:0] | FCHK0 | | Link configuration checksum. Checksum for Lane 0. The checksum for the values programmed into Register 0x450 to Register 0x45C must be calculated according to Section 8.3 of the JESD204B specification and written to this register (SUM(Register 0x450 to Register 0x45C) % 256). This signal must only be programmed while the QBD is held in soft reset (Register 0x475, Bit 3), and must not be changed during normal operation. | 0x0 | R/W |
| 0x46C | LANE_DESKEW | 7 | ILD7 | | Interlane deskew status for Lane 7 (ignore this output when NO_ILAS = 1). | 0x0 | R |
| | | | | 0 | Deskew failed. | | |
| | | | | 1 | Deskew achieved. | | |
| | | 6 | ILS6 | | Initial lane synchronization status for Lane 6 (ignore this output when NO_ILAS = 1). | 0x0 | R |
| | | | | 0 | Synchronization lost. | | |
| | | | | 1 | Synchronization achieved. | | |

| Hex. Addr. | Name | Bits | Bit Name | Settings | Description | Reset | Access |
|---|---|---|---|---|---|---|---|
| | | 5 | ILD5 | | Interlane deskew status for Lane 5 (ignore this output when NO_ILAS = 1). | 0x0 | R |
| | | | | 0 | Deskew failed. | | |
| | | | | 1 | Deskew achieved. | | |
| | | 4 | ILD4 | | Interlane deskew status for Lane 4 (ignore this output when NO_ILAS = 1). | 0x0 | R |
| | | | | 0 | Deskew failed. | | |
| | | | | 1 | Deskew achieved. | | |
| | | 3 | ILD3 | | Interlane deskew status for Lane 3 (ignore this output when NO_ILAS = 1). | 0x0 | R |
| | | | | 0 | Deskew failed. | | |
| | | | | 1 | Deskew achieved. | | |
| | | 2 | ILD2 | | Interlane deskew status for Lane 2 (ignore this output when NO_ILAS = 1). | 0x0 | R |
| | | | | 0 | Deskew failed. | | |
| | | | | 1 | Deskew achieved. | | |
| | | 1 | ILD1 | | Interlane deskew status for Lane 1 (ignore this output when NO_ILAS = 1). | 0x0 | R |
| | | | | 0 | Deskew failed. | | |
| | | | | 1 | Deskew achieved. | | |
| | | 0 | ILD0 | | Interlane deskew status for Lane 0 (ignore this output when NO_ILAS = 1). | 0x0 | R |
| | | | | 0 | Deskew failed. | | |
| | | | | 1 | Deskew achieved. | | |
| 0x46D | BAD_DISPARITY | 7 | BDE7 | | Bad disparity error status for Lane 7. | 0x0 | R |
| | | | | 0 | Error count < ETH[7:0] value. | | |
| | | | | 1 | Error count ≥ ETH[7:0] value. | | |
| | | 6 | BDE6 | | Bad disparity error status for Lane 6. | 0x0 | R |
| | | | | 0 | Error count < ETH[7:0] value. | | |
| | | | | 1 | Error count ≥ ETH[7:0] value. | | |
| | | 5 | BDE5 | | Bad disparity errors status for Lane 5. | 0x0 | R |
| | | | | 0 | Error count < ETH[7:0] value. | | |
| | | | | 1 | Error count ≥ ETH[7:0] value. | | |
| | | 4 | BDE4 | | Bad disparity error status for Lane 4. | 0x0 | R |
| | | | | 0 | Error count < ETH[7:0] value. | | |
| | | | | 1 | Error count ≥ ETH[7:0] value. | | |
| | | 3 | BDE3 | | Bad disparity error status for Lane 3. | 0x0 | R |
| | | | | 0 | Error count < ETH[7:0] value. | | |
| | | | | 1 | Error count ≥ ETH[7:0] value. | | |
| | | 2 | BDE2 | | Bad disparity error status for Lane 2. | 0x0 | R |
| | | | | 0 | Error count < ETH[7:0] value. | | |
| | | | | 1 | Error count ≥ ETH[7:0] value. | | |
| | | 1 | BDE1 | | Bad disparity error status for Lane 1. | 0x0 | R |
| | | | | 0 | Error count < ETH[7:0] value. | | |
| | | | | 1 | Error count ≥ ETH[7:0] value. | | |
| | | 0 | BDE0 | | Bad disparity error status for Lane 0. | 0x0 | R |
| | | | | 0 | Error count < ETH[7:0] value. | | |
| | | | | 1 | Error count ≥ ETH[7:0] value. | | |

| Hex. Addr. | Name | Bits | Bit Name | Settings | Description | Reset | Access |
|---|---|---|---|---|---|---|---|
| 0x46E | NOT_IN_TABLE | 7 | NIT7 | | Not in table error status for Lane 7. | 0x0 | R |
| | | | | 0 | Error count < ETH[7:0] value. | | |
| | | | | 1 | Error count ≥ ETH[7:0] value. | | |
| | | 6 | NIT6 | | Not in table error status for Lane 6. | 0x0 | R |
| | | | | 0 | Error count < ETH[7:0] value. | | |
| | | | | 1 | Error count ≥ ETH[7:0] value. | | |
| | | 5 | NIT5 | | Not in table errors status for Lane 5. | 0x0 | R |
| | | | | 0 | Error count < ETH[7:0] value. | | |
| | | | | 1 | Error count ≥ ETH[7:0] value. | | |
| | | 4 | NIT4 | | Not in table error status for Lane 4. | 0x0 | R |
| | | | | 0 | Error count < ETH[7:0] value. | | |
| | | | | 1 | Error count ≥ ETH[7:0] value. | | |
| | | 3 | NIT3 | | Not in table error status for Lane 3. | 0x0 | R |
| | | | | 0 | Error count < ETH[7:0] value. | | |
| | | | | 1 | Error count ≥ ETH[7:0] value. | | |
| | | 2 | NIT2 | | Not in table error status for Lane 2. | 0x0 | R |
| | | | | 0 | Error count < ETH[7:0] value. | | |
| | | | | 1 | Error count ≥ ETH[7:0] value. | | |
| | | 1 | NIT1 | | Not in table error status for Lane 1. | 0x0 | R |
| | | | | 0 | Error count < ETH[7:0] value. | | |
| | | | | 1 | Error count ≥ ETH[7:0] value. | | |
| | | 0 | NIT0 | | Not in table error status for Lane 0. | 0x0 | R |
| | | | | 0 | Error count < ETH[7:0] value. | | |
| | | | | 1 | Error count ≥ ETH[7:0] value. | | |
| 0x46F | UNEXPECTED_KCHAR | 7 | UEK7 | | Unexpected K character error status for Lane 7. | 0x0 | R |
| | | | | 0 | Error count < ETH[7:0] value. | | |
| | | | | 1 | Error count ≥ ETH[7:0] value. | | |
| | | 6 | UEK6 | | Unexpected K character error status for Lane 6. | 0x0 | R |
| | | | | 0 | Error count < ETH[7:0] value. | | |
| | | | | 1 | Error count ≥ ETH[7:0] value. | | |
| | | 5 | UEK5 | | Unexpected K character error status for Lane 5. | 0x0 | R |
| | | | | 0 | Error count < ETH[7:0] value. | | |
| | | | | 1 | Error count ≥ ETH[7:0] value. | | |
| | | 4 | UEK4 | | Unexpected K character error status for Lane 4. | 0x0 | R |
| | | | | 0 | Error count < ETH[7:0] value. | | |
| | | | | 1 | Error count ≥ ETH[7:0] value. | | |
| | | 3 | UEK3 | | Unexpected K character error status for Lane 3. | 0x0 | R |
| | | | | 0 | Error count < ETH[7:0] value. | | |
| | | | | 1 | Error count ≥ ETH[7:0] value. | | |
| | | 2 | UEK2 | | Unexpected K character error status for Lane 2. | 0x0 | R |
| | | | | 0 | Error count < ETH[7:0] value. | | |
| | | | | 1 | Error count ≥ ETH[7:0] value. | | |
| | | 1 | UEK1 | | Unexpected K character error status for Lane 1. | 0x0 | R |
| | | | | 0 | Error count < ETH[7:0] value. | | |
| | | | | 1 | Error count ≥ ETH[7:0] value. | | |

| Hex. Addr. | Name | Bits | Bit Name | Settings | Description | Reset | Access |
|---|---|---|---|---|---|---|---|
| | | 0 | UEK0 | | Unexpected K character error status for Lane 0. | 0x0 | R |
| | | | | 0 | Error count < ETH[7:0] value. | | |
| | | | | 1 | Error count ≥ ETH[7:0] value. | | |
| 0x470 | CODE_GRP_SYNC | 7 | CGS7 | | Code group sync status for Lane 7. | 0x0 | R |
| | | | | 0 | Synchronization lost. | | |
| | | | | 1 | Synchronization achieved. | | |
| | | 6 | CGS6 | | Code group sync status for Lane 6. | 0x0 | R |
| | | | | 0 | Synchronization lost. | | |
| | | | | 1 | Synchronization achieved. | | |
| | | 5 | CGS5 | | Code group sync status for Lane 5. | 0x0 | R |
| | | | | 0 | Synchronization lost. | | |
| | | | | 1 | Synchronization achieved. | | |
| | | 4 | CGS4 | | Code group sync status for Lane 4. | 0x0 | R |
| | | | | 0 | Synchronization lost. | | |
| | | | | 1 | Synchronization achieved. | | |
| | | 3 | CGS3 | | Code group sync status for Lane 3. | 0x0 | R |
| | | | | 0 | Synchronization lost. | | |
| | | | | 1 | Synchronization achieved. | | |
| | | 2 | CGS2 | | Code group sync status for Lane 2. | 0x0 | R |
| | | | | 0 | Synchronization lost. | | |
| | | | | 1 | Synchronization achieved. | | |
| | | 1 | CGS1 | | Code group sync status for Lane 1. | 0x0 | R |
| | | | | 0 | Synchronization lost. | | |
| | | | | 1 | Synchronization achieved. | | |
| | | 0 | CGS0 | | Code group sync status for Lane 0. | 0x0 | R |
| | | | | 0 | Synchronization lost. | | |
| | | | | 1 | Synchronization achieved. | | |
| 0x471 | FRAME_SYNC | 7 | FS7 | | Frame sync status for Lane 7 (ignore this output when NO_ILAS = 1). | 0x0 | R |
| | | | | 0 | Synchronization lost. | | |
| | | | | 1 | Synchronization achieved. | | |
| | | 6 | FS6 | | Frame sync status for Lane 6 (ignore this output when NO_ILAS = 1). | 0x0 | R |
| | | | | 0 | Synchronization lost. | | |
| | | | | 1 | Synchronization achieved. | | |
| | | 5 | FS5 | | Frame sync status for Lane 5 (ignore this output when NO_ILAS = 1). | 0x0 | R |
| | | | | 0 | Synchronization lost. | | |
| | | | | 1 | Synchronization achieved. | | |
| | | 4 | FS4 | | Frame sync status for Lane 4 (ignore this output when NO_ILAS = 1). | 0x0 | R |
| | | | | 0 | Synchronization lost. | | |
| | | | | 1 | Synchronization achieved. | | |
| | | 3 | FS3 | | Frame sync status for Lane 3 (ignore this output when NO_ILAS = 1). | 0x0 | R |
| | | | | 0 | Synchronization lost. | | |
| | | | | 1 | Synchronization achieved. | | |

| Hex. Addr. | Name | Bits | Bit Name | Settings | Description | Reset | Access |
|---|---|---|---|---|---|---|---|
| | | 2 | FS2 | | Frame sync status for Lane 2 (ignore this output when NO_ILAS = 1). | 0x0 | R |
| | | | | 0 | Synchronization lost. | | |
| | | | | 1 | Synchronization achieved. | | |
| | | 1 | FS1 | | Frame sync status for Lane 1 (ignore this output when NO_ILAS = 1). | 0x0 | R |
| | | | | 0 | Synchronization lost. | | |
| | | | | 1 | Synchronization achieved. | | |
| | | 0 | FS0 | | Frame sync status for Lane 0 (ignore this output when NO_ILAS = 1). | 0x0 | R |
| | | | | 0 | Synchronization lost. | | |
| | | | | 1 | Synchronization achieved. | | |
| 0x472 | GOOD_CHECKSUM | 7 | CKS7 | | Computed checksum status for Lane 7 (ignore this output when NO_ILAS = 1). | 0x0 | R |
| | | | | 0 | Checksum is incorrect. | | |
| | | | | 1 | Checksum is correct. | | |
| | | 6 | CKS6 | | Computed checksum status for Lane 6 (ignore this output when NO_ILAS = 1). | 0x0 | R |
| | | | | 0 | Checksum is incorrect. | | |
| | | | | 1 | Checksum is correct. | | |
| | | 5 | CKS5 | | Computed checksum status for Lane 5 (ignore this output when NO_ILAS = 1). | 0x0 | R |
| | | | | 0 | Checksum is incorrect. | | |
| | | | | 1 | Checksum is correct. | | |
| | | 4 | CKS4 | | Computed checksum status for Lane 4 (ignore this output when NO_ILAS = 1). | 0x0 | R |
| | | | | 0 | Checksum is incorrect. | | |
| | | | | 1 | Checksum is correct. | | |
| | | 3 | CKS3 | | Computed checksum status for Lane 3 (ignore this output when NO_ILAS = 1). | 0x0 | R |
| | | | | 0 | Checksum is incorrect. | | |
| | | | | 1 | Checksum is correct. | | |
| | | 2 | CKS2 | | Computed checksum status for Lane 2 (ignore this output when NO_ILAS = 1). | 0x0 | R |
| | | | | 0 | Checksum is incorrect. | | |
| | | | | 1 | Checksum is correct. | | |
| | | 1 | CKS1 | | Computed checksum status for Lane 1 (ignore this output when NO_ILAS = 1). | 0x0 | R |
| | | | | 0 | Checksum is incorrect. | | |
| | | | | 1 | Checksum is correct. | | |
| | | 0 | CKS0 | | Computed checksum status for Lane 0 (ignore this output when NO_ILAS = 1). | 0x0 | R |
| | | | | 0 | Checksum is incorrect. | | |
| | | | | 1 | Checksum is correct. | | |

| Hex. Addr. | Name | Bits | Bit Name | Settings | Description | Reset | Access |
|---|---|---|---|---|---|---|---|
| 0x473 | INIT_LANE_SYNC | 7 | ILS7 | | Initial lane synchronization status for Lane 7 (ignore this output when NO_ILAS = 1). | 0x0 | R |
| | | | | 0 | Synchronization lost. | | |
| | | | | 1 | Synchronization achieved. | | |
| | | 6 | ILS6 | | Initial lane synchronization status for Lane 6 (ignore this output when NO_ILAS = 1). | 0x0 | R |
| | | | | 0 | Synchronization lost. | | |
| | | | | 1 | Synchronization achieved. | | |
| | | 5 | ILS5 | | Initial lane synchronization status for Lane 5 (ignore this output when NO_ILAS = 1). | 0x0 | R |
| | | | | 0 | Synchronization lost. | | |
| | | | | 1 | Synchronization achieved. | | |
| | | 4 | ILS4 | | Initial lane synchronization status for Lane 4 (ignore this output when NO_ILAS = 1). | 0x0 | R |
| | | | | 0 | Synchronization lost. | | |
| | | | | 1 | Synchronization achieved. | | |
| | | 3 | ILS3 | | Initial lane synchronization status for Lane 3 (ignore this output when NO_ILAS = 1). | 0x0 | R |
| | | | | 0 | Synchronization lost. | | |
| | | | | 1 | Synchronization achieved. | | |
| | | 2 | ILS2 | | Initial lane synchronization status for Lane 2 (ignore this output when NO_ILAS = 1). | 0x0 | R |
| | | | | 0 | Synchronization lost. | | |
| | | | | 1 | Synchronization achieved. | | |
| | | 1 | ILS1 | | Initial lane synchronization status for Lane 1 (ignore this output when NO_ILAS = 1). | 0x0 | R |
| | | | | 0 | Synchronization lost. | | |
| | | | | 1 | Synchronization achieved. | | |
| | | 0 | ILS0 | | Initial lane synchronization status for Lane 0 (ignore this output when NO_ILAS = 1). | 0x0 | R |
| | | | | 0 | Synchronization lost. | | |
| | | | | 1 | Synchronization achieved. | | |
| 0x475 | CTRLREG0 | 7 | RX_DIS | | Level input: disable deframer receiver when this input = 1. This signal must only be programmed while the QBD is held in soft reset (Register 0x475, Bit 3), and must not be changed during normal operation. | 0x0 | R/W |
| | | | | 1 | Disable character replacement of /A/ and /F/ control characters at the end of received frames and multiframes. | | |
| | | | | 0 | Enables the substitution. | | |
| | | 6 | CHAR_REPL_DIS | | When this input = 1, character replacement at the end of frame/multiframe is disabled. This signal must only be programmed while the QBD is held in soft reset (Register 0x475, Bit 3), and must not be changed during normal operation. | 0x0 | R/W |
| | | [5:4] | RESERVED | | Reserved. | 0x0 | R |

| Hex. Addr. | Name | Bits | Bit Name | Settings | Description | Reset | Access |
|---|---|---|---|---|---|---|---|
| | | 3 | SOFTRST | | Soft reset. Active high synchronous reset. Resets all hardware to power-on state. | 0x0 | R/W |
| | | | | 1 | Disables the deframer reception. | | |
| | | | | 0 | Enable deframer logic. | | |
| | | 2 | FORCESYNCREQ | | Command from application to assert a sync request (SYNCOUT). Active high. | 0x0 | R/W |
| | | 1 | RESERVED | | Reserved. | 0x0 | R |
| | | 0 | REPL_FRM_ENA | | When this level input is set, it enables replacement of frames received in error. This signal must only be programmed while the QBD is held in soft reset (Register 0x475, Bit 3), and must not be changed during normal operation. | 0x1 | R/W |
| 0x476 | CTRLREG1 | [7:5] | RESERVED | | Reserved. | 0x0 | R |
| | | 4 | QUAL_RDERR | | Error reporting behavior for concurrent NIT and RD errors. This signal must only be programmed while the QBD is held in soft reset (Register 0x475, Bit 3), and must not be changed during normal operation. | 0x1 | R/W |
| | | | | 0 | NIT has no effect on RD error. | | |
| | | | | 1 | NIT error masks concurrent RD error. | | |
| | | 3 | DEL_SCR | | Alternative descrambler enable. (see JESD204B Section 5.2.4) This signal must only be programmed while the QBD is held in soft reset (Register 0x475, Bit 3), and must not be changed during normal operation. | 0x0 | R/W |
| | | | | 1 | Descrambling begins at Octet 2 of user data. | | |
| | | | | 0 | Descrambling begins at Octet 0 of user data. This is the common usage. | | |
| | | 2 | CGS_SEL | | Determines the QBD behavior after code group sync has been achieved. This signal must only be programmed while the QBD is held in soft reset (Register 0x475, Bit 3), and must not be changed during normal operation. | 0x1 | R/W |
| | | | | 0 | After code group sync is achieved, the QBD asserts SYNCOUT only if there are sufficient disparity errors as per the JESD204B standard. | | |
| | | | | 1 | After code group sync is achieved, if a /K/ is followed by any character other than an /R/ or another /K/, QBD asserts SYNCOUT. | | |

| Hex. Addr. | Name | Bits | Bit Name | Settings | Description | Reset | Access |
|---|---|---|---|---|---|---|---|
| | | 1 | NO_ILAS | | This signal must only be programmed while the QBD is held in soft reset (Register 0x475, Bit 3), and must not be changed during normal operation. | 0x0 | R/W |
| | | | | 1 | For single-lane operation, ILAS is omitted. Code group sync is followed by user data. | | |
| | | | | 0 | Code group sync is followed by ILAS. For multilane operation, NO_ILAS must always be set to 0. | | |
| | | 0 | FCHK_N | | Checksum calculation method. This signal must only be programmed while the QBD is held in soft reset (Register 0x475, Register 3), and must not be changed during normal operation. | 0x0 | R/W |
| | | | | 0 | Calculate checksum by summing individual fields (this more closely matches the definition of the checksum field in the JESD204B standard. | | |
| | | | | 1 | Calculate checksum by summing the registers containing the packed fields (this setting is provided in case the framer of another vendor performs the calculation with this method). | | |
| 0x477 | CTRLREG2 | 7 | ILS_MODE | | Data link layer test mode. This signal must only be programmed while the QBD is held in soft reset (Register 0x475, Bit 3), and must not be changed during normal operation. | 0x0 | R/W |
| | | | | 0 | Normal mode. | | |
| | | | | 1 | Code group sync pattern is followed by a perpetual ILAS sequence. | | |
| | | 6 | RESERVED | | Reserved. | 0x0 | R |
| | | 5 | REPDATATEST | | Repetitive data test enable, using JTSPAT pattern. To enable the test, ILS_MODE must = 0. This signal must only be programmed while the QBD is held in soft reset (Register 0x475, Bit 3), and must not be changed during normal operation. | 0x0 | R/W |
| | | 4 | QUETESTERR | | Queue test error mode. This signal must only be programmed while the QBD is held in soft reset (Register 0x475, Bit 3), and must not be changed during normal operation. | 0x0 | R/W |
| | | | | 0 | Simultaneous errors on multiple lanes are reported as one error. | | |
| | | | | 1 | Detected errors from all lanes are trapped in a counter and sequentially signaled on $\overline{\text{SYNCOUT}}$. | | |

| Hex. Addr. | Name | Bits | Bit Name | Settings | Description | Reset | Access |
|---|---|---|---|---|---|---|---|
| | | 3 | AR_ECNTR | | Automatic reset of error counter. The error counter that causes assertion of SYNCOUT is automatically reset to 0 when AR_ECNTR = 1. All other counters are unaffected. This signal must only be programmed while the QBD is held in soft reset (Register 0x475, Bit 3), and must not be changed during normal operation. | 0x0 | R/W |
| | | [2:0] | RESERVED | | Reserved. | 0x0 | R |
| 0x478 | KVAL | [7:0] | KSYNC | | Number of 4 × K multiframes during ILS. F is the number of octets per frame. Settings of 1, 2, and 4 are valid. Refer to Table 14 and Table 15. This signal must only be programmed while the QBD is held in soft reset (Register 0x475, Bit 3), and must not be changed during normal operation. | 0x1 | R/W |
| 0x47C | ERRORTHRES | [7:0] | ETH | | Error threshold value. Bad disparity, NIT disparity, and unexpected K character errors are counted and compared to the error threshold value. When the count is equal, either an IRQ is generated or SYNCOUT± is asserted per the mask register settings or both. Function is performed in all lanes. This signal must only be programmed while the QBD is held in soft reset (Register 0x475, Bit 3), and must not be changed during normal operation. | 0xFF | R/W |
| 0x47D | SYNC_ASSERT_MASK | [7:3] | RESERVED | | Reserved. | 0x0 | R |
| | | [2:0] | SYNC_ASSERT_MASK | | SYNCOUT assertion enable mask for BD, NIT, and UEK error conditions. Active high, SYNCOUT assertion enable mask for BD, NIT, and UEK error conditions, respectively. When an error counter, in any lane, has reached the error threshold count, ETH[7:0], and the corresponding SYNC_ASSERT_ MASK bit is set, SYNCOUT is asserted. The mask bits are as follows. Note that the bit sequence is reversed with respect to the other error count controls and the error counters.<br>Bit 2 = bad disparity error (BDE).<br>Bit 1 = not in table error (NIT).<br>Bit 0 = unexpected K (UEK) character error. | 0x7 | R/W |
| 0x480 | ECNT_CTRL0 | [7:6] | RESERVED | | Reserved. | 0x0 | R |
| | | [5:3] | ECNT_ENA0 | | Error counter enable for Lane 0. Counters of each lane are addressed as follows:<br>Bit 2 = unexpected K (UEK) character error.<br>Bit 1 = not in table error (NIT).<br>Bit 0 = bad disparity error (BDE). | 0x7 | R/W |

| Hex. Addr. | Name | Bits | Bit Name | Settings | Description | Reset | Access |
|---|---|---|---|---|---|---|---|
| | | [2:0] | ECNT_RST0 | | Reset error counters for Lane 0, active high. Counters of each lane are addressed as follows:<br>Bit 2 = unexpected K (UEK) character error.<br>Bit 1 = not in table error (NIT).<br>Bit 0 = bad disparity error (BDE). | 0x7 | R/W |
| 0x481 | ECNT_CTRL1 | [7:6] | RESERVED | | Reserved. | 0x0 | R |
| | | [5:3] | ECNT_ENA1 | | Error counters enable for Lane 1, active high. Counters of each lane are addressed as follows:<br>Bit 2 = unexpected K (UEK) character error.<br>Bit 1 = not in table error (NIT).<br>Bit 0 = bad disparity error (BDE). | 0x7 | R/W |
| | | [2:0] | ECNT_RST1 | | Reset error counters for Lane 1, active high. Counters of each lane are addressed as follows:<br>Bit 2 = unexpected K (UEK) character error.<br>Bit 1 = not in table error (NIT).<br>Bit 0 = bad disparity error (BDE). | 0x7 | R/W |
| 0x482 | ECNT_CTRL2 | [7:6] | RESERVED | | Reserved. | 0x0 | R |
| | | [5:3] | ECNT_ENA2 | | Error counters enable for Lane 2, active high. Counters of each lane are addressed as follows:<br>Bit 2 = unexpected K (UEK) character error.<br>Bit 1 = not in table error (NIT).<br>Bit 0 = bad disparity error (BDE). | 0x7 | R/W |
| | | [2:0] | ECNT_RST2 | | Reset error counters for Lane 2, active high. Counters of each lane are addressed as follows:<br>Bit 2 = unexpected K (UEK) character error.<br>Bit 1 = not in table error (NIT).<br>Bit 0 = bad disparity error (BDE). | 0x7 | R/W |
| 0x483 | ECNT_CTRL3 | [7:6] | RESERVED | | Reserved. | 0x0 | R |
| | | [5:3] | ECNT_ENA3 | | Error counters enable for Lane 3, active high. Counters of each lane are addressed as follows:<br>Bit 2 = unexpected K (UEK) character error.<br>Bit 1 = not in table error (NIT).<br>Bit 0 = bad disparity error (BDE). | 0x7 | R/W |
| | | [2:0] | ECNT_RST3 | | Reset error counters for Lane 3, active high. Counters of each lane are addressed as follows:<br>Bit 2 = unexpected K (UEK) character error.<br>Bit 1 = not in table error (NIT).<br>Bit 0 = bad disparity error (BDE). | 0x7 | R/W |

| Hex. Addr. | Name | Bits | Bit Name | Settings | Description | Reset | Access |
|---|---|---|---|---|---|---|---|
| 0x484 | ECNT_CTRL4 | [7:6] | RESERVED | | Reserved. | 0x0 | R |
| | | [5:3] | ECNT_ENA4 | | Error counters enable for Lane 4, active high. Counters of each lane are addressed as follows:<br>Bit 2 = unexpected K (UEK) character error.<br>Bit 1 = not in table error (NIT).<br>Bit 0 = bad disparity error (BDE). | 0x7 | R/W |
| | | [2:0] | ECNT_RST4 | | Reset error counters for Lane 4, active high. Counters of each lane are addressed as follows:<br>Bit 2 = unexpected K (UEK) character error.<br>Bit 1 = not in table error (NIT).<br>Bit 0 = bad disparity error (BDE). | 0x7 | R/W |
| 0x485 | ECNT_CTRL5 | [7:6] | RESERVED | | Reserved. | 0x0 | R |
| | | [5:3] | ECNT_ENA5 | | Error counters enable for Lane 5, active high. Counters of each lane are addressed as follows:<br>Bit 2 = unexpected K (UEK) character error.<br>Bit 1 = not in table error (NIT).<br>Bit 0 = bad disparity error (BDE). | 0x7 | R/W |
| | | [2:0] | ECNT_RST5 | | Reset error counters for Lane 5, active high. Counters of each lane are addressed as follows:<br>Bit 2 = unexpected K (UEK) character error.<br>Bit 1 = not in table error (NIT).<br>Bit 0 = bad disparity error (BDE). | 0x7 | R/W |
| 0x486 | ECNT_CTRL6 | [7:6] | RESERVED | | Reserved. | 0x0 | R |
| | | [5:3] | ECNT_ENA6 | | Error counters enable for Lane 6, active high. Counters of each lane are addressed as follows:<br>Bit 2 = unexpected K (UEK) character error.<br>Bit 1 = not in table error (NIT).<br>Bit 0 = bad disparity error (BDE). | 0x7 | R/W |
| | | [2:0] | ECNT_RST6 | | Reset error counters for Lane 6, active high. Counters of each lane are addressed as follows:<br>Bit 2 = unexpected K (UEK) character error.<br>Bit 1 = not in table error (NIT).<br>Bit 0 = bad disparity error (BDE). | 0x7 | R/W |
| 0x487 | ECNT_CTRL7 | [7:6] | RESERVED | | Reserved. | 0x0 | R |
| | | [5:3] | ECNT_ENA7 | | Error counters enable for Lane 7, active high. Counters of each lane are addressed as follows:<br>Bit 2 = unexpected K (UEK) character error.<br>Bit 1 = not in table error (NIT).<br>Bit 0 = bad disparity error (BDE). | 0x7 | R/W |

| Hex. Addr. | Name | Bits | Bit Name | Settings | Description | Reset | Access |
|---|---|---|---|---|---|---|---|
| | | [2:0] | ECNT_RST7 | | Reset error counters for Lane 7, active high. Counters of each lane are addressed as follows:<br>Bit 2 = unexpected K (UEK) character error.<br>Bit 1 = not in table error (NIT).<br>Bit 0 = bad disparity error (BDE). | 0x7 | R/W |
| 0x488 | ECNT_TCH0 | [7:3] | RESERVED | | Reserved. | 0x0 | R |
| | | [2:0] | ECNT_TCH0 | | Terminal count hold enable of error counters for Lane 0. When set, the designated counter is to hold the terminal count value of 0xFF when it is reached until the counter is reset by the user. Otherwise, the designated counter rolls over. Counters of each lane are addressed as follows:<br>Bit 2 = unexpected K (UEK) character error.<br>Bit 1 = not in table error (NIT).<br>Bit 0 = bad disparity error (BDE).<br>This signal must only be programmed while the QBD is held in soft reset (Register 0x475, Bit 3), and must not be changed during normal operation. | 0x7 | R/W |
| 0x489 | ECNT_TCH1 | [7:3] | RESERVED | | Reserved. | 0x0 | R |
| | | [2:0] | ECNT_TCH1 | | Terminal count hold enable of error counters for Lane 1. When set, the designated counter is to hold the terminal count value of 0xFF when it is reached until the counter is reset by the user. Otherwise, the designated counter rolls over. Counters of each lane are addressed as follows:<br>Bit 2 = unexpected K (UEK) character error.<br>Bit 1 = not in table error (NIT).<br>Bit 0 = bad disparity error (BDE).<br>This signal must only be programmed while the QBD is held in soft reset (Register 0x475, Bit 3), and must not be changed during normal operation. | 0x7 | R/W |
| 0x48A | ECNT_TCH2 | [7:3] | RESERVED | | Reserved. | 0x0 | R |
| | | [2:0] | ECNT_TCH2 | | Terminal count hold enable of error counters for Lane 2. When set, the designated counter is to hold the terminal count value of 0xFF when it is reached until the counter is reset by the user. Otherwise, the designated counter rolls over. Counters of each lane are addressed as follows:<br>Bit 2 = unexpected K (UEK) character error. | 0x7 | R/W |

| Hex. Addr. | Name | Bits | Bit Name | Settings | Description | Reset | Access |
|---|---|---|---|---|---|---|---|
| | | | | | Bit 1 = not in table error (NIT).<br>Bit 0 = bad disparity error (BDE).<br>This signal must only be programmed while the QBD is held in soft reset (Register 0x475, Bit 3), and must not be changed during normal operation. | | |
| 0x48B | ECNT_TCH3 | [7:3] | RESERVED | | Reserved. | 0x0 | R |
| | | [2:0] | ECNT_TCH3 | | Terminal count hold enable of error counters for Lane 3. When set, the designated counter is to hold the terminal count value of 0xFF when it is reached until the counter is reset by the user. Otherwise, the designated counter rolls over. Counters of each lane are addressed as follows:<br>Bit 2 = unexpected K (UEK) character error.<br>Bit 1 = not in table error (NIT).<br>Bit 0 = bad disparity error (BDE).<br>This signal must only be programmed while the QBD is held in soft reset (Register 0x475, Bit 3), and must not be changed during normal operation. | 0x7 | R/W |
| 0x48C | ECNT_TCH4 | [7:3] | RESERVED | | Reserved. | 0x0 | R |
| | | [2:0] | ECNT_TCH4 | | Terminal count hold enable of error counters for Lane 4. When set, the designated counter is to hold the terminal count value of 0xFF when it is reached until the counter is reset by the user. Otherwise, the designated counter rolls over. Counters of each lane are addressed as follows:<br>Bit 2 = unexpected K (UEK) character error.<br>Bit 1 = not in table error (NIT).<br>Bit 0 = bad disparity error (BDE).<br>This signal must only be programmed while the QBD is held in soft reset (Register 0x475, Bit 3), and must not be changed during normal operation. | 0x7 | R/W |
| 0x48D | ECNT_TCH5 | [7:3] | RESERVED | | Reserved. | 0x0 | R |
| | | [2:0] | ECNT_TCH5 | | Terminal count hold enable of error counters for Lane 5. When set, the designated counter is to hold the terminal count value of 0xFF when it is reached until the counter is reset by the user. Otherwise, the designated counter rolls over. Counters of each lane are addressed as follows:<br>Bit 2 = unexpected K (UEK) character error.<br>Bit 1 = not in table error (NIT).<br>Bit 0 = bad disparity error (BDE).<br>This signal must only be programmed while the QBD is held in soft reset (Register 0x475, Bit 3), and must not be changed during normal operation. | 0x7 | R/W |

| Hex. Addr. | Name | Bits | Bit Name | Settings | Description | Reset | Access |
|---|---|---|---|---|---|---|---|
| 0x48E | ECNT_TCH6 | [7:3] | RESERVED | | Reserved. | 0x0 | R |
| | | [2:0] | ECNT_TCH6 | | Terminal count hold enable of error counters for Lane 6. When set, the designated counter is to hold the terminal count value of 0xFF when it is reached until the counter is reset by the user. Otherwise, the designated counter rolls over. Counters of each lane are addressed as follows:<br>Bit 2 = unexpected K (UEK) character error.<br>Bit 1 = not in table error (NIT).<br>Bit 0 = bad disparity error (BDE).<br>This signal must only be programmed while the QBD is held in soft reset (Register 0x475, Bit 3), and must not be changed during normal operation. | 0x7 | R/W |
| 0x48F | ECNT_TCH7 | [7:3] | RESERVED | | Reserved. | 0x0 | R |
| | | [2:0] | ECNT_TCH7 | | Terminal count hold enable of error counters for Lane 7. When set, the designated counter is to hold the terminal count value of 0xFF when it is reached until the counter is reset by the user. Otherwise, the designated counter rolls over. Counters of each lane are addressed as follows:<br>Bit 2 = unexpected K (UEK) character error.<br>Bit 1 = not in table error (NIT).<br>Bit 0 = bad disparity error (BDE).<br>This signal must only be programmed while the QBD is held in soft reset (Register 0x475, Bit 3), and must not be changed during normal operation. | 0x7 | R/W |
| 0x490 | ECNT_STAT0 | [7:4] | RESERVED | | Reserved. | 0x0 | R |
| | | 3 | LANE_ENA0 | | This output indicates if Lane 0 is enabled. | 0x0 | R |
| | | | | 0 | Lane 0 is held in soft reset. | | |
| | | | | 1 | Lane 0 is enabled. | | |
| | | [2:0] | ECNT_TCR0 | | Terminal count reached indicator of error counters for Lane 0. Set to 1 when the corresponding counter terminal count value of 0xFF has been reached. Counters of each lane are addressed as follows.<br>Bit 2 = unexpected K (UEK) character error.<br>Bit 1 = not in table error (NIT).<br>Bit 0 = bad disparity error (BDE). | 0x0 | R |
| 0x491 | ECNT_STAT1 | [7:4] | RESERVED | | Reserved. | 0x0 | R |
| | | 3 | LANE_ENA1 | | This output indicates if Lane 1 is enabled. | 0x0 | R |
| | | | | 0 | Lane 1 is held in soft reset. | | |
| | | | | 1 | Lane 1 is enabled. | | |

| Hex. Addr. | Name | Bits | Bit Name | Settings | Description | Reset | Access |
|---|---|---|---|---|---|---|---|
| | | [2:0] | ECNT_TCR1 | | Terminal count reached indicator of error counters for Lane 1. Set to 1 when the corresponding counter terminal count value of 0xFF has been reached. Counters of each lane are addressed as follows.<br>Bit 2 = unexpected K (UEK) character error.<br>Bit 1 = not in table error (NIT).<br>Bit 0 = bad disparity error (BDE). | 0x0 | R |
| 0x492 | ECNT_STAT2 | [7:4] | RESERVED | | Reserved. | 0x0 | R |
| | | 3 | LANE_ENA2 | | This output indicates if Lane 2 is enabled. | 0x0 | R |
| | | | | 0 | Lane 2 is held in soft reset. | | |
| | | | | 1 | Lane 2 is enabled. | | |
| | | [2:0] | ECNT_TCR2 | | Terminal count reached indicator of error counters for Lane 2. Set to 1 when the corresponding counter terminal count value of 0xFF has been reached. Counters of each lane are addressed as follows.<br>Bit 2 = unexpected K (UEK) character error.<br>Bit 1 = not in table error (NIT).<br>Bit 0 = bad disparity error (BDE). | 0x0 | R |
| 0x493 | ECNT_STAT3 | [7:4] | RESERVED | | Reserved. | 0x0 | R |
| | | 3 | LANE_ENA3 | | This output indicates if Lane 3 is enabled. | 0x0 | R |
| | | | | 0 | Lane 3 is held in soft reset. | | |
| | | | | 1 | Lane 3 is enabled. | | |
| | | [2:0] | ECNT_TCR3 | | Terminal count reached indicator of error counters for Lane 3. Set to 1 when the corresponding counter terminal count value of 0xFF has been reached. Counters of each lane are addressed as follows:<br>Bit 2 = unexpected K (UEK) character error.<br>Bit 1 = not in table error (NIT).<br>Bit 0 = bad disparity error (BDE). | 0x0 | R |
| 0x494 | ECNT_STAT4 | [7:4] | RESERVED | | Reserved. | 0x0 | R |
| | | 3 | LANE_ENA4 | | This output indicates if Lane 4 is enabled. | 0x0 | R |
| | | | | 0 | Lane 4 is held in soft reset. | | |
| | | | | 1 | Lane 4 is enabled. | | |
| | | [2:0] | ECNT_TCR4 | | Terminal count reached indicator of error counters for Lane 4. Set to 1 when the corresponding counter terminal count value of 0xFF has been reached. Counters of each lane are addressed as follows:<br>Bit 2 = unexpected K (UEK) character error.<br>Bit 1 = not in table error (NIT).<br>Bit 0 = bad disparity error (BDE). | 0x0 | R |

| Hex. Addr. | Name | Bits | Bit Name | Settings | Description | Reset | Access |
|---|---|---|---|---|---|---|---|
| 0x495 | ECNT_STAT5 | [7:4] | RESERVED | | Reserved. | 0x0 | R |
| | | 3 | LANE_ENA5 | | This output indicates if Lane 5 is enabled. | 0x0 | R |
| | | | | 0 | Lane 5 is held in soft reset. | | |
| | | | | 1 | Lane 5 is enabled. | | |
| | | [2:0] | ECNT_TCR5 | | Terminal count reached indicator of error counters for Lane 5. Set to 1 when the corresponding counter terminal count value of 0xFF has been reached. Counters of each lane are addressed as follows:<br>Bit 2 = unexpected K (UEK) character error.<br>Bit 1 = not in table error (NIT).<br>Bit 0 = bad disparity error (BDE). | 0x0 | R |
| 0x496 | ECNT_STAT6 | [7:4] | RESERVED | | Reserved. | 0x0 | R |
| | | 3 | LANE_ENA6 | | This output indicates if Lane 6 is enabled. | 0x0 | R |
| | | | | 0 | Lane 6 is held in soft reset. | | |
| | | | | 1 | Lane 6 is enabled. | | |
| | | [2:0] | ECNT_TCR6 | | Terminal count reached indicator of error counters for Lane 6. Set to 1 when the corresponding counter terminal count value of 0xFF has been reached. Counters of each lane are addressed as follows:<br>Bit 2 = unexpected K (UEK) character error.<br>Bit 1 = not in table error (NIT).<br>Bit 0 = bad disparity error (BDE). | 0x0 | R |
| 0x497 | ECNT_STAT7 | [7:4] | RESERVED | | Reserved. | 0x0 | R |
| | | 3 | LANE_ENA7 | | This output indicates if Lane 7 is enabled. | 0x0 | R |
| | | | | 0 | Lane 7 is held in soft reset. | | |
| | | | | 1 | Lane 7 is enabled. | | |
| | | [2:0] | ECNT_TCR7 | | Terminal count reached indicator of error counters for Lane 7. Set to 1 when the corresponding counter terminal count value of 0xFF has been reached. Counters of each lane are addressed as follows:<br>Bit 2 = unexpected K (UEK) character error.<br>Bit 1 = not in table error (NIT).<br>Bit 0 = bad disparity error (BDE). | 0x0 | R |
| 0x498 | BD_CNT0 | [7:0] | BD_CNT0 | | Bad disparity 8-bit error counters for Lane 0. | 0x0 | R |
| 0x499 | BD_CNT1 | [7:0] | BD_CNT1 | | Bad disparity 8-bit error counters for Lane 1. | 0x0 | R |
| 0x49A | BD_CNT2 | [7:0] | BD_CNT2 | | Bad disparity 8-bit error counters for Lane 2. | 0x0 | R |
| 0x49B | BD_CNT3 | [7:0] | BD_CNT3 | | Bad disparity 8-bit error counters for Lane 3. | 0x0 | R |
| 0x49C | BD_CNT4 | [7:0] | BD_CNT4 | | Bad disparity 8-bit error counters for Lane 4. | 0x0 | R |

| Hex. Addr. | Name | Bits | Bit Name | Settings | Description | Reset | Access |
|---|---|---|---|---|---|---|---|
| 0x49D | BD_CNT5 | [7:0] | BD_CNT5 | | Bad disparity 8-bit error counters for Lane 5. | 0x0 | R |
| 0x49E | BD_CNT6 | [7:0] | BD_CNT6 | | Bad disparity 8-bit error counters for Lane 6. | 0x0 | R |
| 0x49F | BD_CNT7 | [7:0] | BD_CNT7 | | Bad disparity 8-bit error counters for Lane 7. | 0x0 | R |
| 0x4A0 | NIT_CNT0 | [7:0] | NIT_CNT0 | | Not in table 8-bit error counters for Lane 0. | 0x0 | R |
| 0x4A1 | NIT_CNT1 | [7:0] | NIT_CNT1 | | Not in table 8-bit error counters for Lane 1. | 0x0 | R |
| 0x4A2 | NIT_CNT2 | [7:0] | NIT_CNT2 | | Not in table 8-bit error counters for Lane 2. | 0x0 | R |
| 0x4A3 | NIT_CNT3 | [7:0] | NIT_CNT3 | | Not in table 8-bit error counters for Lane 3. | 0x0 | R |
| 0x4A4 | NIT_CNT4 | [7:0] | NIT_CNT4 | | Not in table 8-bit error counters for Lane 4. | 0x0 | R |
| 0x4A5 | NIT_CNT5 | [7:0] | NIT_CNT5 | | Not in table 8-bit error counters for Lane 5. | 0x0 | R |
| 0x4A6 | NIT_CNT6 | [7:0] | NIT_CNT6 | | Not in table 8-bit error counters for Lane 6. | 0x0 | R |
| 0x4A7 | NIT_CNT7 | [7:0] | NIT_CNT7 | | Not in table 8-bit error counters for Lane 7. | 0x0 | R |
| 0x4A8 | UEK_CNT0 | [7:0] | UEK_CNT0 | | Unexpected K character 8-bit error counters for Lane 0. | 0x0 | R |
| 0x4A9 | UEK_CNT1 | [7:0] | UEK_CNT1 | | Unexpected K character 8-bit error counters for Lane 1. | 0x0 | R |
| 0x4AA | UEK_CNT2 | [7:0] | UEK_CNT2 | | Unexpected K character 8-bit error counters for Lane 2. | 0x0 | R |
| 0x4AB | UEK_CNT3 | [7:0] | UEK_CNT3 | | Unexpected K character 8-bit error counters for Lane 3. | 0x0 | R |
| 0x4AC | UEK_CNT4 | [7:0] | UEK_CNT4 | | Unexpected K character 8-bit error counters for Lane 4. | 0x0 | R |
| 0x4AD | UEK_CNT5 | [7:0] | UEK_CNT5 | | Unexpected K character 8-bit error counters for Lane 5. | 0x0 | R |
| 0x4AE | UEK_CNT6 | [7:0] | UEK_CNT6 | | Unexpected K character 8-bit error counters for Lane 6. | 0x0 | R |
| 0x4AF | UEK_CNT7 | [7:0] | UEK_CNT7 | | Unexpected K character 8-bit error counters for Lane 7. | 0x0 | R |
| 0x4B0 | LINK_STATUS0 | 7 | BDE0 | | Bad disparity errors status for Lane 0. | 0x0 | R |
| | | | | 0 | Error count < ETH[7:0] value. | | |
| | | | | 1 | Error count ≥ ETH[7:0] value. | | |
| | | 6 | NIT0 | | Not in table errors status for Lane 0. | 0x0 | R |
| | | | | 0 | Error count < ETH[7:0] value. | | |
| | | | | 1 | Error count ≥ ETH[7:0] value. | | |
| | | 5 | UEK0 | | Unexpected K character errors status for Lane 0. | 0x0 | R |
| | | | | 0 | Error count < ETH[7:0] value. | | |
| | | | | 1 | Error count ≥ ETH[7:0] value. | | |
| | | 4 | ILD0 | | Interlane deskew status for Lane 0 (ignore this output when NO_ILAS = 1). | 0x0 | R |
| | | | | 0 | Deskew failed. | | |
| | | | | 1 | Deskew achieved. | | |

| Hex. Addr. | Name | Bits | Bit Name | Settings | Description | Reset | Access |
|---|---|---|---|---|---|---|---|
| | | 3 | ILS0 | | Initial lane synchronization status for Lane 0 (ignore this output when NO_ILAS = 1). | 0x0 | R |
| | | | | 0 | Synchronization lost. | | |
| | | | | 1 | Synchronization achieved. | | |
| | | 2 | CKS0 | | Computed checksum status for Lane 0 (ignore this output when NO_ILAS = 1). | 0x0 | R |
| | | | | 0 | Checksum is incorrect. | | |
| | | | | 1 | Checksum is correct. | | |
| | | 1 | FS0 | | Frame sync status for Lane 0 (ignore this output when NO_ILAS = 1). | 0x0 | R |
| | | | | 0 | Synchronization lost. | | |
| | | | | 1 | Synchronization achieved. | | |
| | | 0 | CGS0 | | Code group sync status for Lane 0. | 0x0 | R |
| | | | | 0 | Synchronization lost. | | |
| | | | | 1 | Synchronization achieved. | | |
| 0x4B1 | LINK_STATUS1 | 7 | BDE1 | | Bad disparity errors status for Lane 1. | 0x0 | R |
| | | | | 0 | Error count < ETH[7:0] value. | | |
| | | | | 1 | Error count ≥ ETH[7:0] value. | | |
| | | 6 | NIT1 | | Not in table errors status for Lane 1. | 0x0 | R |
| | | | | 0 | Error count < ETH[7:0] value. | | |
| | | | | 1 | Error count ≥ ETH[7:0] value. | | |
| | | 5 | UEK1 | | Unexpected K character errors status for Lane 1. | 0x0 | R |
| | | | | 0 | Error count < ETH[7:0] value. | | |
| | | | | 1 | Error count ≥ ETH[7:0] value. | | |
| | | 4 | ILD1 | | Interlane deskew status for Lane 1 (ignore this output when NO_ILAS = 1). | 0x0 | R |
| | | | | 0 | Deskew failed. | | |
| | | | | 1 | Deskew achieved. | | |
| | | 3 | ILS1 | | Initial lane synchronization status for Lane 1 (ignore this output when NO_ILAS = 1). | 0x0 | R |
| | | | | 0 | Synchronization lost. | | |
| | | | | 1 | Synchronization achieved. | | |
| | | 2 | CKS1 | | Computed checksum status for Lane 1 (ignore this output when NO_ILAS = 1). | 0x0 | R |
| | | | | 0 | Checksum is incorrect. | | |
| | | | | 1 | Checksum is correct. | | |
| | | 1 | FS1 | | Frame sync status for Lane 1 (ignore this output when NO_ILAS = 1). | 0x0 | R |
| | | | | 0 | Synchronization lost. | | |
| | | | | 1 | Synchronization achieved. | | |
| | | 0 | CGS1 | | Code group sync status for Lane 1. | 0x0 | R |
| | | | | 0 | Synchronization lost. | | |
| | | | | 1 | Synchronization achieved. | | |
| 0x4B2 | LINK_STATUS2 | 7 | BDE2 | | Bad disparity errors status for Lane 2. | 0x0 | R |
| | | | | 0 | Error count < ETH[7:0] value. | | |
| | | | | 1 | Error count ≥ ETH[7:0] value. | | |

| Hex. Addr. | Name | Bits | Bit Name | Settings | Description | Reset | Access |
|---|---|---|---|---|---|---|---|
| | | 6 | NIT2 | | Not in table errors status for Lane 2. | 0x0 | R |
| | | | | 0 | Error count < ETH[7:0] value. | | |
| | | | | 1 | Error count ≥ ETH[7:0] value. | | |
| | | 5 | UEK2 | | Unexpected K character errors status for Lane 2. | 0x0 | R |
| | | | | 0 | Error count < ETH[7:0] value. | | |
| | | | | 1 | Error count ≥ ETH[7:0] value. | | |
| | | 4 | ILD2 | | Interlane deskew status for Lane 2 (ignore this output when NO_ILAS = 1). | 0x0 | R |
| | | | | 0 | Deskew failed. | | |
| | | | | 1 | Deskew achieved. | | |
| | | 3 | ILS2 | | Initial lane synchronization status for Lane 2 (ignore this output when NO_ILAS = 1). | 0x0 | R |
| | | | | 0 | Synchronization lost. | | |
| | | | | 1 | Synchronization achieved. | | |
| | | 2 | CKS2 | | Computed checksum status for Lane 2 (ignore this output when NO_ILAS = 1). | 0x0 | R |
| | | | | 0 | Checksum is incorrect. | | |
| | | | | 1 | Checksum is correct. | | |
| | | 1 | FS2 | | Frame sync status for Lane 2 (ignore this output when NO_ILAS = 1). | 0x0 | R |
| | | | | 0 | Synchronization lost. | | |
| | | | | 1 | Synchronization achieved. | | |
| | | 0 | CGS2 | | Code group sync status for Lane 2. | 0x0 | R |
| | | | | 0 | Synchronization lost. | | |
| | | | | 1 | Synchronization achieved. | | |
| 0x4B3 | LINK_STATUS3 | 7 | BDE3 | | Bad disparity errors status for Lane 3. | 0x0 | R |
| | | | | 0 | Error count < ETH[7:0] value. | | |
| | | | | 1 | Error count ≥ ETH[7:0] value. | | |
| | | 6 | NIT3 | | Not in table errors status for Lane 3. | 0x0 | R |
| | | | | 0 | Error count < ETH[7:0] value. | | |
| | | | | 1 | Error count ≥ ETH[7:0] value. | | |
| | | 5 | UEK3 | | Unexpected K character errors status for Lane 3. | 0x0 | R |
| | | | | 0 | Error count < ETH[7:0] value. | | |
| | | | | 1 | Error count ≥ ETH[7:0] value. | | |
| | | 4 | ILD3 | | Interlane deskew status for Lane 3 (ignore this output when NO_ILAS = 1). | 0x0 | R |
| | | | | 0 | Deskew failed. | | |
| | | | | 1 | Deskew achieved. | | |
| | | 3 | ILS3 | | Initial lane synchronization status for Lane 3 (ignore this output when NO_ILAS = 1). | 0x0 | R |
| | | | | 0 | Synchronization lost. | | |
| | | | | 1 | Synchronization achieved. | | |
| | | 2 | CKS3 | | Computed checksum status for Lane 3 (ignore this output when NO_ILAS = 1). | 0x0 | R |
| | | | | 0 | Checksum is incorrect. | | |
| | | | | 1 | Checksum is correct. | | |

| Hex. Addr. | Name | Bits | Bit Name | Settings | Description | Reset | Access |
|---|---|---|---|---|---|---|---|
| | | 1 | FS3 | | Frame sync status for Lane 3 (ignore this output when NO_ILAS = 1). | 0x0 | R |
| | | | | 0 | Synchronization lost. | | |
| | | | | 1 | Synchronization achieved. | | |
| | | 0 | CGS3 | | Code group sync status for Lane 3. | 0x0 | R |
| | | | | 0 | Synchronization lost. | | |
| | | | | 1 | Synchronization achieved. | | |
| 0x4B4 | LINK_STATUS4 | 7 | BDE4 | | Bad disparity errors status for Lane 4. | 0x0 | R |
| | | | | 0 | Error count < ETH[7:0] value. | | |
| | | | | 1 | Error count ≥ ETH[7:0] value. | | |
| | | 6 | NIT4 | | Not in table errors status for Lane 4. | 0x0 | R |
| | | | | 0 | Error count < ETH[7:0] value. | | |
| | | | | 1 | Error count ≥ ETH[7:0] value. | | |
| | | 5 | UEK4 | | Unexpected K character errors status for Lane 4. | 0x0 | R |
| | | | | 0 | Error count < ETH[7:0] value. | | |
| | | | | 1 | Error count ≥ ETH[7:0] value. | | |
| | | 4 | ILD4 | | Interlane deskew status for Lane 4 (ignore this output when NO_ILAS = 1). | 0x0 | R |
| | | | | 0 | Deskew failed. | | |
| | | | | 1 | Deskew achieved. | | |
| | | 3 | ILS4 | | Initial lane synchronization status for Lane 4 (ignore this output when NO_ILAS = 1). | 0x0 | R |
| | | | | 0 | Synchronization lost. | | |
| | | | | 1 | Synchronization achieved. | | |
| | | 2 | CKS4 | | Computed checksum status for Lane 4 (ignore this output when NO_ILAS = 1). | 0x0 | R |
| | | | | 0 | Checksum is incorrect. | | |
| | | | | 1 | Checksum is correct. | | |
| | | 1 | FS4 | | Frame sync status for Lane 4 (ignore this output when NO_ILAS = 1). | 0x0 | R |
| | | | | 0 | Synchronization lost. | | |
| | | | | 1 | Synchronization achieved. | | |
| | | 0 | CGS4 | | Code group sync status for Lane 4. | 0x0 | R |
| | | | | 0 | Synchronization lost. | | |
| | | | | 1 | Synchronization achieved. | | |
| 0x4B5 | LINK_STATUS5 | 7 | BDE5 | | Bad disparity errors status for Lane 5. | 0x0 | R |
| | | | | 0 | Error count < ETH[7:0] value. | | |
| | | | | 1 | Error count ≥ ETH[7:0] value. | | |
| | | 6 | NIT5 | | Not in table errors status for Lane 5. | 0x0 | R |
| | | | | 0 | Error count < ETH[7:0] value. | | |
| | | | | 1 | Error count ≥ ETH[7:0] value. | | |
| | | 5 | UEK5 | | Unexpected K character errors status for Lane 5. | 0x0 | R |
| | | | | 0 | Error count < ETH[7:0] value. | | |
| | | | | 1 | Error count ≥ ETH[7:0] value. | | |

| Hex. Addr. | Name | Bits | Bit Name | Settings | Description | Reset | Access |
|---|---|---|---|---|---|---|---|
| | | 4 | ILD5 | | Interlane deskew status for Lane 5 (ignore this output when NO_ILAS = 1). | 0x0 | R |
| | | | | 0 | Deskew failed. | | |
| | | | | 1 | Deskew achieved. | | |
| | | 3 | ILS5 | | Initial lane synchronization status for Lane 5 (ignore this output when NO_ILAS = 1). | 0x0 | R |
| | | | | 0 | Synchronization lost. | | |
| | | | | 1 | Synchronization achieved. | | |
| | | 2 | CKS5 | | Computed checksum status for Lane 5 (ignore this output when NO_ILAS = 1). | 0x0 | R |
| | | | | 0 | Checksum is incorrect. | | |
| | | | | 1 | Checksum is correct. | | |
| | | 1 | FS5 | | Frame sync status for Lane 5 (ignore this output when NO_ILAS = 1). | 0x0 | R |
| | | | | 0 | Synchronization lost. | | |
| | | | | 1 | Synchronization achieved. | | |
| | | 0 | CGS5 | | Code group sync status for Lane 5. | 0x0 | R |
| | | | | 0 | Synchronization lost. | | |
| | | | | 1 | Synchronization achieved. | | |
| 0x4B6 | LINK_STATUS6 | 7 | BDE6 | | Bad disparity errors status for Lane 6. | 0x0 | R |
| | | | | 0 | Error count < ETH[7:0] value. | | |
| | | | | 1 | Error count ≥ ETH[7:0] value. | | |
| | | 6 | NIT6 | | Not in table errors status for Lane 6. | 0x0 | R |
| | | | | 0 | Error count < ETH[7:0] value. | | |
| | | | | 1 | Error count ≥ ETH[7:0] value. | | |
| | | 5 | UEK6 | | Unexpected K character errors status for Lane 6. | 0x0 | R |
| | | | | 0 | Error count < ETH[7:0] value. | | |
| | | | | 1 | Error count ≥ ETH[7:0] value. | | |
| | | 4 | ILD6 | | Interlane deskew status for Lane 6 (ignore this output when NO_ILAS = 1). | 0x0 | R |
| | | | | 0 | Deskew failed. | | |
| | | | | 1 | Deskew achieved. | | |
| | | 3 | ILS6 | | Initial lane synchronization status for Lane 6 (ignore this output when NO_ILAS = 1). | 0x0 | R |
| | | | | 0 | Synchronization lost. | | |
| | | | | 1 | Synchronization achieved. | | |
| | | 2 | CKS6 | | Computed checksum status for Lane 6 (ignore this output when NO_ILAS = 1). | 0x0 | R |
| | | | | 0 | Checksum is incorrect. | | |
| | | | | 1 | Checksum is correct. | | |
| | | 1 | FS6 | | Frame sync status for Lane 6 (ignore this output when NO_ILAS = 1). | 0x0 | R |
| | | | | 0 | Synchronization lost. | | |
| | | | | 1 | Synchronization achieved. | | |
| | | 0 | CGS6 | | Code group sync status for Lane 6. | 0x0 | R |
| | | | | 0 | Synchronization lost. | | |
| | | | | 1 | Synchronization achieved. | | |

| Hex. Addr. | Name | Bits | Bit Name | Settings | Description | Reset | Access |
|---|---|---|---|---|---|---|---|
| 0x4B7 | LINK_STATUS7 | 7 | BDE7 | | Bad disparity errors status for Lane 7. | 0x0 | R |
| | | | | 0 | Error count < ETH[7:0] value. | | |
| | | | | 1 | Error count ≥ ETH[7:0] value. | | |
| | | 6 | NIT7 | | Not in table errors status for Lane 7. | 0x0 | R |
| | | | | 0 | Error count < ETH[7:0] value. | | |
| | | | | 1 | Error count ≥ ETH[7:0] value. | | |
| | | 5 | UEK7 | | Unexpected K character errors status for Lane 7. | 0x0 | R |
| | | | | 0 | Error count < ETH[7:0] value. | | |
| | | | | 1 | Error count ≥ ETH[7:0] value. | | |
| | | 4 | ILD7 | | Interlane deskew status for Lane 7 (ignore this output when NO_ILAS = 1). | 0x0 | R |
| | | | | 0 | Deskew failed. | | |
| | | | | 1 | Deskew achieved. | | |
| | | 3 | ILS7 | | Initial lane synchronization status for Lane 7 (ignore this output when NO_ILAS = 1). | 0x0 | R |
| | | | | 0 | Synchronization lost. | | |
| | | | | 1 | Synchronization achieved. | | |
| | | 2 | CKS7 | | Computed checksum status for Lane 7 (ignore this output when NO_ILAS = 1). | 0x0 | R |
| | | | | 0 | Checksum is incorrect. | | |
| | | | | 1 | Checksum is correct. | | |
| | | 1 | FS7 | | Frame sync status for Lane 7 (ignore this output when NO_ILAS = 1). | 0x0 | R |
| | | | | 0 | Synchronization lost. | | |
| | | | | 1 | Synchronization achieved. | | |
| | | 0 | CGS7 | | Code group sync status for Lane 7. | 0x0 | R |
| | | | | 0 | Synchronization lost. | | |
| | | | | 1 | Synchronization achieved. | | |
| 0x4B8 | JESD_IRQ_ENABLEA | 7 | EN_BDE | | Bad disparity error counter. | 0x0 | R/W |
| | | 6 | EN_NIT | | Not in table error counter. | 0x0 | R/W |
| | | 5 | EN_UEK | | Unexpected K error counter. | 0x0 | R/W |
| | | 4 | EN_ILD | | Interlane deskew. | 0x0 | R/W |
| | | 3 | EN_ILS | | Initial lane sync. | 0x0 | R/W |
| | | 2 | EN_CKS | | Good checksum. This is an interrupt that compares two checksums: the checksum that the transmitter sent over the link during the ILAS, and the checksum that the receiver calculated from the ILAS data that the transmitter sent over the link. Note that the checksum IRQ never at any time looks at the checksum that is programmed over the SPI into Register 0x45D. The checksum IRQ only looks at the data sent by the transmitter, and never looks at any data programmed via the SPI. | 0x0 | R/W |
| | | 1 | EN_FS | | Frame sync. | 0x0 | R/W |
| | | 0 | EN_CGS | | Code group sync. | 0x0 | R/W |

| Hex. Addr. | Name | Bits | Bit Name | Settings | Description | Reset | Access |
|---|---|---|---|---|---|---|---|
| 0x4B9 | JESD_IRQ_ENABLEB | [7:1] | RESERVED | | Reserved. | 0x0 | R |
| | | 0 | EN_ILAS | | Configuration mismatch (checked for Lane 0 only). The ILAS IRQ compares the two sets of ILAS data that the receiver has: the ILAS data sent over the JESD204B link by the transmitter, and the ILAS data programmed into the receiver via the SPI (Register 0x450 to Register 0x45D). If the data differs, the IRQ is triggered. Note that all of the ILAS data (including the checksum) is compared. | 0x0 | R/W |
| 0x4BA | JESD_IRQ_STATUSA | 7 | IRQ_BDE | | Bad disparity error counter. | 0x0 | R/W |
| | | 6 | IRQ_NIT | | Not in table error counter. | 0x0 | R/W |
| | | 5 | IRQ_UEK | | Unexpected K error counter. | 0x0 | R/W |
| | | 4 | IRQ_ILD | | Interlane deskew. | 0x0 | R/W |
| | | 3 | IRQ_ILS | | Initial lane sync. | 0x0 | R/W |
| | | 2 | IRQ_CKS | | Good checksum. | 0x0 | R/W |
| | | 1 | IRQ_FS | | Frame sync. | 0x0 | R/W |
| | | 0 | IRQ_CGS | | Code group sync. | 0x0 | R/W |
| 0x4BB | JESD_IRQ_STATUSB | [7:1] | RESERVED | | Reserved. | 0x0 | R |
| | | 0 | IRQ_ILAS | | Configuration mismatch (checked for Lane 0 only). | 0x0 | R/W |
| 0x800 | HOPF_CTRL | [7:6] | HOPF_MODE | | Frequency switch mode. | 0x0 | R/W |
| | | | | 00 | Phase continuous switch. Changes frequency tuning word, and the phase accumulator continues to accumulate to the new FTW. | | |
| | | | | 01 | Phase discontinuous switch. Changes the frequency tuning word and resets the phase accumulator. | | |
| | | | | 10 | Reserved. | | |
| | | [5:0] | RESERVED | | Reserved. | 0x0 | R |

# OUTLINE DIMENSIONS

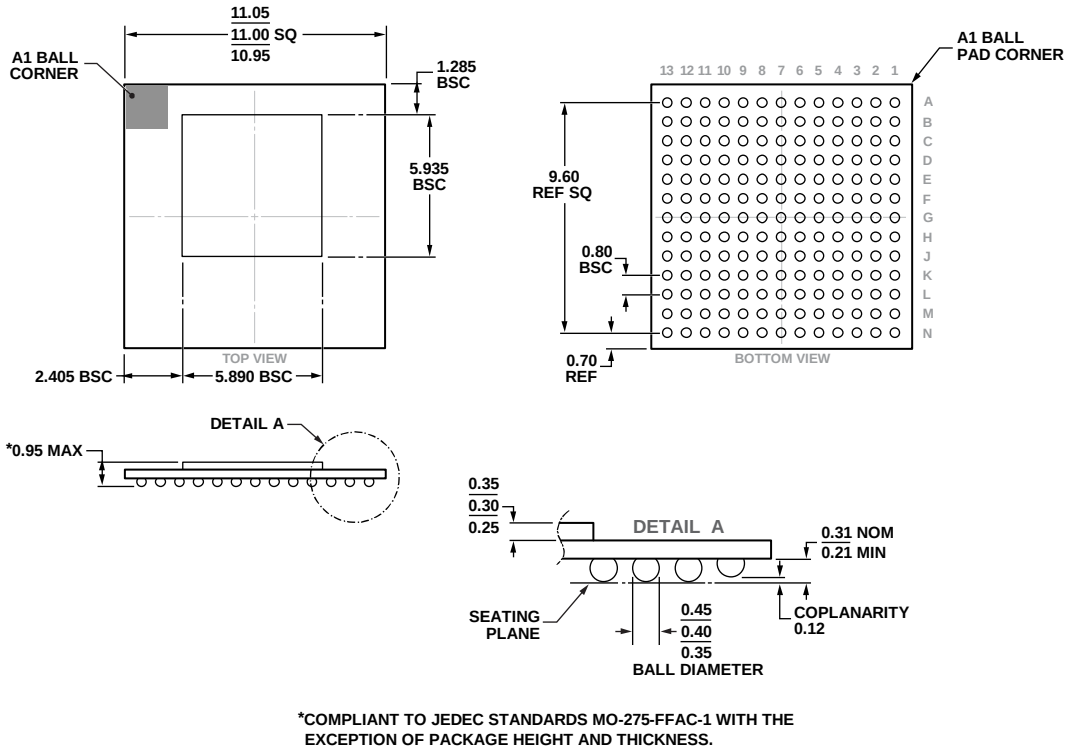


*Figure 141. 169-Ball Chip Scale Package Ball Grid Array [CSP_BGA]*
*(BC-169-2)*
*Dimensions shown in millimeters*

## ORDERING GUIDE

| Model[1] | Temperature Range | Package Description | Package Option |
|---|---|---|---|
| AD9163BBCZ | −40°C to +85°C | 169-Ball Chip Scale Package Ball Grid Array [CSP_BGA] | BC-169-2 |
| AD9163BBCZRL | −40°C to +85°C | 169-Ball Chip Scale Package Ball Grid Array [CSP_BGA] | BC-169-2 |
| AD9163-FMCC-EBZ | | Evaluation Board | |

[1] Z = RoHS Compliant Part.

©2016–2017 Analog Devices, Inc. All rights reserved. Trademarks and registered trademarks are the property of their respective owners.
D14415-0-7/17(C)