# THE DATASHEET OF

# AD9171BBPZ

Data Sheet

# AD9171

## Dual, 16-Bit, 6.2 GSPS RF DAC with Single Channelizer

## FEATURES

- Supports single-band wireless applications
  - 1 complex data input channel per RF DAC
  - 516 MSPS maximum complex input data rate per input channel
  - 1 independent NCO per input channel
- Proprietary, low spurious and distortion design
  - 2-tone IMD = −83 dBc at 1.8 GHz, −7 dBFS/tone RF output
  - SFDR < −80 dBc at 1.8 GHz, −7 dBFS RF output
- Flexible 8-lane, 15.4 Gbps JESD204B interface
  - Supports single-band use cases
  - Supports 12-bit high density mode for increased data throughput
- Multiple chip synchronization
  - Supports JESD204B Subclass 1
- Selectable interpolation filter for a complete set of input data rates
  - 2×, 3×, 4×, and 6× configurable data channel interpolation
  - 6× and 8× configurable final interpolation
- Final 48-bit NCO that operates at the DAC rate to support frequency synthesis up to 3.1 GHz
- Transmit enable function allows extra power saving and downstream circuitry protection
- High performance, low noise PLL clock multiplier
  - Supports 6.2 GSPS DAC update rate
  - Observation ADC clock driver with selectable divide ratios
- Low power
  - 1.45 W at 6 GSPS, single-channel mode
- 10 mm × 10 mm, 144-ball BGA_ED with metal enhanced thermal lid, 0.80 mm pitch

## APPLICATIONS

- Wireless communications infrastructure
  - Single-band base station radios
- Instrumentation, automatic test equipment (ATE)

## GENERAL DESCRIPTION

The AD9171 is a high performance, dual, 16-bit digital-to-analog converter (DAC) that supports DAC sample rates to 6.2 GSPS. The device features an 8-lane, 15.4 Gbps JESD204B data input port, a high performance, on-chip DAC clock multiplier, and digital signal processing capabilities targeted at single-band direct to radio frequency (RF) wireless applications.

The AD9171 features one complex data input channels per RF DAC. Each data input channel includes a configurable gain stage, an interpolation filter, and a channel numerically controlled oscillator (NCO) for flexible, frequency planning. The device supports up to a 516 MSPS complex data rate per input channel.

The AD9171 is available in a 144-ball BGA_ED package.

## PRODUCT HIGHLIGHTS

1. Supports one complex data input channel per RF DAC at a maximum complex input data rate of 513 MSPS with 12-bit resolution and 516 MSPS with 16-bit resolution options. There is one independent NCO per input channel.
2. Low power dual converter decreases the amount of power consumption needed in high bandwidth and multichannel applications.

Rev. C

DOCUMENT FEEDBACK
TECHNICAL SUPPORT

Information furnished by Analog Devices is believed to be accurate and reliable "as is". 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.

## TABLE OF CONTENTS


Features ..... 1
Applications ..... 1
General Description ..... 1
Product Highlights ..... 1
Functional Block Diagram ..... 3
Specifications ..... 4
DC Specifications ..... 4
Digital Specifications ..... 4
Power Supply DC Specifications ..... 5
Serial Port and CMOS Pin Specifications ..... 6
Digital Input Data Timing Specifications ..... 7
JESD204B Interface Electrical and Speed Specifications ..... 7
Input Data Rates and Signal Bandwidth Specifications ..... 8
AC Specifications ..... 8
Absolute Maximum Ratings ..... 11
Reflow Profile ..... 11
Thermal Characteristics ..... 11
ESD Caution ..... 11
Pin Configuration and Function Descriptions ..... 12
Typical Performance Characteristics ..... 15
Terminology ..... 18
Theory of Operation ..... 20
Serial Port Operation ..... 21
Data Format ..... 21
Serial Port Pin Descriptions ..... 21
Serial Port Options ..... 21
JESD204B Serial Data Interface ..... 23
JESD204B Overview ..... 23
Physical Layer ..... 25
Data Link Layer ..... 27
Syncing LMFC Signals ..... 29
Transport Layer ..... 35
JESD204B Test Modes ..... 36
JESD204B Error Monitoring ..... 38
Digital Datapath ..... 40
Total Datapath Interpolation ..... 40
Channel Digital Datapath ..... 41
Main Digital Datapath ..... 44
Interrupt Request Operation ..... 50
Interrupt Service Routine ..... 50
Applications Information ..... 51
Hardware Considerations ..... 51
Analog Interface Considerations ..... 54
DAC Input Clock Configurations ..... 54
Clock Output Driver ..... 56
Analog Outputs ..... 56
Start-Up Sequence ..... 57
Register Summary ..... 64
Register Details ..... 75
Outline Dimensions ..... 134
Ordering Guide ..... 134
Evaluation Boards ..... 134


## REVISION HISTORY

**6/2022—Rev. B to Rev. C**

Deleted Figure 12 and Figure 13; Renumbered Sequentially ..... 15
Changes to Table 44 ..... 57
Changes to Table 47 and Table 48 ..... 58
Changes to Table 53 ..... 63
Change to Table 54 ..... 64
Changes to Table 55 ..... 75

## FUNCTIONAL BLOCK DIAGRAM

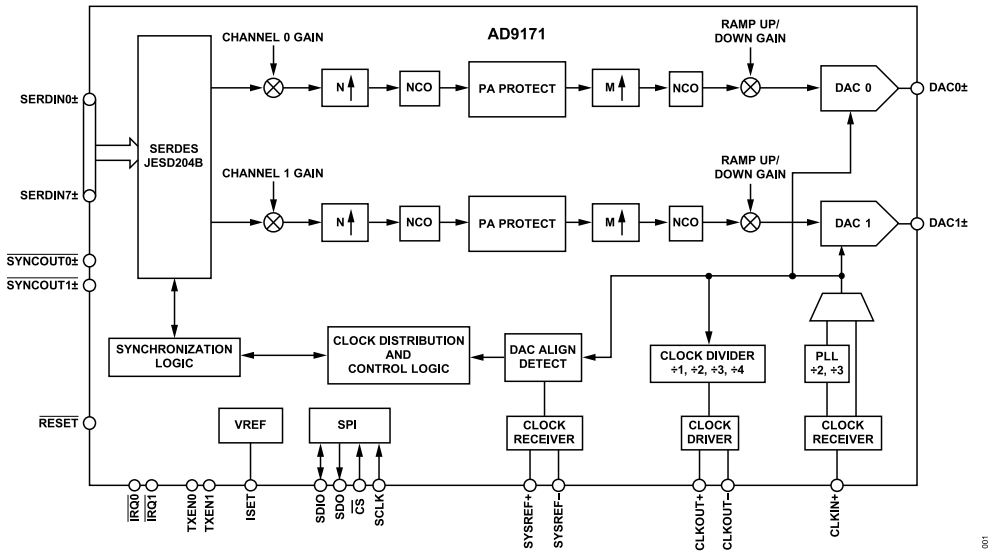


*Figure 1. Functional Block Diagram*

# SPECIFICATIONS

## DC SPECIFICATIONS

AVDD1.0 = 1.0 V, AVDD1.8 = 1.8 V, DVDD1.0 = 1.0 V, DVDD1.8 = 1.8 V, SVDD1.0 = 1.0 V, and DAC output full-scale current ($I_{OUTFS}$) = 20 mA, unless otherwise noted. For the minimum and maximum values, $T_J$ = −40°C to +118°C. For the typical values, $T_A$ = 25°C, which corresponds to $T_J$ = 51°C.

***Table 1.***

| Parameter | Test Conditions/Comments | Min | Typ | Max | Unit |
|---|---|---|---|---|---|
| RESOLUTION | | 16 | | | Bit |
| ACCURACY | | | | | |
| Integral Nonlinearity (INL) | | | ±7 | | LSB |
| Differential Nonlinearity (DNL) | | | ±7 | | LSB |
| ANALOG OUTPUTS (DAC0+, DAC0−, DAC1+, DAC1−) | | | | | |
| Gain Error (with Internal ISET Reference) | | | ±15 | | % |
| Full-Scale Output Current | | | | | |
| Minimum | $R_{SET}$ = 5 kΩ | 14.2 | 16 | 17.8 | mA |
| Maximum | $R_{SET}$ = 5 kΩ | 23.6 | 26 | 28.8 | mA |
| Common-Mode Voltage | | | 0 | | V |
| Differential Impedance | | | 100 | | Ω |
| DAC DEVICE CLOCK INPUT (CLKIN+, CLKIN−) | | | | | |
| Differential Input Power | $R_{LOAD}$ = 100 Ω differential on-chip | | | | |
| Minimum | | | 0 | | dBm |
| Maximum | | | 6 | | dBm |
| Differential Input Impedance[1] | | | 100 | | Ω |
| Common-Mode Voltage | AC-coupled | | 0.5 | | V |
| CLOCK OUTPUT DRIVER (CLKOUT+, CLKOUT−) | | | | | |
| Differential Output Power | | | | | |
| Minimum | | | −9 | | dBm |
| Maximum | | | 0 | | dBm |
| Differential Output Impedance | | | 100 | | Ω |
| Common-Mode Voltage | AC-coupled | | 0.5 | | V |
| Output Frequency | | 727.5 | | 3000 | MHz |
| TEMPERATURE DRIFT | | | | | |
| Gain | | | 10 | | ppm/°C |
| REFERENCE | | | | | |
| Internal Reference Voltage | | | 0.495 | | V |
| ANALOG SUPPLY VOLTAGES | | | | | |
| AVDD1.0 | | 0.95 | 1.0 | 1.05 | V |
| AVDD1.8 | | 1.71 | 1.8 | 1.89 | V |
| DIGITAL SUPPLY VOLTAGES | | | | | |
| DVDD1.0 | | 0.95 | 1.0 | 1.05 | V |
| DAVDD1.0 | | 0.95 | 1.0 | 1.05 | V |
| DVDD1.8 | | 1.71 | 1.8 | 1.89 | V |
| SERDES SUPPLY VOLTAGES | | | | | |
| SVDD1.0 | | 0.95 | 1.0 | 1.05 | V |

[1] See the DAC Input Clock Configurations section for more details.

## DIGITAL SPECIFICATIONS

AVDD1.0 = 1.0 V, AVDD1.8 = 1.8 V, DVDD1.0 = 1.0 V, DVDD1.8 = 1.8 V, SVDD1.0 = 1.0 V, and DAC $I_{OUTFS}$ = 20 mA, unless otherwise noted. For the minimum and maximum values, $T_J$ = −40°C to +118°C. For the typical values, $T_A$ = +25°C, which corresponds to $T_J$ = 51°C.

## SPECIFICATIONS

*Table 2.*

| Parameter | Test Conditions/Comments | Min | Typ | Max | Unit |
|---|---|---|---|---|---|
| DAC UPDATE RATE | | | | | |
| Minimum | | | | 2.91 | GSPS |
| Maximum[1] | | 6.2 | | | GSPS |
| Adjusted[2] | | 516 | | | MSPS |
| DAC PHASE-LOCKED LOOP (PLL) VOLTAGE CONTROLLED OSCILLATOR (VCO) FREQUENCY RANGES | | | | | |
| VCO Output Divide by 2 | | 4.37 | | 6.2 | GSPS |
| VCO Output Divide by 3 | | 2.91 | | 4.14 | GSPS |
| PHASE FREQUENCY DETECT INPUT FREQUENCY RANGES | | | | | |
| 9.96 GHz ≤ VCO Frequency ≤ 10.87 GHz | | 25 | | 225 | MHz |
| VCO Frequency < 9.96 GHz or VCO Frequency > 10.87 GHz | | 25 | | 770 | MHz |
| DAC DEVICE CLOCK INPUT (CLKIN+, CLKIN−) FREQUENCY RANGES | | | | | |
| PLL Off | | 2.91 | | 6.2 | GHz |
| PLL On | M divider set to divide by 1 | 25 | | 770 | MHz |
| | M divider set to divide by 2 | 50 | | 1540 | MHz |
| | M divider set to divide by 3 | 75 | | 2310 | MHz |
| | M divider set to divide by 4 | 100 | | 3080 | MHz |

[1] The maximum DAC update rate varies depending on the selected JESD204B mode and the lane rate for the given configuration used. The maximum DAC rate according to lane rate and voltage supply levels is listed in Table 3.

[2] The adjusted DAC update rate is calculated as $f_{DAC}$, divided by the minimum required interpolation factor for a given mode or the maximum channel data rate for a given mode. Different modes have different maximum DAC update rates, minimum interpolation factors, and maximum channel data rates, as shown in Table 12.

### POWER SUPPLY DC SPECIFICATIONS

AVDD1.0 = 1.0 V, AVDD1.8 = 1.8 V, DVDD1.0 = 1.0 V, DVDD1.8 = 1.8 V, SVDD1.0 = 1.0 V, and DAC $I_{OUTFS}$ = 20 mA, unless otherwise noted. For the minimum and maximum values, $T_J$ = −40°C to +118°C. For the typical values, $T_A$ = 25°C, which corresponds to $T_J$ = 51°C.

*Table 3.*

| Parameter | Test Conditions/Comments | Min | Typ | Max | Unit |
|---|---|---|---|---|---|
| DUAL-LINK MODE, Mode 0 (L = 1, M = 2, NP = 16, N = 16) | 5.89824 GSPS DAC rate, 184.32 MHz PLL reference clock, 16× total interpolation (2×, 8×), 40 MHz tone at −3 dBFS, channel NCO disabled, main NCO = 1.8425 GHz, $\overline{\text{SYNCOUTx±}}$ in LVDS mode | | | | |
| AVDD1.0 | All supply levels set to nominal values | | 400 | 670 | mA |
| | All supplies at 5% tolerance | | 425 | 745 | mA |
| AVDD1.8 | | | 110 | 130 | mA |
| DVDD1.0 | Combined current consumption with the DAVDD1.0 supply | | | | |
| | All supply levels set to nominal values | | 625 | 960 | mA |
| | All supplies at 5% tolerance | | 670 | 1070 | mA |
| DVDD1.8 | | | 35 | 50 | mA |
| SVDD1.0 | | | 175 | 340 | mA |
| Total Power Dissipation | | | 1.45 | 2.15 | W |
| DUAL-LINK, MODE 3 (NCO ONLY, SINGLE-CHANNEL MODE, NO SERDES) | 6 GSPS DAC rate, 300 MHz PLL reference clock, 8× total interpolation (1×, 8×), no input tone (dc internal level = 0x50FF), channel NCO = 40 MHz, main NCO = 1.8425 GHz | | | | |
| AVDD1.0 | | | | | |
| | All supply levels set to nominal values | | 410 | 660 | mA |
| | All supplies at 5% tolerance | | 435 | 750 | mA |
| AVDD1.8 | | | 110 | 130 | mA |
| DVDD1.0 | Combined current consumption with the DAVDD1.0 supply | | | | |

## SPECIFICATIONS

*Table 3.*

| Parameter | Test Conditions/Comments | Min | Typ | Max | Unit |
|---|---|---|---|---|---|
| | All supply levels set to nominal values | | 500 | 780 | mA |
| | All supplies at 5% tolerance | | 515 | 950 | mA |
| DVDD1.8 | | | 0.3 | 1 | mA |
| SVDD1.0 | All supply levels set to nominal values | | 5 | 100 | mA |
| | All supplies at 5% tolerance | | 3 | 120 | mA |
| Total Power Dissipation | | | 1.1 | 1.671 | W |

## SERIAL PORT AND CMOS PIN SPECIFICATIONS

AVDD1.0 = 1.0 V, AVDD1.8 = 1.8 V, DVDD1.0 = 1.0 V, DVDD1.8 = 1.8 V, SVDD1.0 = 1.0 V, and DAC $I_{OUTFS}$ = 20 mA, unless otherwise noted. For the minimum and maximum values, $T_J$ = −40°C to +118°C. For the typical values, $T_A$ = 25°C, which corresponds to $T_J$ = 51°C.

*Table 4.*

| Parameter | Symbol | Test Comments/Conditions | Min | Typ | Max | Unit |
|---|---|---|---|---|---|---|
| WRITE OPERATION | | See Figure 24 | | | | |
| Maximum SCLK Clock Rate | $f_{SCLK}$, $1/t_{SCLK}$ | | 80 | | | MHz |
| SCLK Clock High | $t_{PWH}$ | SCLK = 20 MHz | 5.03 | | | ns |
| SCLK Clock Low | $t_{PWL}$ | SCLK = 20 MHz | 1.6 | | | ns |
| SDIO to SCLK Setup Time | $t_{DS}$ | | 1.154 | | | ns |
| SCLK to SDIO Hold Time | $t_{DH}$ | | 0.577 | | | ns |
| $\overline{CS}$ to SCLK Setup Time | $t_S$ | | 1.036 | | | ns |
| SCLK to $\overline{CS}$ Hold Time | $t_H$ | | −5.3 | | | ps |
| READ OPERATION | | See Figure 23 | | | | |
| SCLK Clock Rate | $f_{SCLK}$, $1/t_{SCLK}$ | | | | 48.58 | MHz |
| SCLK Clock High | $t_{PWH}$ | | 5.03 | | | ns |
| SCLK Clock Low | $t_{PWL}$ | | 1.6 | | | ns |
| SDIO to SCLK Setup Time | $t_{DS}$ | | 1.158 | | | ns |
| SCLK to SDIO Hold Time | $t_{DH}$ | | 0.537 | | | ns |
| $\overline{CS}$ to SCLK Setup Time | $t_S$ | | 1.036 | | | ns |
| SCLK to SDIO Data Valid Time | $t_{DV}$ | | 9.6 | | | ns |
| SCLK to SDO Data Valid Time | $t_{DV}$ | | 13.7 | | | ns |
| $\overline{CS}$ to SDIO Output Valid to High-Z | | Not shown in Figure 23 or Figure 24 | 5.4 | | | ns |
| $\overline{CS}$ to SDO Output Valid to High-Z | | Not shown in Figure 23 or Figure 24 | 9.59 | | | ns |
| INPUTS (SDIO, SCLK, $\overline{CS}$, $\overline{RESET}$, TXEN0, and TXEN1) | | | | | | |
| Voltage Input | | | | | | |
| High | $V_{IH}$ | | 1.48 | | | V |
| Low | $V_{IL}$ | | | | 0.425 | V |
| Current Input | | | | | | |
| High | $I_{IH}$ | | | | ±100 | nA |
| Low | $I_{IL}$ | | ±100 | | | nA |
| OUTPUTS (SDIO, SDO) | | | | | | |
| Voltage Output | | | | | | |
| High | $V_{OH}$ | | | | | |
| 0 mA load | | | 1.69 | | | V |
| 4 mA load | | | 1.52 | | | V |
| Low | $V_{OL}$ | | | | | |
| 0 mA load | | | | | 0.045 | V |
| 4 mA load | | | | | 0.175 | V |
| Current Output | | | | | | |

## SPECIFICATIONS

*Table 4.*

| Parameter | Symbol | Test Comments/Conditions | Min | Typ | Max | Unit |
|---|---|---|---|---|---|---|
| High | $I_{OH}$ | | | 4 | | mA |
| Low | $I_{OL}$ | | | 4 | | mA |
| INTERRUPT OUTPUTS ($\overline{IRQ0}$, $\overline{IRQ1}$) | | | | | | |
| Voltage Output | | | | | | |
| High | $V_{OH}$ | | 1.71 | | | V |
| Low | $V_{OL}$ | | | | 0.075 | V |

## DIGITAL INPUT DATA TIMING SPECIFICATIONS

AVDD1.0 = 1.0 V, AVDD1.8 = 1.8 V, DVDD1.0 = 1.0 V, DVDD1.8 = 1.8 V, SVDD1.0 = 1.0 V, and DAC $I_{OUTFS}$ = 20 mA, unless otherwise noted. For the minimum and maximum values, $T_J$ = −40°C to +118°C. For the typical values, $T_A$ = 25°C, which corresponds to $T_J$ = 51°C.

*Table 5.*

| Parameter | Test Conditions/Comments | Min | Typ | Max | Unit |
|---|---|---|---|---|---|
| LATENCY[1] | | | | | |
| Channel Interpolation Factor, Main Datapath Interpolation Factor | LMFC_VAR_x = 12, LMFC_DELAY_x = 12, unless otherwise noted | | | | |
| 2×, 6× | JESD204B Mode 3 | | 1970 | | DAC clock cycle |
| | JESD204B Mode 5 | | 1770 | | DAC clock cycle |
| 2×, 8× | JESD204B Mode 0 | | 2020 | | DAC clock cycle |
| | JESD204B Mode 3 | | 2500 | | DAC clock cycle |
| 3×, 6× | JESD204B Mode 3 | | 2880 | | DAC clock cycle |
| | JESD204B Mode 5 | | 2630 | | DAC clock cycle |
| 3×, 8× | JESD204B Mode 3 | | 3310 | | DAC clock cycle |
| | JESD204B Mode 5 | | 2980 | | DAC clock cycle |
| 4×, 6× | JESD204B Mode 0 | | 2410 | | DAC clock cycle |
| 4×, 8× | JESD204B Mode 0 | | 3090 | | DAC clock cycle |
| 6×, 6× | JESD204B Mode 0 | | 3190 | | DAC clock cycle |
| 6×, 8× | JESD204B Mode 0 | | 4130 | | DAC clock cycle |
| DETERMINISTIC LATENCY | | | | | |
| Fixed | | | | 13 | PCLK[2] |
| Variable | | | | 2 | PCLK cycles |
| SYSREF± TO LOCAL MULTIFRAME CLOCK (LMFC) DELAY | Indicates the relationship between the SYSREF± signal and the LMFC Clock Phase 0 | | 0 | | 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.

[2] PCLK is the internal processing clock for the AD9171 and equals the lane rate ÷ 40.

## JESD204B INTERFACE ELECTRICAL AND SPEED SPECIFICATIONS

AVDD1.0 = 1.0 V, AVDD1.8 = 1.8 V, DVDD1.0 = 1.0 V, DVDD1.8 = 1.8 V, SVDD1.0 = 1.0 V, and DAC $I_{OUTFS}$ = 20 mA, unless otherwise noted. For the minimum and maximum values, $T_J$ = −40°C to +118°C. For the typical values, $T_A$ = 25°C, which corresponds to $T_J$ = 51°C.

*Table 6.*

| Parameter | Symbol | Test Conditions/Comments | Min | Typ | Max | Unit |
|---|---|---|---|---|---|---|
| JESD204B SERIAL INTERFACE RATE (SERIAL LANE RATE) | | | 3 | | 15.4 | Gbps |
| JESD204B DATA INPUTS | | | | | | |
| Input Leakage Current | | $T_A$ = 25°C | | | | |
| Logic High | | Input level = 1.0 V ± 0.25 V | | 10 | | µA |
| Logic Low | | Input level = 0 V | | −4 | | µA |

## SPECIFICATIONS

*Table 6.*

| Parameter | Symbol | Test Conditions/Comments | Min | Typ | Max | Unit |
|---|---|---|---|---|---|---|
| Unit Interval | UI | | 333 | | 66.7 | ps |
| Common-Mode Voltage | $V_{RCM}$ | AC-coupled | −0.05 | | +1.1 | V |
| Differential Voltage | $R_V_{DIFF}$ | | 110 | | 1050 | mV |
| Differential Impedance | $Z_{RDIFF}$ | At dc | 80 | 100 | 120 | Ω |
| SYSREF± INPUT | | | | | | |
| Differential Impedance | | | | 100 | | Ω |
| DIFFERENTIAL OUTPUTS ($\overline{\text{SYNCOUT0±}}$, $\overline{\text{SYNCOUT1±}}$)[1] | | Driving 100 Ω differential load | | | | |
| Output Differential Voltage | $V_{OD}$ | | 320 | 390 | 460 | mV |
| Output Offset Voltage | $V_{OS}$ | | 1.08 | 1.12 | 1.15 | V |
| SINGLE-ENDED OUTPUTS ($\overline{\text{SYNCOUT0±}}$, $\overline{\text{SYNCOUT1±}}$) | | Driving 100 Ω differential load | | | | |
| Output Voltage | | | | | | |
| High | $V_{OH}$ | | 1.69 | | | V |
| Low | $V_{OL}$ | | | | 0.045 | V |
| Current Output | | | | | | |
| High | $I_{OH}$ | | | 0 | | mA |
| Low | $I_{OL}$ | | | 0 | | mA |

[1] IEEE Standard 1596.3 LVDS compatible.

## INPUT DATA RATES AND SIGNAL BANDWIDTH SPECIFICATIONS

AVDD1.0 = 1.0 V, AVDD1.8 = 1.8 V, DVDD1.0 = 1.0 V, DVDD1.8 = 1.8 V, SVDD1.0 = 1.0 V, and DAC output full-scale current ($I_{OUTFS}$) = 20 mA, unless otherwise noted. For the minimum and maximum values, $T_J$ = −40°C to +118°C. For the typical values, $T_A$ = 25°C, which corresponds to $T_J$ = 51°C.

*Table 7.*

| Parameter[1] | Test Conditions/Comments | Min | Typ | Max | Unit |
|---|---|---|---|---|---|
| INPUT DATA RATE PER INPUT CHANNEL | 1 complex channel enabled | | | 516 | MSPS |
| COMPLEX SIGNAL BANDWIDTH PER INPUT CHANNEL | 1 complex channel enabled (0.8 × $f_{DATA}$) | | | 413.34 | MHz |
| MAXIMUM NCO CLOCK RATE | | | | | |
| Channel NCO | | −258 | | +258 | MHz |
| Main NCO | | | | 6.2 | GHz |
| MAXIMUM NCO SHIFT FREQUENCY RANGE | | | | | |
| Channel NCO | | −258.34 | | +258.34 | MHz |
| Main NCO | | −3.1 | | +3.1 | GHz |
| MAXIMUM FREQUENCY SPACING ACROSS INPUT CHANNELS | Maximum NCO output frequency × 0.8 | | | 412.8 | MHz |

[1] Values listed for these parameters are the maximum possible when considering all JESD204B modes of operation. Some modes are more limiting, based on other parameters.

## AC SPECIFICATIONS

AVDD1.0 = 1.0 V, AVDD1.8 = 1.8 V, DVDD1.0 = 1.0 V, DVDD1.8 = 1.8 V, SVDD1.0 = 1.0 V, and DAC $I_{OUTFS}$ = 20 mA, unless otherwise noted. For the minimum and maximum, $T_J$ = −40°C to +118°C. For the typical values, $T_A$ = 25°C, which corresponds to $T_J$ = 51°C.

*Table 8.*

| Parameter | Test Conditions/Comments | Min | Typ | Max | Unit |
|---|---|---|---|---|---|
| SPURIOUS-FREE DYNAMIC RANGE (SFDR) | | | | | |
| Single Tone, $f_{DAC}$ = 6000 MSPS, Mode 0 (L = 1, M = 2) | −7 dBFS, shuffle enabled | | | | |

## SPECIFICATIONS

*Table 8.*

| Parameter | Test Conditions/Comments | Min | Typ | Max | Unit |
|---|---|---|---|---|---|
| $f_{OUT}$ = 100 MHz | | | −85 | | dBc |
| $f_{OUT}$ = 500 MHz | | | −85 | | dBc |
| $f_{OUT}$ = 950 MHz | | | −78 | | dBc |
| $f_{OUT}$ = 1840 MHz | | | −75 | | dBc |
| $f_{OUT}$ = 2650 MHz | | | −69 | | dBc |
| Single-Band Application—Band 3 (1805 MHz to 1880 MHz) | Mode 0, 2× to 8×, $f_{DAC}$ = 6000 MSPS, 368.64 MHz reference clock | | | | |
| SFDR Harmonics | −7 dBFS, shuffle enabled | | | | |
| In-Band | | | −82 | | dBc |
| Digital Predistortion (DPD) Band | DPD bandwidth = data rate × 0.8 | | −80 | | dBc |
| Second Harmonic | | | −82 | | dBc |
| Third Harmonic | | | −80 | | dBc |
| Fourth and Fifth Harmonic | | | −95 | | dBc |
| SFDR Nonharmonics | −7 dBFS, shuffle enabled | | | | |
| In-Band | | | −74 | | dBc |
| DPD Band | | | −74 | | dBc |
| ADJACENT CHANNEL LEAKAGE RATIO | | | | | |
| 4C-WCDMA | −1 dBFS digital backoff | | | | |
| $f_{DAC}$ = 6000 MSPS, Mode 0 (L = 1, M = 2) | $f_{OUT}$ = 1840 MHz | | −71 | | dBc |
| | $f_{OUT}$ = 2650 MHz | | −66 | | dBc |
| THIRD-ORDER INTERMODULATION DISTORTION (IMD) | Two-tone test, −7 dBFS/tone, 1 MHz spacing | | | | |
| $f_{DAC}$ = 6000 MSPS, Mode 0 (L = 1, M = 2) | $f_{OUT}$ = 1840 MHz | | −74 | | dBc |
| | $f_{OUT}$ = 2650 MHz | | −72 | | dBc |
| NOISE SPECTRAL DENSITY (NSD) | 0 dBFS, NSD measurement taken at 10% away from $f_{OUT}$, shuffle on | | | | |
| Single Tone, $f_{DAC}$ = 6000 MSPS | | | | | |
| $f_{OUT}$ = 200 MHz | | | −164 | | dBc/Hz |
| $f_{OUT}$ = 500 MHz | | | −163 | | dBc/Hz |
| $f_{OUT}$ = 950 MHz | | | −161 | | dBc/Hz |
| $f_{OUT}$ = 1850 MHz | | | −157 | | dBc/Hz |
| $f_{OUT}$ = 2150 MHz | | | −155 | | dBc/Hz |
| Single Tone, $f_{DAC}$ = 3000 MSPS | | | | | |
| $f_{OUT}$ = 200 MHz | | | -163 | | dBc/Hz |
| $f_{OUT}$ = 500 MHz | | | -159 | | dBc/Hz |
| $f_{OUT}$ = 950 MHz | | | -155 | | dBc/Hz |
| SINGLE-SIDEBAND PHASE NOISE OFFSET | Loop filter component values according to Figure 64 are as follows: C1 = 22 nF, R1 = 232 Ω, C2 = 2.4 nF, C3 = 33 nF; PFD frequency = 500 MHz, $f_{OUT}$ = 1.8 GHz, $f_{DAC}$ = 6 GHz | | | | |
| 1 kHz | | | −97 | | dBc/Hz |
| 10 kHz | | | −105 | | dBc/Hz |
| 100 kHz | | | −114 | | dBc/Hz |
| 600 kHz | | | −126 | | dBc/Hz |
| 1.2 MHz | | | −133 | | dBc/Hz |
| 1.8 MHz | | | −137 | | dBc/Hz |
| 6 MHz | | | −148 | | dBc/Hz |
| DAC TO DAC OUTPUT ISOLATION | Taken using the AD9171-FMC-EBZ evaluation board | | | | |
| Single-Band—$f_{DAC}$ = 6000 MSPS, Mode 0 (L = 1, M = 2) | | | | | |
| | $f_{OUT}$ = 1840 MHz | | −81 | | dBc |

## SPECIFICATIONS

*Table 8.*

| Parameter | Test Conditions/Comments | Min | Typ | Max | Unit |
|---|---|---|---|---|---|
| | $f_{OUT}$ = 2100 MHz | | −78 | | dBc |
| | $f_{OUT}$ = 2650 MHz | | −73 | | dBc |

## ABSOLUTE MAXIMUM RATINGS

*Table 9.*

| Parameter | Rating |
|---|---|
| ISET, FILT_COARSE, FILT_BYP, FILT_VCM | −0.3 V to AVDD1.8 + 0.3 V |
| SERDINx± | −0.2 V to SVDD1.0 + 0.2 V |
| $\overline{\text{SYNCOUT0}}$±, $\overline{\text{SYNCOUT1}}$±, $\overline{\text{RESET}}$, TXEN0, TXEN1, $\overline{\text{IRQ0}}$, $\overline{\text{IRQ1}}$, $\overline{\text{CS}}$, SCLK, SDIO, SDO | −0.3 V to DVDD1.8 + 0.3 V |
| DAC0±, DAC1±, CLKIN±, CLKOUT±, FILT_FINE | −0.2 V to AVDD1.0 + 0.2 V |
| SYSREF± | −0.2 V to DVDD1.0 + 0.2 V |
| AVDD1.0, DVDD1.0, SVDD1.0 to GND | −0.2 V to +1.2 V |
| AVDD1.8, DVDD1.8 to GND | −0.3 V to 2.2 V |
| Maximum Junction Temperature ($T_J$)[1] | 118°C |
| Storage Temperature Range | −65°C to +150°C |
| Reflow | 260°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 AD9171 reflow profile is in accordance with the JEDEC JESD20 criteria for Pb-free devices. The maximum reflow temperature is 260°C.

## THERMAL CHARACTERISTICS

Thermal performance is directly linked to printed circuit board (PCB) design and operating environment. Careful attention to PCB thermal design is required.

$\theta_{JA}$ is the natural convection junction to ambient thermal resistance measured in a one cubic foot sealed enclosure. $\theta_{JC}$ is the junction to case thermal resistance.

Thermal resistances and thermal characterization parameters are specified vs. the number of PCB layers in different airflow velocities (in m/sec). The use of appropriate thermal management techniques is recommended to ensure that the maximum junction temperature does not exceed the limits shown in Table 9.

Use the values in Table 10 in compliance with JEDEC 51-12.

*Table 10. Simulated Thermal Resistance vs. PCB Layers[1]*

| PCB Type | Airflow Velocity (m/sec) | $\theta_{JA}$ | $\theta_{JC_TOP}$ | $\theta_{JC_BOT}$ | Unit |
|---|---|---|---|---|---|
| JEDEC 2s2p Board | 0.0 | 25.3 | 2.4[2] | 3.0[3] | °C/W |
| | 1.0 | 22.6 | N/A | N/A | °C/W |
| | 2.5 | 21.0 | N/A | N/A | °C/W |
| 12-Layer PCB[4] | 0.0 | 15.4 | 2.4 | 2.6 | °C/W |
| | 1.0 | 13.1 | N/A | N/A | °C/W |
| | 2.5 | 11.6 | N/A | N/A | °C/W |

[1] N/A means not applicable.

[2] 1SOP PCB with no vias in PCB.

[3] 1SOP PCB with 7 × 7 standard JEDEC vias.

[4] Non JEDEC thermal resistance.

## ESD CAUTION

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

## PIN CONFIGURATION AND FUNCTION DESCRIPTIONS

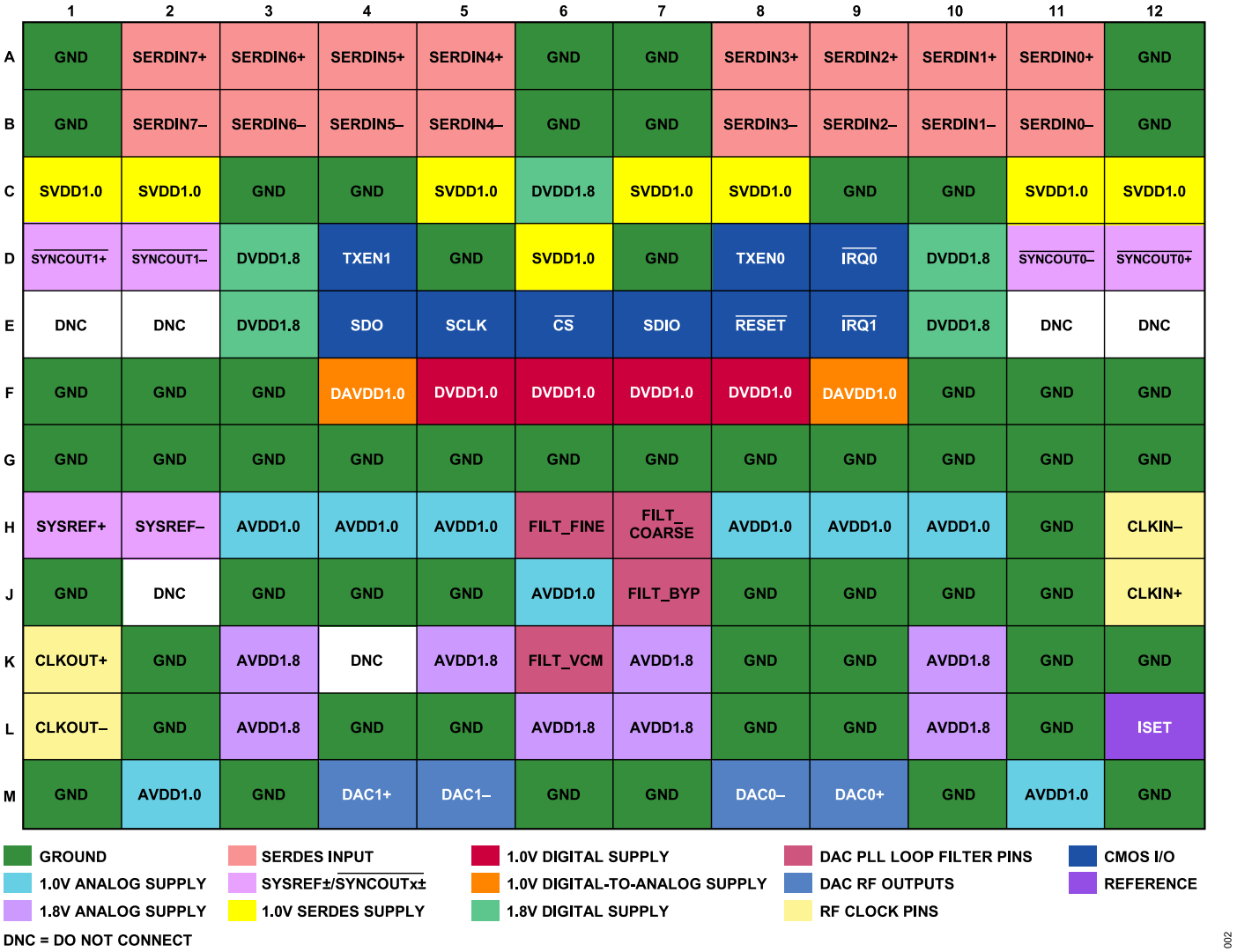


*Figure 2. Pin Configuration*

*Table 11. Pin Function Descriptions*

| Pin No. | Mnemonic | Description |
|---|---|---|
| 1.0 V Supply | | |
| H3 to H5, H8 to H10, J6, M2, M11 | AVDD1.0 | 1.0 V Clock and Analog Supplies. These pins supply the clock receivers, clock distribution, the on-chip DAC clock multiplier, and the DAC analog core. Clean power supply rail sources are required on these pins. |
| F5 to F8 | DVDD1.0 | 1.0 V Digital Supplies. These pins supply power to the DAC digital circuitry. Clean power supply rail sources are required on these pins. |
| F4, F9 | DAVDD1.0 | 1.0 V Digital to Analog Supplies. These pins can share a supply rail with the DVDD1.0 supply (electrically connected) but must have separate supply plane and decoupling capacitors for the PCB layout to improve isolation for these two pins. Clean power supply rail sources are required on these pins. |
| C1, C2, C5, C7, C8, C11, C12, D6 | SVDD1.0 | 1.0 V SERDES Supplies to the JESD204B Data Interface. Clean power supply rail sources are required on these pins. |
| 1.8 V Supply | | |
| K3, K5, K7, K10, L3, L6, L7, L10 | AVDD1.8 | 1.8 V Analog Supplies to the On-Chip DAC Clock Multiplier and the DAC Analog Core. Clean power supply rail sources are required on these pins. |

## PIN CONFIGURATION AND FUNCTION DESCRIPTIONS

*Table 11. Pin Function Descriptions*

| Pin No. | Mnemonic | Description |
|---|---|---|
| C6, D3, D10, E3, E10 | DVDD1.8 | 1.8 V Digital Supplies to the JESD204B Data Interface and the Other Input/Output Circuitry, Such as the Serial Port Interface (SPI). Clean power supply rail sources are required on these pins. |
| Ground | | |
| A1, A6, A7, A12, B1, B6, B7, B12, C3, C4, C9, C10, D5, D7, F1 to F3, F10 to F12, G1 to G12, H11, J1, J3 to J5, J8 to J11, K2, K8, K9, K11, K12, L2, L4, L5, L8, L9, L11, M1, M3, M6, M7, M10, M12 | GND | Device Common Ground. |
| RF Clock | | |
| J12 | CLKIN+ | Positive Device Clock Input. This pin is the clock input for the on-chip DAC clock multiplier, REFCLK, when the DAC PLL is on. This pin is also the clock input for the DAC sample clock or device clock (DACCLK) when the DAC PLL is off. AC couple this input. There is an internal 100 Ω resistor between this pin and CLKIN−. |
| H12 | CLKIN− | Negative Device Clock Input. |
| K1 | CLKOUT+ | Positive Device Clock Output. This pin is the clock output of a divided down DACCLK and is available with the DAC PLL on and off. The divide down ratios are by 1, 2, or 4. |
| L1 | CLKOUT− | Negative Device Clock Output. |
| System Reference | | |
| H1 | SYSREF+ | Positive System Reference Input. It is recommended to ac couple this pin, but dc coupling is also acceptable. See Table 6 for the dc common-mode voltage. |
| H2 | SYSREF− | Negative System Reference Input. It is recommended to ac couple this pin, but dc coupling is also acceptable. See Table 6 for the dc common-mode voltage. |
| On-Chip DAC PLL Loop Filter | | |
| H6 | FILT_FINE | On-Chip DAC Clock Multiplier and PLL Fine Loop Filter Input. |
| H7 | FILT_COARSE | On-Chip DAC Clock Multiplier and PLL Coarse Loop Filter Input. |
| J7 | FILT_BYP | On-Chip DAC Clock Multiplier and LDO Bypass. |
| K6 | FILT_VCM | On-Chip DAC Clock Multiplier and VCO Common-Mode Input. |
| SERDES Data Bits | | |
| A2 | SERDIN7+ | SERDES Data Bit 7, Positive. |
| B2 | SERDIN7− | SERDES Data Bit 7, Negative. |
| A3 | SERDIN6+ | SERDES Data Bit 6, Positive. |
| B3 | SERDIN6− | SERDES Data Bit 6, Negative. |
| A4 | SERDIN5+ | SERDES Data Bit 5, Positive. |
| B4 | SERDIN5− | SERDES Data Bit 5, Negative. |
| A5 | SERDIN4+ | SERDES Data Bit 4, Positive. |
| B5 | SERDIN4− | SERDES Data Bit 4, Negative. |
| A8 | SERDIN3+ | SERDES Data Bit 3, Positive. |
| B8 | SERDIN3− | SERDES Data Bit 3, Negative. |
| A9 | SERDIN2+ | SERDES Data Bit 2, Positive. |
| B9 | SERDIN2− | SERDES Data Bit 2, Negative. |
| A10 | SERDIN1+ | SERDES Data Bit 1, Positive. |
| B10 | SERDIN1− | SERDES Data Bit 1, Negative. |
| A11 | SERDIN0+ | SERDES Data Bit 0, Positive. |
| B11 | SERDIN0− | SERDES Data Bit 0, Negative. |
| Sync Output | | |
| D12 | $\overline{\text{SYNCOUT0+}}$ | Positive Sync (Active Low) Output Signal, Channel Link 0. This pin is LVDS or CMOS selectable. |
| D11 | $\overline{\text{SYNCOUT0−}}$ | Negative Sync (Active Low) Output Signal, Channel Link 0. This pin is LVDS or CMOS selectable. |
| D1 | $\overline{\text{SYNCOUT1+}}$ | Positive Sync (Active Low) Output Signal, Channel Link 1. This pin is LVDS or CMOS selectable. |
| D2 | $\overline{\text{SYNCOUT1−}}$ | Negative Sync (Active Low) Output Signal, Channel Link 1. This pin is LVDS or CMOS selectable. |
| Serial Port Interface | | |

## PIN CONFIGURATION AND FUNCTION DESCRIPTIONS

*Table 11. Pin Function Descriptions*

| Pin No. | Mnemonic | Description |
|---|---|---|
| E4 | SDO | Serial Port Data Output (CMOS Levels with Respect to DVDD1.8). |
| E7 | SDIO | Serial Port Data Input/Output (CMOS Levels with Respect to DVDD1.8). |
| E5 | SCLK | Serial Port Clock Input (CMOS Levels with Respect to DVDD1.8). |
| E6 | $\overline{CS}$ | Serial Port Chip Select, Active Low (CMOS Levels with Respect to DVDD1.8). |
| E8 | $\overline{RESET}$ | Reset, Active Low (CMOS Levels with Respect to DVDD1.8). |
| Interrupt Request | | |
| D9 | $\overline{IRQ0}$ | Interrupt Request 0. This pin is an open-drain, active low output (CMOS levels with respect to DVDD1.8). Connect a pull-up resistor to DVDD1.8 to prevent this pin from floating when inactive. |
| E9 | $\overline{IRQ1}$ | Interrupt Request 1. This pin is an open-drain, active low output (CMOS levels with respect to DVDD1.8). Connect a pull-up resistor to DVDD1.8 to prevent this pin from floating when inactive. |
| CMOS Input/Outputs | | |
| D8 | TXEN0 | Transmit Enable for DAC0. The CMOS levels are determined with respect to DVDD1.8. |
| D4 | TXEN1 | Transmit Enable for DAC1. The CMOS levels are determined with respect to DVDD1.8. |
| DAC Analog Outputs | | |
| M9 | DAC0+ | DAC0 Positive Current Output. |
| M8 | DAC0− | DAC0 Negative Current Output. |
| M4 | DAC1+ | DAC1 Positive Current Output. |
| M5 | DAC1− | DAC1 Negative Current Output. |
| Reference | | |
| L12 | ISET | Device Bias Current Setting Pin. Connect a 5 kΩ resistor from this pin to GND, preferably with <0.1% tolerance and <±25 ppm/°C temperature coefficient. |
| Do Not Connect | | |
| E1, E2, E11, E12, J2, K4 | DNC | Do Not Connect. Do not connect to these pins. |

## TYPICAL PERFORMANCE CHARACTERISTICS

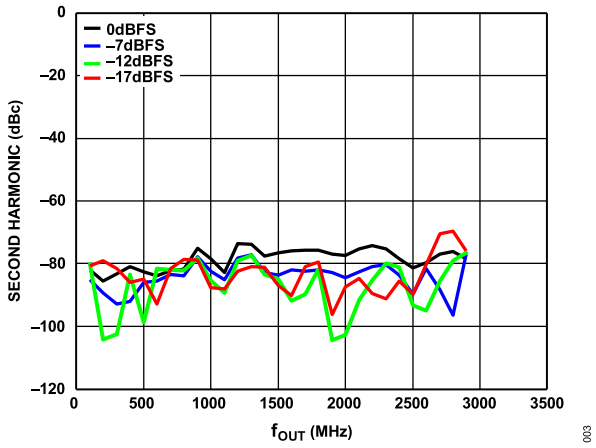


*Figure 3. Second Harmonic (dBc) vs. $f_{OUT}$ over Digital Scale (Mode 0),6 GHz DAC Sample Rate, Channel Interpolation 2×, Main Interpolation 8×*

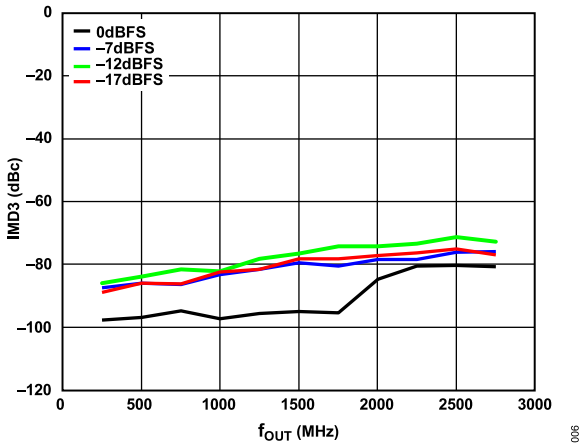


*Figure 6. IMD3 vs. $f_{OUT}$ over Digital Scale (Mode 0) 6 GHz DAC Sample Rate, Channel Interpolation 2×, Main Interpolation 8×, 1 MHz Tone Spacing*

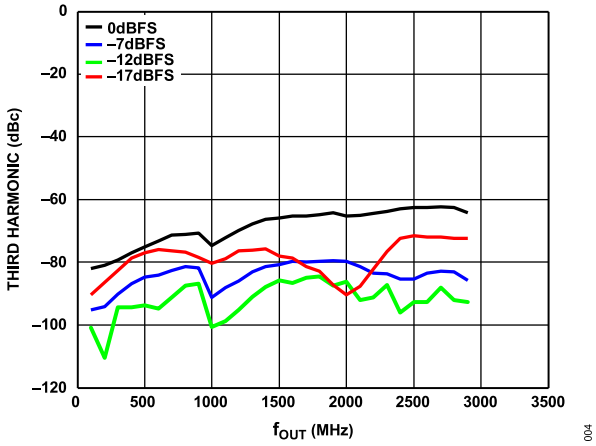


*Figure 4. Third Harmonic (dBc) vs. $f_{OUT}$ over Digital Scale (Mode 0),6 GHz DAC Sample Rate, Channel Interpolation 2×, Main Interpolation 8×*

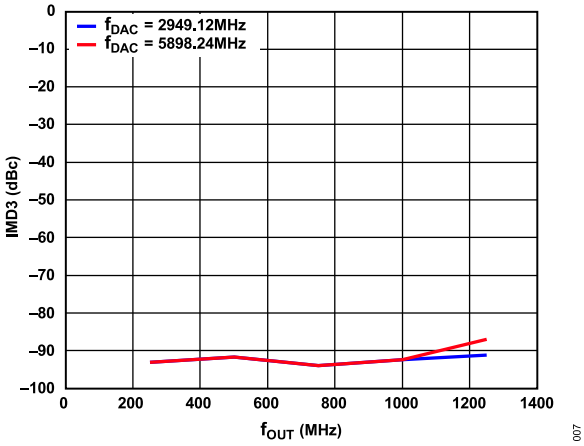


*Figure 7. IMD3 vs. $f_{OUT}$ over $f_{DAC}$ (Mode 0), Channel Interpolation 2×, Main Interpolation 8×, 1 MHz Tone Spacing*

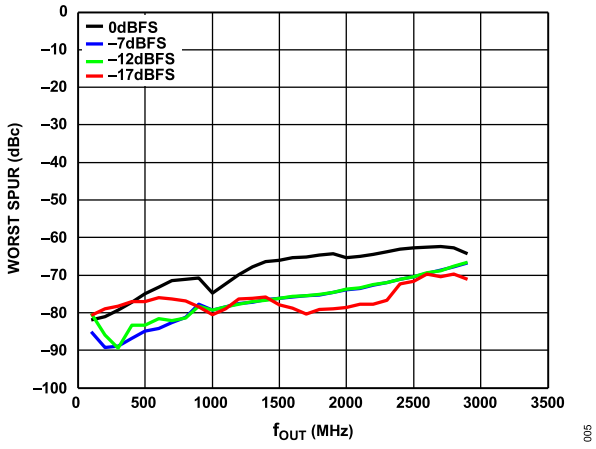


*Figure 5. Worst Spur (dBc) vs. $f_{OUT}$ over Digital Scale (Mode 0),6 GHz DAC Sample Rate, Channel Interpolation 2×, Main Interpolation 8×*

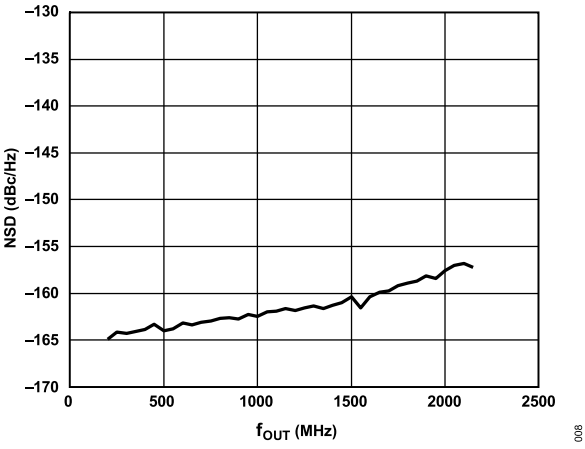


*Figure 8. Single-Tone NSD Measured at 70 MHz vs. $f_{OUT}$, $f_{DAC}$ = 5.89824 GHz, 16-Bit Resolution, Shuffle On*

## TYPICAL PERFORMANCE CHARACTERISTICS

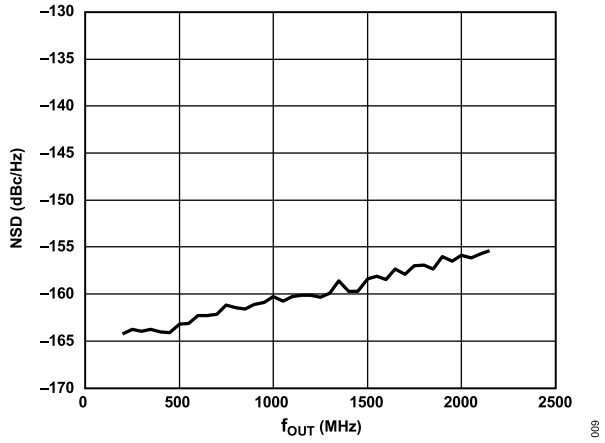


*Figure 9. Single-Tone NSD Measured at 10% Offset from $f_{OUT}$ vs. $f_{OUT}$, $f_{DAC}$ = 5.89824 GHz, 16-Bit Resolution, Shuffle On*

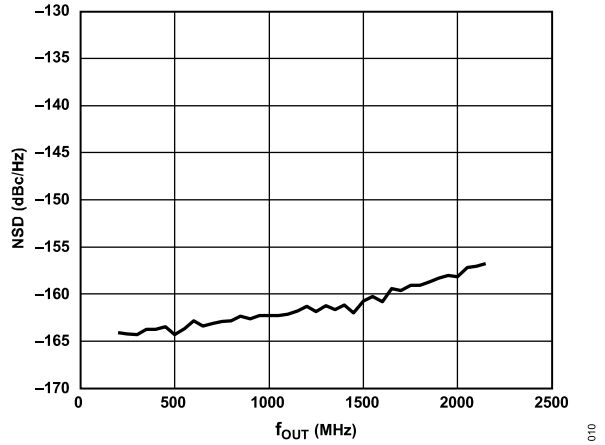


*Figure 10. Single-Tone NSD Measured at 70 MHz vs. $f_{OUT}$, $f_{DAC}$ = 5.89824 GHz,12-Bit Resolution, Shuffle On*

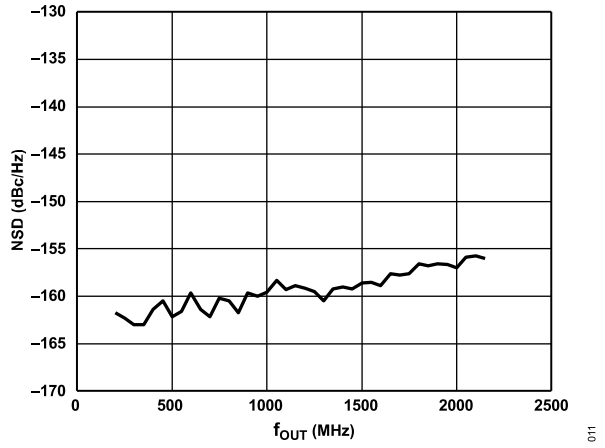


*Figure 11. Single-Tone NSD Measured at 10% Offset from $f_{OUT}$ vs. $f_{OUT}$, $f_{DAC}$ = 5.89824 GHz, 12-Bit Resolution, Shuffle On*

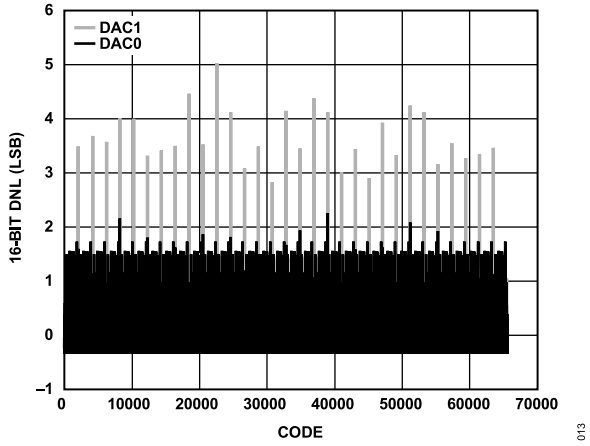


*Figure 12. DNL, $I_{OUTFS}$ = 26 mA, 16-Bit Resolution*

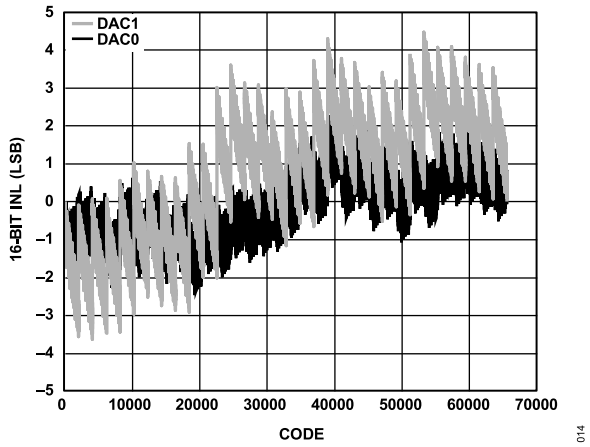


*Figure 13. INL, $I_{OUTFS}$ = 26 mA, 16-Bit Resolution*

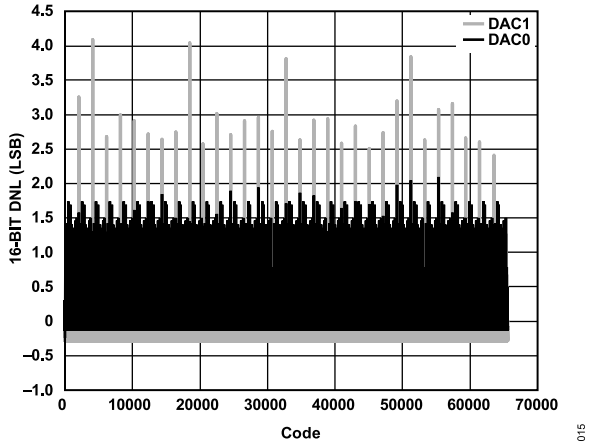


*Figure 14. DNL, $I_{OUTFS}$ = 20 mA, 16-Bit Resolution*

## TYPICAL PERFORMANCE CHARACTERISTICS

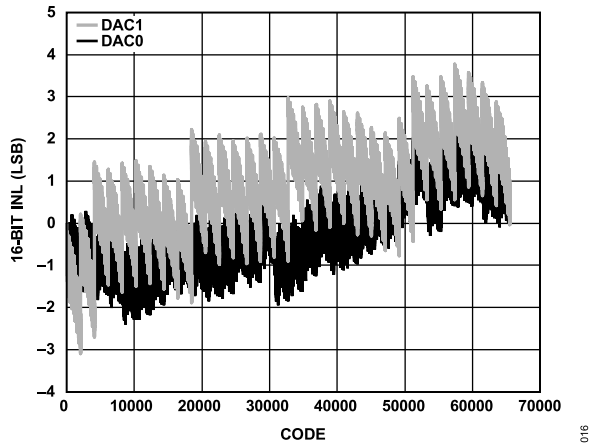


Figure 15. INL, $I_{OUTFS}$ = 20 mA, 16-Bit Resolution

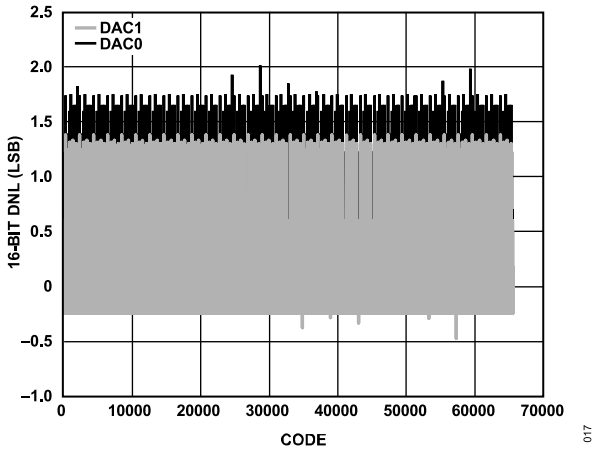


Figure 16. DNL, $I_{OUTFS}$ = 15.6 mA, 16-Bit Resolution

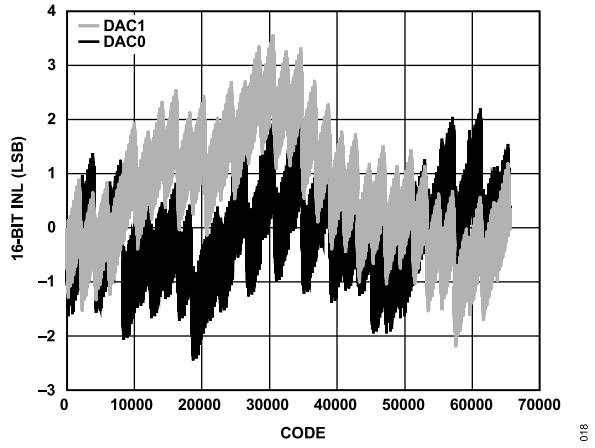


Figure 17. INL, $I_{OUTFS}$ = 15.6 mA, 16-Bit Resolution

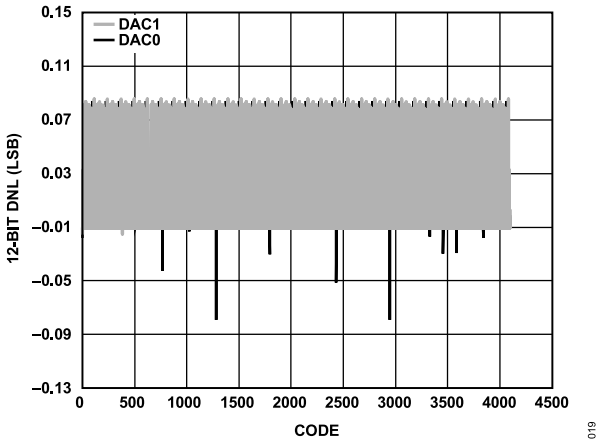


Figure 18. DNL, $I_{OUTFS}$ = 20 mA, 12-Bit Resolution

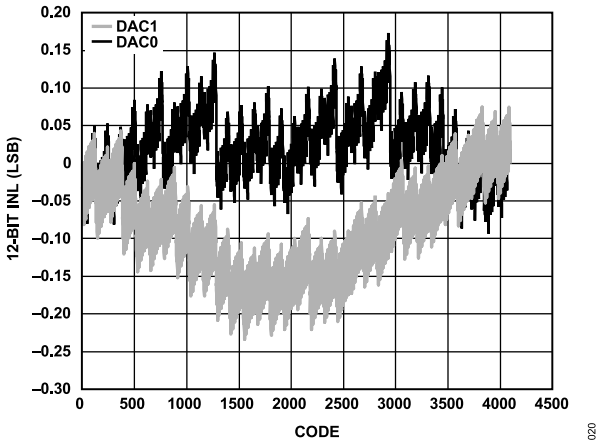


Figure 19. INL, $I_{OUTFS}$ = 20 mA, 12-Bit Resolution

## 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 value of 0 mA. For DACx+, a 0 mA output is expected when all inputs are set to 0. For DACx−, a 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.

### Output Compliance Range

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

### Temperature Drift

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

### Channel Datapath

The channel datapaths, sometimes referred to as channelizers, are the complex data channel datapaths, before the summing node in the chip, that can be used or bypassed depending on the mode of operation chosen. When these channelizers are in use, complex data input is required. The channel datapaths include independently controlled optional gain stages and channel NCOs per channel. There is also a selectable channel interpolation block that is configurable (same setting for all channel interpolation blocks) depending on the mode of operation chosen.

### Main Datapath

The main datapath refers to the portion of the digital datapath after the summing node in the chip, up to each of the main DAC analog cores. Each of these main datapaths includes an optional PA protection block with a feed forward to the ramp up/down gain stage block for muting the DAC outputs before damaging a power amplifier in the transmit path. There is a selectable main interpolation block that is configurable (same setting for both main interpolation blocks) depending on the mode of operation chosen. Each main datapath also contains an individually programmable main NCO per main DAC datapath that can be optionally used depending on the mode of operation.

### 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 (PHY) 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 per link. When paging Link 0 (Register 0x300, Bit 2 = 0), Link Lane x = Logical Lane x.

## TERMINOLOGY

When paging Link 1 (Register 0x300, Bit 2 = 1, dual link only), Link Lane x = Logical Lane x + 4.

## THEORY OF OPERATION

The AD9171 is a 16-bit, dual RF DAC with a high speed JESD204B SERDES interface, compliant with Subclass 0 and Subclass 1 operation. Figure 1 shows a functional block diagram of the AD9171. Each DAC core has one channelizer that supports up to 516 MSPS of complex data rate input per channel. Eight high speed serial lanes carry data at a maximum of 15.4 Gbps to the channel datapaths. The JESD204B interface supports both single-link and dual-link modes of operation, depending on the selected mode configuration. 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 can be sourced with a PLL reference clock used by the on-chip PLL to generate a DAC clock, or a high fidelity, direct external DAC sampling clock. The device can be configured to operate in one- or two-lane per link modes, depending on the required input data rate.

The digital datapath of the AD9171 offers selectable 2×, 3×, 4×, and 6× interpolation options for the channel datapaths, and 6× and 8× interpolation options for the main datapaths. See Table 12 for a summary of the various JESD204B modes available, as well as the respective interpolation options.

For each of the channel digital datapaths, there are individually programmable gain stages and NCO blocks available. The NCO blocks have a 48-bit modulus NCO option to enable digital frequency shifts of signals with near infinite precision.

The NCO can operate alone in NCO only mode using a programmable dc value input via the SPI or with digital data from the SERDES interface and digital datapath. At the end of the channelizer datapaths, the data then passes along to each of the main DAC datapaths for further digital feature options.

Each of the main DAC datapaths contain an optional power amplifier (PA) protection block, a main datapath interpolation block, a main NCO with an optional modulus feature, and a ramp-up/ramp-down gain block that is fed by the PA protection block. Additionally, there is an optional calibration tone feature, as well as four modulator switch modes that are part of the main NCO block.

The AD9171 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 signal (SYSREF±) makes the AD9171 JESD204B 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 software support.

This data sheet describes the various blocks of the AD9171 in 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.

***Table 12. JESD204B Supported Operating Modes and Interpolation Combinations***

| | JESD204B Operation Modes | | | | Channel Datapath | | Main DAC Datapath | |
|---|---|---|---|---|---|---|---|---|
| Application | Link Modes | JESD204B Modes | Lanes per Link | Channels per DAC | Maximum Channel Data Rate (MSPS)[1] | Channel Interpolations | Main Datapath Interpolations | Maximum DAC Rate (GSPS) |
| Single Channel, 375 MHz (N = 16 Bits) | Single, dual | 0 | 1 | 1 | 385 | 2× | 8× | 6.2 |
| | | | | | | 4×, 6× | 6×, 8× | 6.2 |
| Single Channel, 500 MHz (N = 12 Bits, NP = 12 Bits) | Single, dual | 5 | 1 | 1 | 513 | 2× | 6× | 6.2 |
| | | | | | | 3× | 6×, 8× | 6.2 |
| Single Channel, 500 MHz (N = 16 Bits) | Single, dual | 3 | 2 | 1 | 516 | 2×, 3× | 6×, 8× | 6.2 |

[1] The maximum data rate is calculated based on either the maximum lane rate as listed in Table 6 or the maximum DAC rate. The data rate is calculated either using the formula lane rate = (10/8) × NP × data rate × (M/L), where the NP, M, and L values depend on the selected mode or the DAC rate is data rate × minimum channel interpolation × minimum main datapath interpolation, whichever value for the data rate is smallest.

## 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 is compatible with most synchronous transfer formats, including both the Motorola, Inc., SPI and Intel® SSR protocols. The interface allows read and write access to all registers that configure the AD9171. 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 input/output (SDIO).

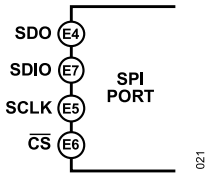


*Figure 20. Serial Port Interface Pins (144-Ball BGA_ED)*

There are two phases to a communication cycle with the AD9171. 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 input/output 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 required 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 NCO phase offsets, which change only when the frequency tuning word load request bit (DDSM_FTW_LOAD_REQ or DDSC_FTW_LOAD_REQ) 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 for 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 80 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 communication 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 Input/Output (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 (LSBFIRST 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,

## SERIAL PORT OPERATION

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.

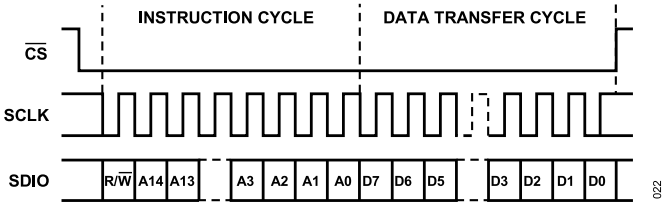


*Figure 21. Serial Register Interface Timing, MSB First, Register 0x000, Bit 6 and Bit 1 = 0*

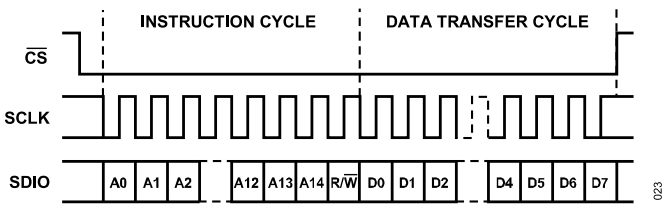


*Figure 22. Serial Register Interface Timing, LSB First, Register 0x000, Bit 6 and Bit 1 = 1*

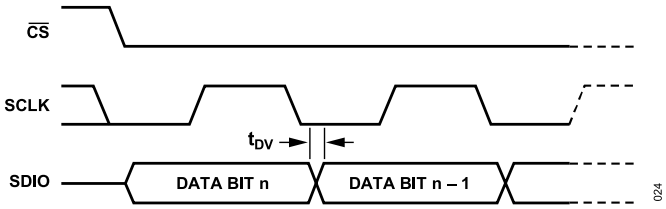


*Figure 23. Timing Diagram for Serial Port Register Read*

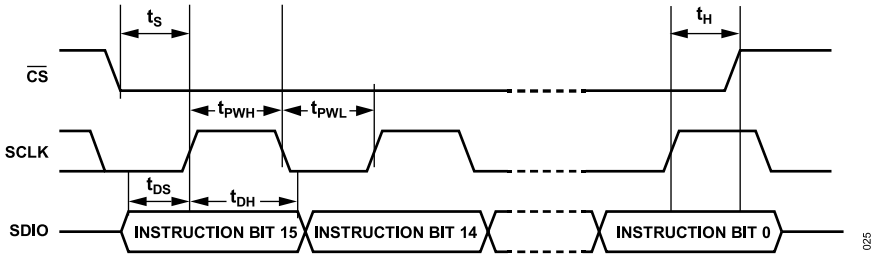


*Figure 24. Timing Diagram for Serial Port Register Write*

## JESD204B SERIAL DATA INTERFACE

### JESD204B OVERVIEW

The AD9171 has eight JESD204B data ports that receive data. The eight JESD204B ports can be configured as part of a single or dual JESD204B link that uses a single system reference (SYSREF±) and device clock (CLKIN±).

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 25 shows the communication layers implemented in the AD9171 serial data interface to recover the clock and deserialize, descramble, and deframe the data before it is sent to each of the digital signal processing channelizers of the device.

The physical layer establishes a reliable channel between the transmitter (Tx) and the receiver (Rx), and 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 AD9171 also has a descrambling option (see the Descrambler section for more information). The AD9171 has the ability to use 12-bit packing mode (NP = 12, N = 12) to increase the maximum data rate achievable by this device for applications that do not require 16-bit resolution capabilities.

The AD9171 has multiple single-link and dual-link mode options available for various application purposes. These modes and their respective JESD204B link parameters are described in Table 14.

There are different interpolation combinations available for the channel and main datapaths, as well as whether single-link and dual-link options are available depending on which JESD204B mode is chosen. Table 12 lists the possible link and interpolation combinations available.

The AD9171 has two DAC outputs; however, for the purposes of complex signal processing on chip, the converter count, represented by the M JESD204B parameter, reflects the number of complex subchannels of data required per link when using a total interpolation greater than 1×. The number of complex subchannels of data per link required to be sent to the device also depends on the number of channelizers being used, based on the mode of operation chosen. If the channelizer datapaths are bypassed (channel interpolation is set to 1×) and the main datapath interpolation is set to 1×, the converter count (M) reflects the number of real converters per link being used in the mode of operation; in this case, complex data is not required.

For a particular JESD204B mode of operation, the following relationships exist:

*Total Interpolation* = *Channel Interpolation* × *Main Interpolation*

*Data Rate* = *DAC Rate*/*Total Interpolation*

*Lane Rate* = (*M*/*L*) × *NP* × (10/8) × *Data Rate*

where:
*Lane Rate* must be between 3 Gbps and 15 Gbps.
*M*, *L*, and *NP* are JESD204B link parameters for the chosen JESD204B operating mode.

Achieving and recovering synchronization of the lanes is important. To simplify the interface to the transmitter, the AD9171 designates a master synchronization signal for each JESD204B link. The SYNCOUT0± and SYNCOUT1± pins are used as the master signal for all lanes on each link. 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.

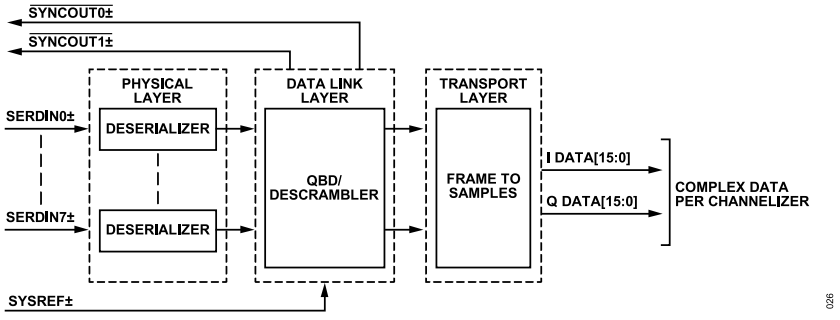


*Figure 25. Functional Block Diagram of Serial Link Receiver*

## JESD204B SERIAL DATA INTERFACE

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

| | Single-Link or Dual-Link JESD204B Modes | | |
|---|---|---|---|
| **Parameter** | **0** | **3** | **5** |
| Lane Count (L) | 1 | 2 | 1 |
| Converter Count (M) | 2 | 2 | 2 |
| Octets per Frame per Lane (F) | 4 | 2 | 3 |
| Samples per Converter per Frame (S) | 1 | 1 | 1 |
| Total Number of Bits per Sample (NP) | 16 | 16 | 12 |
| Converter Resolution (N) | 16 | 16 | 12 |
| Frames per Multiframe (K) | 32 | 32 | 32 |
| High Density User Data Format (HD) | 1 | 1 | 1 |

*Table 15. Data Structure per Lane for F = 2 JESD204B Operating Modes[1]*

| | | Frame 0 | | Frame 1 | |
|---|---|---|---|---|---|
| **JESD204B Mode and Parameters** | **Link Logical Lane** | **Octet 0** | **Octet 1** | **Octet 0** | **Octet 2** |
| Mode 3 (L = 2, M = 2, S = 1, NP = 16, N = 16) | Lane 0 | M0S0[15:8] | M0S0[7:0] | M0S1[15:8] | M0S1[7:0] |
| | Lane 1 | M1S0[15:8] | M1S0[7:0] | M1S1[15:8] | M1S1[7:0] |

[1] Mx is the converter number and Sy is the sample number. For example, M0S0 means Converter 0, Sample 0.

*Table 16. Data Structure per Lane for F = 3 JESD204B Operating Modes[1]*

| | | Frame 0 | | | | | |
|---|---|---|---|---|---|---|---|
| | | Octet 0 | | Octet 1 | | Octet 2 | |
| **JESD204B Mode and Parameters** | **Link Logical Lane** | **Nibble 0** | **Nibble1** | **Nibble 0** | **Nibble1** | **Nibble 0** | **Nibble1** |
| Mode 5 (L = 1, M = 2, S = 1, NP = 12, N = 12) | Lane 0 | M0S0[11:8] | M0S0[7:4] | M0S0[3:0] | M1S0[11:8] | M1S0[7:4] | M1S0[3:0] |

[1] Mx is the converter number and Sy is the sample number. For example, M0S0 means Converter 0, Sample 0.

*Table 17. Data Structure per Lane for F = 4 JESD204B Operating Modes[1]*

| | | Frame 0 | | | | Frame 1 | | | |
|---|---|---|---|---|---|---|---|---|---|
| **JESD204B Mode and Parameters** | **Link Logical Lane** | **Octet 0** | **Octet 1** | **Octet 2** | **Octet 3** | **Octet 0** | **Octet 1** | **Octet 2** | **Octet 3** |
| Mode 0 (L = 1, M = 2, S = 1, NP = 16, N = 16) | Lane 0 | M0S0[15:8] | M0S0[7:0] | M1S0[15:8] | M1S0[7:0] | M0S1[15:8] | M0S1[7:0] | M1S1[15:8] | M1S1[7:0] |

[1] Mx is the converter number and Sy is the sample number. For example, M0S0 means Converter 0, Sample 0.

## JESD204B SERIAL DATA INTERFACE

### PHYSICAL LAYER

The physical layer of the JESD204B interface, hereafter referred to as the deserializer, has eight identical channels. Each channel consists of the termination, an equalizer, a clock and data recovery (CDR) circuit, and the 1:40 demux function (see Figure 26).

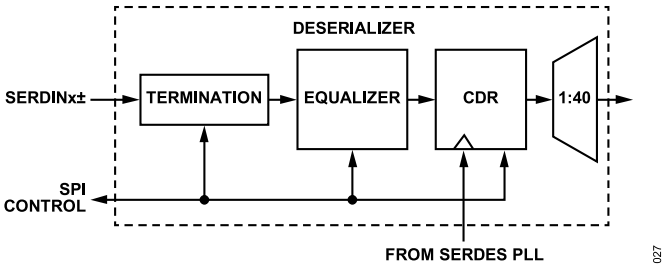


*Figure 26. Deserializer Block Diagram*

JESD204B data is input to the AD9171 via the SERDINx± differential input pins, 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 AD9171 autocalibrates the input termination to 50 Ω. This calibration routine is performed automatically when the JESD204B interface blocks are configured and does not require any additional SPI register writes.

### Receiver Eye Mask

The AD9171 is compatible with the JESD204B specification regarding the receiver eye mask and is capable of capturing data that complies with the mask in Figure 27. Figure 27 shows the receiver eye normalized to the data rate interval. The AD9171 also supports an increased insertion loss limit, as defined in the Equalization section.

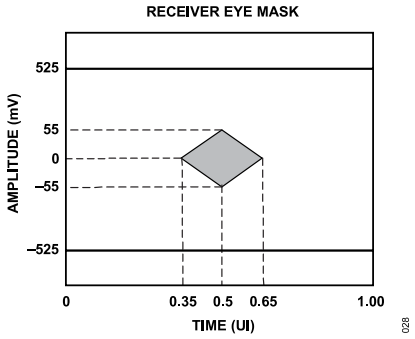


*Figure 27. Receiver Eye Mask*

### 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:

*Data Rate = DAC Rate/Total Interpolation*

*Lane Rate = (M/L) × NP × (10/8) × Data Rate*

*Byte Rate = Lane Rate*/10

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

*PCLK Rate = Byte Rate*/4

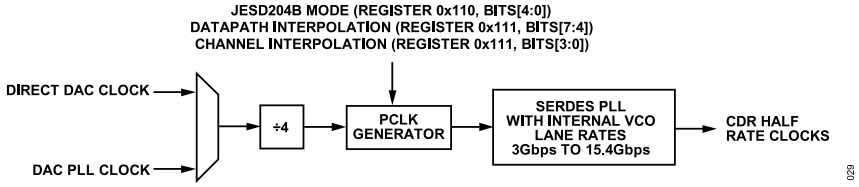


*Figure 28. SERDES PLL Synthesizer Block Diagram Including VCO Divider Block*

## JESD204B SERIAL DATA INTERFACE

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

*Frame Rate = Byte Rate/F*

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

*PCLK Factor = Frame Rate/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.
*NP* is the JESD204B parameter for the total number of bits per sample.

### 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 is capable of providing quadrature clocks to allow a wide range of data rates (3 Gbps to 15 Gbps) with no gaps.

These clocks are the input to the CDR 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). For more information about the SERDES circuitry setup and relevant register writes, see the Start-Up Sequence section. The SERDES PLL block automatically tunes to the appropriate divider range for the lane rate based on the SERDES mode being used. It takes the DAC clock generated by either the DAC PLL, if in use, or from the direct clock being sourced at the CLKIN± pins, divides the DAC clock frequency by 4, and uses the JESD204B parameters corresponding to the mode and interpolation values programmed in Register 0x110 and Register 0x111 to determine the proper dividers for generating the PCLK frequency (lane rate ÷ 40), as shown in Figure 28.

Confirm that the SERDES PLL is working by reading Register 0x281. If Register 0x281, Bit 0 = 1, the SERDES PLL has locked.

### 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 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.

### Power-Down Unused PHYs

Note that any unused physical 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 AD9171 employs an easy to use, low power equalizer on each JESD204B channel. The AD9171 equalizers operating at the maximum lane rate of 15 Gbps can compensate for up to 16 dB of insertion loss. This equalizer performance is shown in Figure 29 for 15 Gbps, near the maximum baud rate for the AD9171. The channel must also meet the insertion loss deviation requirement of the JESD204B specification (less than 1.5 dB from 50 MHz to 0.75 times the baud rate).

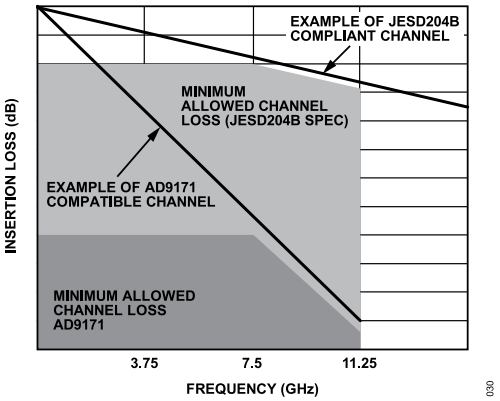


*Figure 29. Insertion Loss Allowed*

To ensure the AD9171 compensates for the amount of insertion loss in the system, set the equalizer block appropriately. Table 18 shows the settings for the equalizer boost, equalizer gain, and feedback controls, depending on the level of insertion loss in the system. The equalizer boost setting is programmed for each PHY lane (2-bit control for each) being used in Register 0x240 and Register 0x241. Similarly, the equalizer gain settings are programmed for each PHY lane (2-bit control for each) used in Register 0x242 and Register 0x243. The feedback control is programmed per PHY lane (5-bit control for each, one control per register) in Register 0x244 to Register 0x24B.

*Table 18. Equalizer Register Control Settings per PHY Control*

| Insertion Loss | ≤11 dB | >11 dB |
|---|---|---|
| Equalizer Boost | 0x02 | 0x03 |
| Equalizer Gain | 0x01 | 0x03 |
| Feedback | 0x1F | 0x1F |

Figure 30 and Figure 31 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,

## JESD204B SERIAL DATA INTERFACE

respectively. See the Hardware Considerations section for specific layout recommendations for the JESD204B channel.

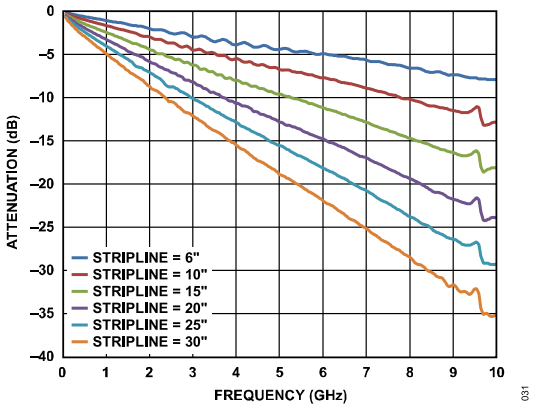


*Figure 30. Insertion Loss of 50 Ω Striplines on FR4*

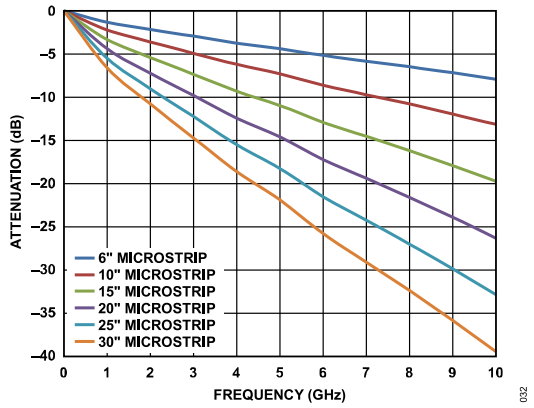


*Figure 31. Insertion Loss of 50 Ω Microstrips on FR4*

## DATA LINK LAYER

The data link layer of the AD9171 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 32. The data link layer consists of a synchronization first in/first out (FIFO) for each lane, a crossbar switch, a deframer, and a descrambler.

The AD9171 can operate as a single-link or dual-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 AD9171 decodes 8-bit/10-bit control characters, allowing marking of the start and end of the frame and alignment between serial lanes. Each AD9171 serial interface link can issue a synchronization request by setting its $\overline{\text{SYNCOUTx±}}$ signals low. The synchronization protocol follows Section 4.9 of the JESD204B standard. When a stream of four consecutive /K/ symbols is received, the AD9171 deactivates the synchronization request by setting the $\overline{\text{SYNCOUTx±}}$ signals high at the next internal LMFC rising edge. Then, the AD9171 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 33).

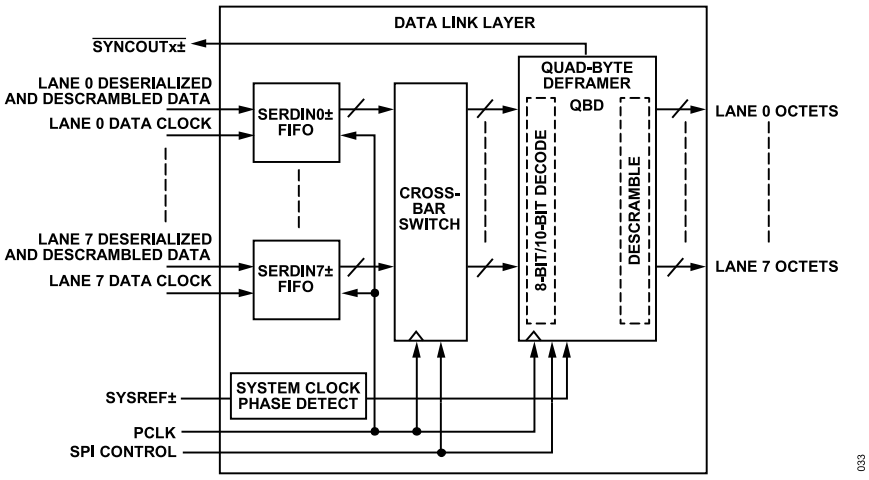


*Figure 32. Data Link Layer Block Diagram*

## JESD204B SERIAL DATA INTERFACE

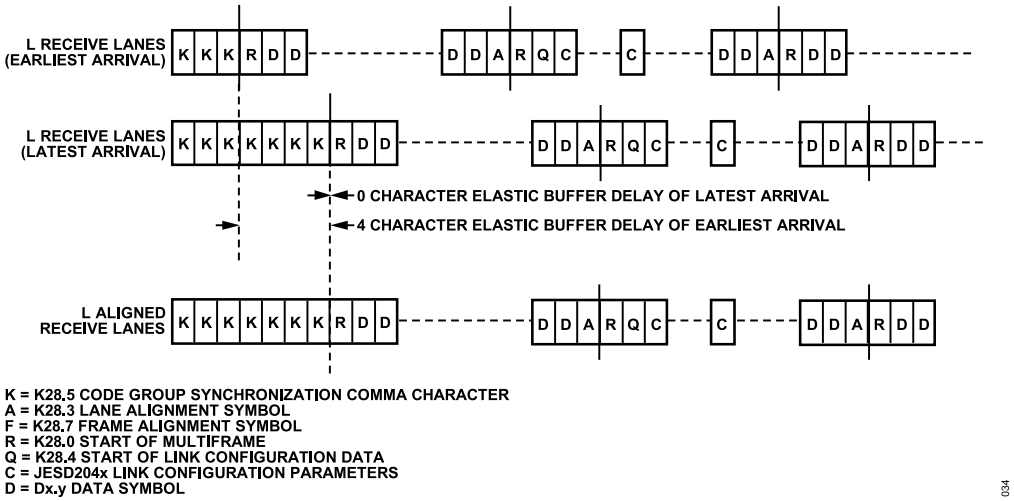


*Figure 33. Lane Alignment During ILAS*

### 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 specification 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 $\overline{\text{SYNCOUTx±}}$ signals to the transmitter block at the receiver LMFC edge.

The transmitter captures the change in the $\overline{\text{SYNCOUTx±}}$ signals 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 the ILAS. 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 AD9171 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 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):

- $\overline{\text{SYNCOUTx±}}$ signal assertion: resynchronization ($\overline{\text{SYNCOUTx±}}$ signals 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 the respective $\overline{\text{SYNCOUTx±}}$ pins.
- 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.

## JESD204B SERIAL DATA INTERFACE

### 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 lane 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 AD9171 consists of two quad-byte deframers (QBDs) paged by the LINK_PAGE control in Register 0x300, Bit 2. 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 AD9171 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 AD9171 requires a synchronization (sync) to align the LMFC and other internal clocks before the SERDES links are brought online. The synchronization is a one-shot sync, where the synchronization process begins on the next edge of the alignment signal following the assertion of the SYSREF_MODE_ONESHOT control in Register 0x03A, Bit 1.

In Subclass 1, the SYSREF± rising edge acts as the alignment edge; in Subclass 0, an internal processing clock acts as the alignment edge. When a sync has completed, the SYNC_ROTATION_DONE (Register 0x03A, Bit 4) bit is asserted and remains asserted until another sync is requested.

After a synchronization occurs, the JESD204B link can be enabled. In Subclass 1, the latency of the JESD204B system is deterministic and allows synchronization across multiple devices, if desired.

### 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 AD9171 supports a periodic SYSREF± signal. The periodicity can be continuous, strobed, or gapped periodic. The SYSREF± signal can be dc-coupled with a common-mode voltage of 0.6 V to 2.2 V and differential swing of 200 mV p-p to 1 V p-p. When dc-coupled, a small amount of common-mode current (up to 0.3 mA) is drawn from the SYSREF± pins. See Figure 34 and Figure 35 for the SYSREF± internal circuit for dc-coupled and ac-coupled configurations. Ensure that the SYSREF_INPUTMODE bit (Register 0x084, Bit 6) is set to 1, dc-coupled, to prevent overstress on the SYSREF± receiver pins.

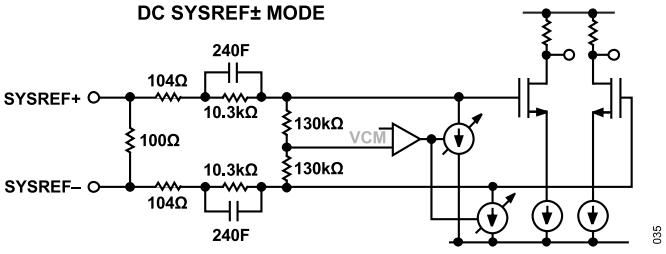


*Figure 34. DC-Coupled SYSREF± Receiver Circuitry*

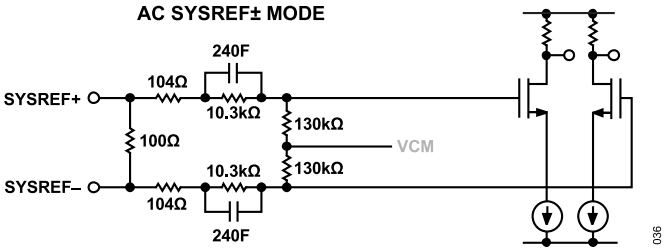


*Figure 35. AC-Coupled SYSREF± Receiver Circuitry*

To avoid this common-mode current draw, the SYSREF± receiver can be ac-coupled using 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 35 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 allow SYSREF± sampling clocks to correctly sample the rising SYSREF± edge before the next sample clock.

When ac coupling the SYSREF± inputs, ensure that the SYSREF_INPUTMODE bit (Register 0x084, Bit 6) is set to 0, ac-coupled, to enable the internal receiver biasing circuitry and prevent overstress on the SYSREF± receiver pins. AC coupling allows a differential voltage swing from 200 mV to 1 V on the SYSREF± pins.

## SYSREF± Sampling

The SYSREF± signal is sampled by a divide by 4 version of the DAC clock. Thus, the minimum pulse width of the SYSREF± signal must exceed 4 DAC clock periods to ensure accurate sampling.

The delay between the SYSREF± and DAC clock input signal does not need to be timing constrained.

By default, the first SYSREF± rising edge at the SYSREF± inputs that is detected after asserting the SYSREF_MODE_ONESHOT bit (Register 0x03A, Bit 1) begins the synchronization and aligns the internal LMFC signal with the sampled SYSREF± edge.

Register 0x036 (SYSREF_COUNT) indicates how many captured SYSREF± edges are ignored after the SYSREF_MODE_ONE-SHOT bit is asserted before the synchronization takes place. For example, if SYSREF_COUNT is set to 3, the AD9171 does not sync after the SYSREF_MODE_ONESHOT bit is asserted until the arrival of the 4th SYSREF± edge.

## SYSREF± Jitter IRQ

In Subclass 1, after the one-shot synchronization occurs, the SYS-REF± signal is monitored to ensure that the subsequent SYSREF± edges do not deviate from the internal LMFC clock by more than a target amount.

Register 0x039 (SYSREF_ERR_WINDOW) indicates the size of the error window allowed, in DAC clock units. If a SYSREF± edge varies from the internal LMFC clock by more than the number of DAC clock units set in SYSREF_ERR_WINDOW, the IRQ_SYS-REF_JITTER is asserted.

*Table 20. SYSREF± Jitter Window Tolerance*

| SYSREF± Jitter Window Tolerance (DAC Clock Cycles) | SYSREF_ERR_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.

The IRQ_SYSREF_JITTER can be configured as described in the Interrupt Request Operation section to indicate the SYSREF± signal has varied, and to request the SPI sequence for a sync be performed again.

## Sync Procedure

The procedure for enabling the sync is as follows:

1. Set up the DAC and the SERDES PLL, and enable the CDR (see the Start-Up Sequence section).

## JESD204B SERIAL DATA INTERFACE

2. Set Register 0x03B to 0xF1 to enable the synchronization circuitry. If using the soft on/off feature, set Register 0x03B to Register 0xF3 to ramp the datapath data before and after the synchronization.
3. If Subclass 1, configure the SYSREF± settings as follows:
   a. Set Register 0x039 (SYSREF± jitter window). See Table 20 for settings.
   b. Set Register 0x036 = SYSREF_COUNT; leave as 0 to bypass.
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.
5. If Subclass 1, send a SYSREF± edge. If pulse counting, multiple SYSREF± edges are required. Sending SYSREF± edges triggers the synchronization.
6. Read back the SYNC_ROTATION_DONE bit (Register 0x03A, Bit 4) to confirm the rotation occurred.

### Resynchronizing LMFC Signals

If desired, the sync procedure can be repeated to realign the LMFC clock to the reference signal by repeating Step 2 to Step 6, described in the Sync Procedure section. When the one-shot sync is triggered (writing Register 0x03A = 0x02), the $\overline{\text{SYNCOUTx±}}$ signals deassert to drop the JESD204B links and reassert the links after the rotation completes.

### Deterministic Latency

JESD204B systems contain various clock domains distributed throughout. 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 AD9171 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 several PCLK cycles. It does not require any signal on the SYSREF± pins, which can be left disconnected.

Subclass 0 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 synchronized to within a few DAC clock cycles. Across the full operating range, for both supply and temperature, it is within ±2.5 DAC clock periods for a 6 GHz DAC clock rate or ±4 DAC clock periods for a 12.6 GHz DAC clock rate. If both supply and temperature stability are maintained, the link can be synchronized to within ±1.5 DAC clock periods for a 6 GHz DAC clock rate or ±2.5 DAC clock periods for a 12.6 GHz DAC clock rate. Achieving this latency requires an external, low jitter 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, as follows:

- The SYSREF± signal distribution skew within the system must be less than the desired uncertainty.
- The total latency variation across all lanes, links, and devices must be ≤12 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.

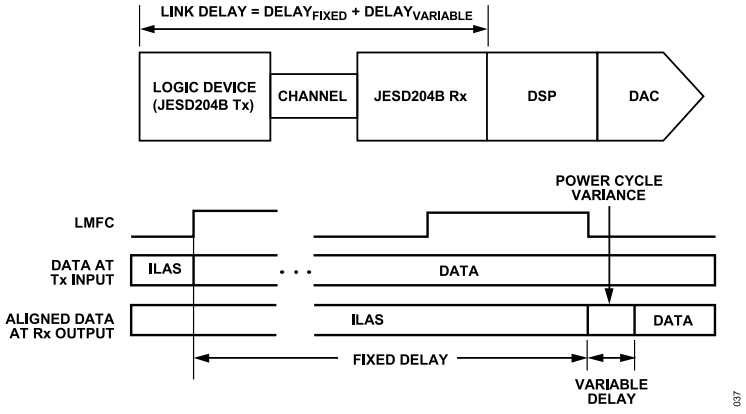


*Figure 36. JESD204B Link Delay = Fixed Delay + Variable Delay*

## JESD204B SERIAL DATA INTERFACE

### 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 36.

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 AD9171, this is not necessarily the case; instead, the AD9171 use 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 AD9171 can achieve proper performance with a smaller total latency. Figure 37 and Figure 38 show a case where the link delay is greater than an LMFC period. Note that it can be accommodated by delaying $LMFC_{Rx}$.

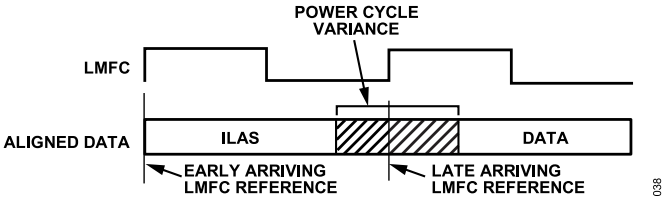


*Figure 37. Link Delay > LMFC Period Example*

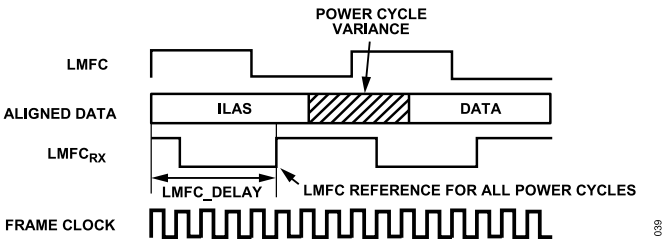


*Figure 38. LMFC_DELAY_x to Compensate for Link Delay > LMFC*

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 and the Link Delay Setup Example, Without Known Delay section. Note that the setting for LMFCDel must not equal or exceed the number of PCLK cycles per LMFC period in the current mode.

Similarly, LMFCVar must not exceed the number of PCLK cycles per LMFC period in the current mode or be set to <12 (whichever value is smaller).

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 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 one frame clock cycle to /K/ frame clock cycles, and the RBD of the AD9171 takes values from 0 PCLK cycle to 12 PCLK cycles. As a result, up to 12 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.

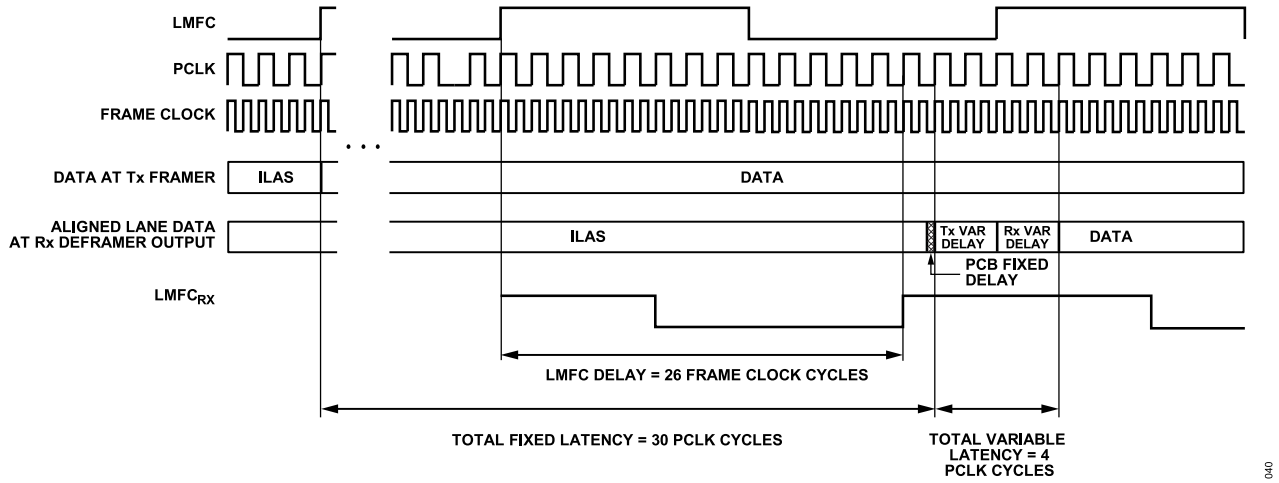


*Figure 39. LMFC Delay Calculation Example*

## JESD204B SERIAL DATA INTERFACE

### 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 39 is demonstrated in the following steps. Note that this example is in Subclass 1 to achieve deterministic latency, and the example uses the case for F = 2; therefore, the number of PCLK cycles per multiframe = 16. Because PCBFixed << PCLK Period, PCBFixed is negligible in this example and not included in the calculations.

1. Find the receiver delays using Table 5

   *RxFixed* = 13 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 field programmable gate array (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:

$$\begin{aligned} MinDelayLane &= \text{floor}(RxFixed + TxFixed + PCBFixed) \\ &= \text{floor}(13 + 13.5 + 0) \\ &= \text{floor}(26.5) \end{aligned}$$

$$MinDelayLane = 26$$

5. Calculate MaxDelayLane as follows:

$$\begin{aligned} MaxDelayLane &= \text{ceiling}(RxFixed + RxVar + TxFixed + \\ &\quad TxVar + PCBFixed) \\ &= \text{ceiling}(13 + 2 + 13.5 + 1 + 0) \\ &= \text{ceiling}(29.5) \end{aligned}$$

$$MaxDelayLane = 30$$

6. Calculate LMFCVar as follows:

$$\begin{aligned} LMFCVar &= (MaxDelay + 1) - (MinDelay - 1) \\ &= (30 + 1) - (26 - 1) = 31 - 25 \end{aligned}$$

$$LMFCVar = 6 \text{ PCLK cycles}$$

7. Calculate LMFCDel as follows:

$$\begin{aligned} LMFCDel &= (MinDelay - 1) \,\%\, (PCLKsperMF) \\ &= ((26 - 1)) \,\%\, 16 \\ &= 25 \,\%\, 16 \end{aligned}$$

$$LMFCDel = 9 \text{ PCLK cycles}$$

   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 AD9171 can read back the link latency between $LMFC_{RX}$ for each link (with the LMFCDel setting subtracted out) and the SYSREF± aligned LMFC. This information is then used to calculate LMFCVar and LMFCDel.

Figure 41 shows how DYN_LINK_LATENCY_0 (Register 0x302) provides a readback showing the delay (in PCLK cycles) between $LMFC_{RX}$ minus the LMFC_DELAY_x (fixed delay) setting set in the SPI at that time and the transition from the 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 41, for Link A, Link B, and Link C, the system containing the AD9171 (including the transmitter) is power cycled and configured 20 times. The AD9171 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 41, 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.

The example shown in Figure 41 is demonstrated in the following steps. Note that this example is in Subclass 1 to achieve deterministic latency, and the example uses the case for F = 1; therefore, the number of 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. Set LFMCVar to the maximum of 12 PCLK cycles. If latency is required to be minimized for a given application, calculate the total delay variation (with 2 PCLK cycles of guard band on each end) across all power cycles, links, and devices as follows:

$$\begin{aligned} LMFCVar &= (MaxDelay + 2) - (MinDelay - 2) \\ &= (9 + 2) - (4 - 2) = 11 - 2 = \text{PCLK cycles} \end{aligned}$$

4. Calculate the minimum delay in PCLK cycles (with 2 PCLK cycles of guard band) across all power cycles, links, and devices as follows:

## JESD204B SERIAL DATA INTERFACE

$$LMFCDel = (MinDelay - 2)\,\%\,(PCLKsperMF)$$
$$= (4 - 2)\,\%\,8$$
$$= 2\,\%\,8 = 2 \text{ PCLK cycles}$$

5. Write LMFCDel to Register 0x304 for all devices in the system. Write LMFCVar to Register 0x306 for all devices in the system.

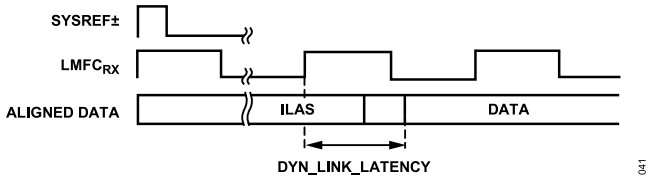


*Figure 40. DYN_LINK_LATENCY_x Illustration*

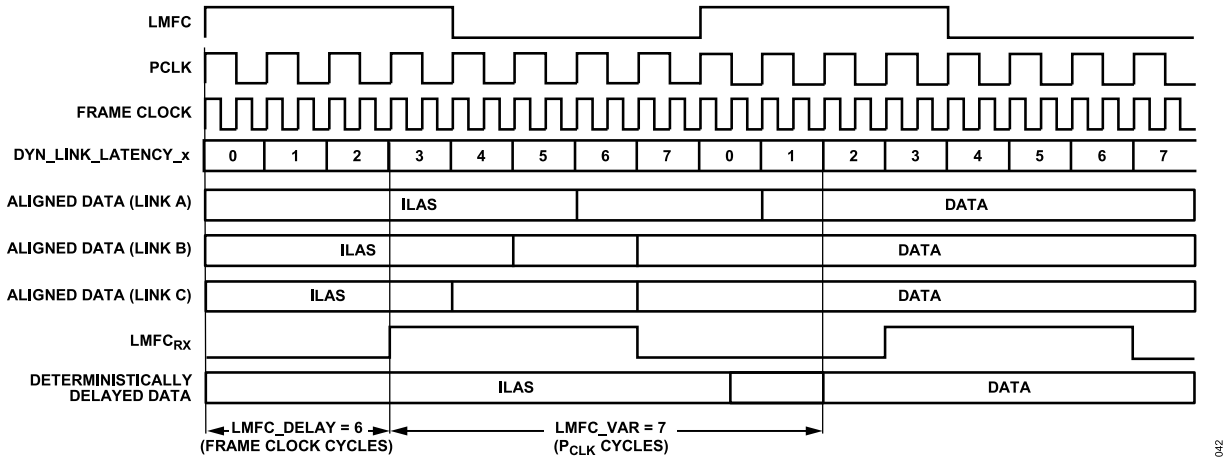


*Figure 41. Multilink Synchronization Settings, Derived Method Example*

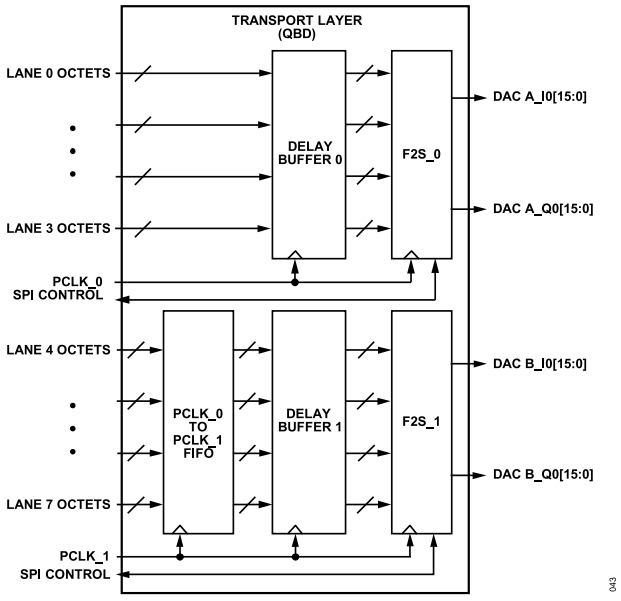


*Figure 42. Transport Layer Block Diagram*

## JESD204B SERIAL DATA INTERFACE

### TRANSPORT LAYER

The transport layer receives the descrambled JESD204B frames and converts them to DAC samples based on the programmed JESD204B parameters shown in Table 21. The device parameters are defined in Table 22.

*Table 21. JESD204B Transport Layer Parameters*

| Parameter | Description |
|---|---|
| F | Number of octets per frame per lane is 2, 3, or 4 |
| K | Number of frames per multiframe is 32 |
| L | Number of lanes per converter device (per link) is: 1 or 2 |
| M | Number of converters per device (per link), as follows: for complex data modes, M is the number of complex data subchannels |
| S | Number of samples per converter, per frame is 1 |

*Table 22. Table 22. 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. |

*Table 22. Table 22. JESD204B Device Parameters*

| Parameter | Description |
|---|---|
| HD | High density user data format. This parameter is always set to 1. |
| N | Converter resolution. |
| N'[1] | Total number of bits per sample. |

[1] N' can also be referred to as NP.

Certain combinations of these parameters are supported by the AD9171. See Table 24 for a list of supported single-link and dual-link modes, respectively. Table 24 lists the JESD204B parameters for each of the modes. Table 23 lists JESD204B parameters that have fixed values.

*Table 23. JESD204B Parameters with Fixed Values*

| Parameter | Value |
|---|---|
| K | 32 |
| CF | 0 |
| HD | 1 |
| CS | 0 |

*Table 24. Single-Link or Dual-Link JESD204B Operating Modes*

| | Single-Link or Dual-Link JESD204B Modes | | |
|---|---|---|---|
| Parameter | 0 | 3 | 5 |
| Lane Count (L) | 1 | 2 | 1 |
| Converter Count (M) | 2 | 2 | 2 |
| Octets per Frame per Lane (F) | 4 | 2 | 3 |
| Samples per Converter per Frame (S) | 1 | 1 | 1 |
| Total Number of Bits per Sample (NP) | 16 | 16 | 12 |
| Converter Resolution (N) | 16 | 16 | 12 |
| Frames per Multiframe (K) | 32 | 32 | 32 |
| High Density User Data Format (HD) | 1 | 1 | 1 |

## JESD204B SERIAL DATA INTERFACE

### Configuration Parameters

The AD9171 modes refer to the link configuration parameters for L, K, M, N, NP, S, and F. Table 25 provides the description and addresses for these settings.

*Table 25. 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. | 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] |

The AD9171 truncates the output of the main digital datapath to the value of N bits for the selected mode, which is then sent to the DAC core. It is possible to send the value of NP number of bits worth of data with the lower NP − N LSBs padded as 0s, or to send the full NP number of bits data across the SERDES lanes. In either case, the lower NP − N LSBs are truncated prior to the DAC core.

### 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 number of lanes for the mode setting chosen in Register 0x110, Bits[4:0]. 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.

The logical lanes must be enabled and deskewed on a per link basis using the LINK_PAGE control (Register 0x300, Bit 2). Set Bit x in Register 0x46C to 1 to deskew Link Logical Lane x for the selected link page.

## JESD204B TEST MODES

### PHY PRBS Testing

The JESD204B receiver on the AD9171 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 performed 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 AD9171 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 26.
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).
   b. 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 read-

ing the PHY_PRBS_ERR_CNT_xBITS (Register 0x31A to Register 0x31C). The maximum error count is $2^{24} - 1$. If all bits of Register 0x31A to Register 0x31C are high, the maximum error count on the selected lane is exceeded.

*Table 26. 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 AD9171 supports the short transport layer (STPL) test as described in the JESD204B standard. Use this test to verify the data mapping between the JESD204B transmitter and receiver. To perform this test, this function must be implemented and enabled in the logic device. 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 the test samples that 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 AD9171 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 six complex subchannels (M = 6), and each frame may contain up to eight samples (S = 8). Configure the SHORT_TPL_ REF_SP_MSB bits (Register 0x32E) and the SHORT_TPL_ REF_SP_LSB bits (Register 0x32D) to match one of the samples for one converter within one frame. For N = 12 modes, the expected sample value is multiplied by 16.
4. Set SHORT_TPL_M_SEL (Register 0x32C, Bits[3:2]) to select the channel.
5. Set SHORT_TPL_IQ_SAMPLE_SEL (Register 0x32F, Bit 6) to select the I or Q subchannel.
6. Set SHORT_TPL_SP_SEL (Register 0x32C, Bits[7:4]) to select the sample within one frame for the selected.
7. Set SHORT_TPL_TEST_EN (Register 0x32C, Bit 0) to 1.
8. Set SHORT_TPL_TEST_RESET (Register 0x32C, Bit 1) to 1, then back to 0.
9. 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.
10. Read the test result at SHORT_TPL_FAIL (Register 0x32F, Bit 0).
11. Choose another sample for the same or another M to continue with the test, until all samples for both converters from one frame are verified.

## Internal Loop Back Test

The AD9171 integrates one internal PRBS generator that can be used to test the JESD204B PHYs without an external SERDES signal input. The process for performing internal loopback testing on the AD9171 is as follows:

1. Set the EQ_BOOST_PHYx bits (Register 0x240, Bits[7:0] and Register 0x241, Bit[7:0]) to 0.
2. Set SEL_IF_PARDATAINV_DES_RC_CH bits (Register 0x234, Bits[7:0]) to 0 to make sure lanes not inverted.
3. Enable the loop back test for all lanes being tested by writing to EN_LBT_DES_RC_CH (Register 0x250). Each bit of Register 0x250 enables the loop back test for the corresponding lane. For example, writing a 1 to Bit 0 enables the test for Physical Lane 0.
4. For half rate, set EN_LBT_HALFRATE_DES_RC (Register 0x251, Bit 1) to 1; otherwise, set it to 0.
5. Toggle INIT_LBT_SYNC_DES_RC (Register 0x251, Bit 0) from 0 to 1 then back to 0.
6. Refer to the PHY PRBS Testing section for information on how to run a PRBS7 check.

## Repeated CGS and ILAS Test

As per Section 5.3.3.8.2 of the JESD204B specification, the AD9171 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 AD9171 SERDES inputs. Next, set up the device and enable the links. Ensure that the /K28.5/ characters are being received by verifying that $\overline{\text{SYNCOUT}}$± is deasserted and that CGS passes 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, and enable the links. When the device recognizes four CGS characters on each lane, it deasserts the $\overline{\text{SYNCOUT}}$x±. At this point, the transmitter starts sending a repeated ILAS sequence.

Read Register 0x473 to verify that the 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 AD9171 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.

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.

### 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 Table 55. 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 1 to the corresponding bit to reset that error counter.
3. Register 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, $\overline{\text{SYNCOUTx±}}$ 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 $\overline{\text{SYNCOUTx±}}$ 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 $\overline{\text{SYNCOUTx±}}$ (Register 0x47D, Bits[2:0] = 0b111). When setting the SYNC_ASSERT_MASK bits, LINK_PAGE (Register 0x300, Bit 2) must be set to 1.
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 is reached. These status bits are located in Register 0x490, Bits[2:0] to Register 0x497, Bits[2:0]. Bit 3 can be read back to indicate whether a particular lane is active.

### Error Counter and IRQ Control

For error counter and IRQ control, follow these steps:

1. Enable the interrupts. 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 $\overline{\text{SYNCOUTx±}}$

When one or more disparity, NIT, or unexpected control character errors occur, the error is reported on the $\overline{\text{SYNCOUTx±}}$ pin as per Section 7.6 of the JESD204B specification. The JESD204B specification states that the $\overline{\text{SYNCOUTx±}}$ signal is asserted for exactly two frame periods when an error occurs. For the AD9171, the width of the $\overline{\text{SYNCOUTx±}}$ pulse can be programmed to ½, 1, or 2 PCLK cycles. The settings to achieve a $\overline{\text{SYNCOUTx±}}$ pulse of two frame clock cycles are given in Table 27.

***Table 27. Setting $\overline{\text{SYNCOUTx±}}$ 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 |
| 3 | 1.5 | 2 |
| 4 | 1 | 2 |
| 8 | 0.5 | 4 |

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

## JESD204B SERIAL DATA INTERFACE

### 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 or four NIT characters are received as per Section 7.1 of the JESD204B specification. When a link reinitialization occurs, the resync request is at least 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 28. These are enabled by default.
2. Write a 0 to the corresponding bit to Register 0x480, Bits[2:0] to Register 0x487, Bits[2:0] to take counter out of reset.
3. Enable the sync assertion mask for each type of error by writing to SYNC_ASSERT_MASK (Register 0x47D, Bits[2:0]) according to Table 28.
4. Program the desired error counter threshold into ERRORTHRES (Register 0x47C).
5. For each error type enabled in the SYNC_ASSERT_MASK register, if the error counter on any lane reaches the programmed threshold, $\overline{\text{SYNCOUTx±}}$ 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 28. Sync Assertion Mask (SYNC_ASSERT_MASK)*

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

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 AD9171 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 settings received by the device (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.

## DIGITAL DATAPATH

The full digital datapath of the AD9171 consists of channel datapaths that are bypassable, as well as a main DAC datapath leading up to the analog DAC core. There is a variety of different digital feature blocks available within each of the channel and main DAC datapaths.

### TOTAL DATAPATH INTERPOLATION

The AD9171 contains two separate blocks of interpolation: one located in each of the channelizer datapaths, and a second interpolation stage located in the main DAC datapath. The total interpolation for the chip can be determined by multiplying the channel interpolation factor by the main datapath interpolation factor. The relationship between the DAC sample rate and input data rate is shown in the following equation:

$$TotalInterpolation = (ChannelInterpolation \times MainInterpolation)$$

$$f_{DATA} = f_{DAC} / (ChannelInterpolation \times MainInterpolation)$$

Each of the various cascaded half-band filters has 80% signal bandwidth capabilities. Therefore, if using interpolation (not bypass or 1× total interpolation for both channel and main datapaths), the available signal bandwidth (BW) with respect to the input data rate is calculated as follows:

$$SignalBW = 0.8 \times (f_{DATA}/2)$$

where the total interpolation > 1.

$$SignalBW = (f_{DATA}/2)$$

where the total interpolation = 1.

The interpolation values are programmed as shown in Table 29.

*Table 29. Interpolation Factor Register Settings*

| Interpolation Factor | Main Datapath, Register 0x111, Bits[7:4] | Channel Datapath, Register 0x111, Bits[3:0] |
|---|---|---|
| 2× | Not applicable | 0x2 |
| 3× | Not applicable | 0x3 |
| 4× | Not applicable | 0x4 |
| 6× | 0x6 | 0x6 |
| 8× | 0x8 | Not applicable |

*Table 30. Interpolation Modes and Useable Bandwidth*

| Total Interpolation | Available Signal Bandwidth | $f_{DATA}$ |
|---|---|---|
| 12×, 16×, 18×, 24×, 32×, 36×, 48×, 64× | 80% × $f_{DATA}$ /2 | $f_{DAC}$/total interpolation |

### 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.

The usable bandwidth, as shown in Table 30 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.

The maximum pass-band amplitude of all filters is the same; they are different in the illustration to improve understanding.

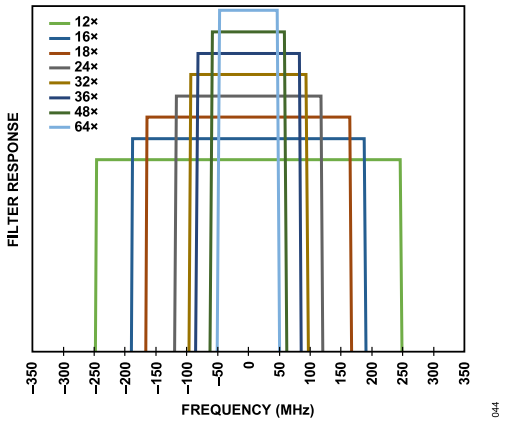


*Figure 43. Band Responses of Total Interpolation Rates for 8×, 12×, 16×, 18×, 24×, 32×, 36×, 48×, and 64× at a 6 GHz DAC Rate, 16-Bit*

### NCO Only Mode

The AD9171 is capable of operating in a mode with only the NCO enabled. In this mode, a single-tone sine wave is generated using a programmable internal dc amplitude level that is injected into the NCO block to modulate the data into a single tone. This internal dc level is injected into either the channel or main datapaths through various controls, as shown in Figure 44. Mode 3 can be used to generate a single-channel NCO only mode of operation. It is not necessary to bring up the JESD204B link in this mode. This mode is a useful option to bring up a transmitter radio signal chain without requiring a digital data source, because the device generates the NCO data internally.

This mode can also be used in applications where a sine wave is all that is required, such as in a local oscillator application. There is an additional optional calibration NCO block that can be used as part of the initial system calibration without needing to reprogram the final main datapath NCO to the final system configuration. This feature is discussed in more detail in the Calibration NCO section.

When the NCO test mode is enabled, the data source of the digital datapath is the dc test data word, meaning that the JESD204B link can be brought up and data can be transferred to the device over the link, but the data is not presented to the DAC when in this test mode. Connection to the SERDES data source is only achieved by disabling this test mode. The SPI control bit for this test mode can be set on the fly, but because disabling the mode and switching to the SERDES datapath normally requires the lanes and/or interpolation mode to also be set, on the fly setting or resetting of the NCO only mode control bit is normally not practical.

## DIGITAL DATAPATH

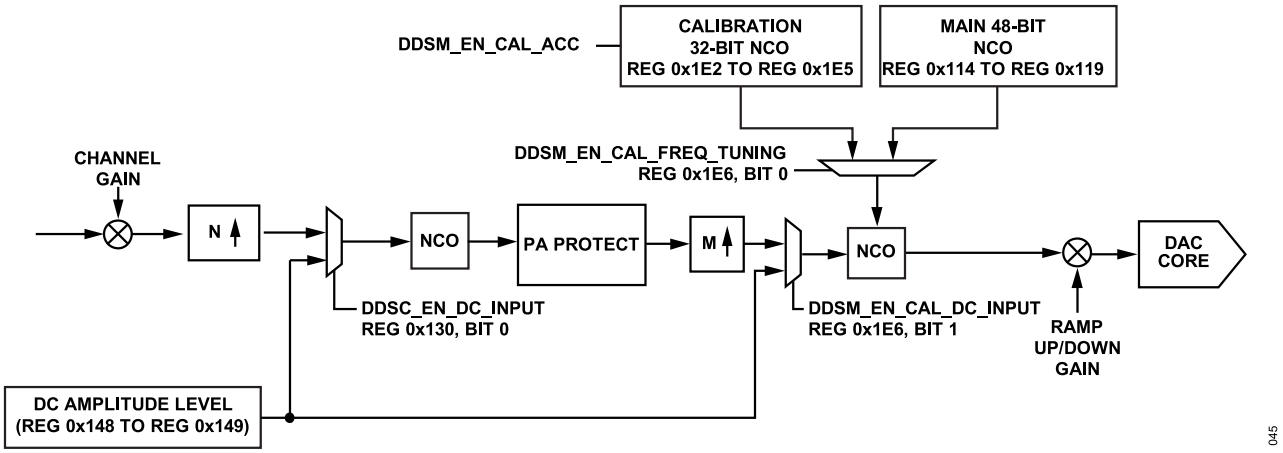


*Figure 44. DC Amplitude Injection for NCO Only Mode Block Diagram*

## CHANNEL DIGITAL DATAPATH

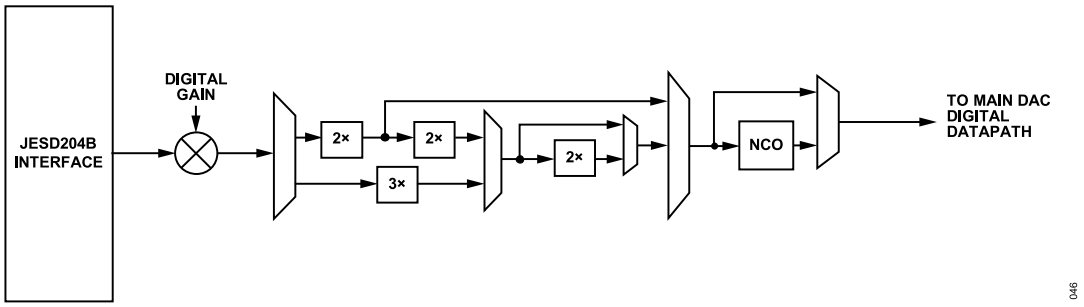


*Figure 45. Block Diagram of the Channel Digital Datapath per the Main DAC Output*

Each main DAC output has one channelizer, or channel datapath, that is selected based on the JESD204B mode selected. Each channelizer consists of a digital gain stage, complex interpolation block, and complex 48-bit NCO with modulus. The interpolation selection is the same for all the channelizers; however, the gain stage and complex NCO values can all be independently configured. The controls for these blocks are paged by the channel paging mask in the CHANNEL_PAGE bits (Register 0x008, Bit 0 and Bit 3) as described in Table 31. Each bit of the page mask corresponds to a channel datapath. These channels can be paged individually to set the values for each channel uniquely, or can be paged simultaneously to set all channels to the same value for the control being configured.

*Table 31. Channel Page Mask*

| CHANNEL_PAGE (Register 0x008, Bits[5:0]) | Channel Paged | Channel Datapath Updated |
|---|---|---|
| 0x01 (Bit 0) | Channel 0 | Channel for DAC0 |
| 0x08 (Bit 3) | Channel 1 | Channel for DAC1 |

Each of the digital blocks in the channels is described in more detail in the following sections.

### Digital Gain

The AD9171 has individual channel gain controls that allow unique gain scaling capabilities for each complex data input channel.

The gain code for each channel is 12-bit resolution, located in Register 0x146 and Register 0x147, and can be calculated by the following formula:

$$0 \le Gain \le (2^{12} - 1)/2^{11}$$

$$-\infty\ dB < dBGain \le +6.018\ dB$$

$$Gain = Gain\ Code \times (1/2048)$$

$$dBGain = 20 \times log_{10}(Gain)$$

$$Gain\ Code = 2048 \times Gain = 2^{11} \times 10^{(dBGain/20)}$$

The gain code control (CHNL_GAIN) is paged with the channel page mask (CHANNEL_PAGE) in Register 0x008, Bits[5:0]. The digital gain feature is available in all modes, except when 1× channel interpolation is used because the channel datapaths are bypassed in that mode. The summing node that combines the three channels may clip the summed signals if the combined amplitude is too large. For example, if all three channels are being used and all three data streams are >1/3 full-scale amplitude, clipping may

## DIGITAL DATAPATH

occur. The sum of the data values when any number of channels is used must be between $-2^{15}$ and $+2^{15}$.

### Channel Interpolation

The channel interpolation options available are bypass (1×), 2×, 3×, 4×, 6×, and 8×. Each of the half-band filters used for interpolation has 80% bandwidths with 85 dB of stop band rejection. The channel half-band cascaded configuration is shown in Figure 46, with each of the useable bandwidths of the channel interpolation filters listed in Table 32.

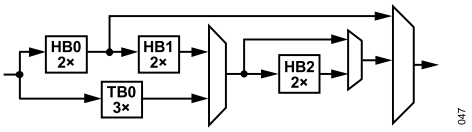


*Figure 46. Channel Interpolation Half-Band Filter Block Diagram*

*Table 32. Channel Interpolation Useable Bandwidths and Rejection*

| Half-Band Filter | Bandwidth (×$f_{IN_FILTER}$[1]) (%) | Stop Band Rejection (dB) |
|---|---|---|
| HB0 | 80 | 85 |
| TB0 | 54 | 85 |
| HB1 | 40 | 85 |
| HB2 | 27 | 85 |

[1] $f_{IN_FILTER}$ is the frequency at the input of the half-band filter.

### Channel Digital Modulation

The AD9171 has digital modulation features to modulate the baseband quadrature signal to a desired frequency. There are two stages of complex digital modulation available in the AD9171: channel modulation and main modulation. Each main DAC has three channels, each with its own NCO, that can individually modulate the data from each channel with a unique frequency and phase offset. The AD9171 is equipped with several NCO modes. The default NCO is a 48-bit, integer NCO. There is an additional modulus option for each channel NCO where 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 33. These controls are paged per the channel page masks in the CHANNEL_PAGE bits (Register 0x008, Bits[5:0]).

*Table 33. Channel Modulation Mode Selection*

| | Modulation Type | |
|---|---|---|
| Modulation Mode | Register 0x130, Bit 6 | Register 0x130, Bit 2 |
| None | 0b0 | 0b0 |
| 48-Bit Integer NCO | 0b1 | 0b0 |
| 48-Bit Dual Modulus NCO | 0b1 | 0b1 |

The channel NCO blocks also contain sideband selection controls as well as options for how the FTW and phase offset controls are updated.

Calculate the phase offset word control as follows:

$$-180° \le Degrees\ Offset \le +180°$$

$$Degrees\ Offset = 180° \times (DDSC_NCO_PHASE_OFFSET/2^{15})$$

where *DDSC_NCO_PHASE_OFFSET* is a 16-bit twos complement value programmed in the registers listed in Table 34.

*Table 34. Channel NCO Phase Offset Registers*

| Address | Value | Description |
|---|---|---|
| 0x138 | DDSC_NCO_PHASE_OFFSET[7:0] | 8 LSBs of phase offset |
| 0x139 | DDSC_NCO_PHASE_OFFSET[15:8] | 8 MSBs of phase offset |

### 48-Bit Dual Modulus NCO

The 48-bit dual modulation mode uses an NCO, a phase shifter, and a complex modulator to modulate the signal by a programable carrier signal, as shown in Figure 47. This configuration allows output signals to be placed anywhere in the output spectrum up to $\pm f_{NCO}/2$ 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 47.

### Integer NCO Mode

The channel 48-bit NCOs can be used as integer NCOs. The FTW for the channel NCOs depends on the speed at which the channel NCO block is running ($f_{NCO}$), and can be calculated by using the following formulas:

$$f_{NCO} = f_{DATA} \times Channel\ Interpolation$$

or

$$f_{NCO} = f_{DAC}/Main\ Interpolation$$

The FTWs for each individual channel can be programmed separately and are calculated by using the following formula:

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

$$DDSC_FTW = (f_{CARRIER}/f_{NCO}) \times 2^{48}$$

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

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

*Table 35. Channel NCO FTW Registers*

| Address | Value | Description |
|---|---|---|
| 0x132 | DDSC_FTW[7:0] | 8 LSBs of FTW |
| 0x133 | DDSC_FTW[15:8] | Next 8 bits of FTW |
| 0x134 | DDSC_FTW[23:16] | Next 8 bits of FTW |

## DIGITAL DATAPATH

*Table 35. Channel NCO FTW Registers*

| Address | Value | Description |
|---|---|---|
| 0x135 | DDSC_FTW[31:24] | Next 8 bits of FTW |
| 0x136 | DDSC_FTW[39:32] | Next 8 bits of FTW |
| 0x137 | DDSC_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 DDSC_FTW_LOAD_REQ (Register 0x131, Bit 0). After an update request, DDSC_FTW_LOAD_ACK (Register 0x131, Bit 1) must be high to acknowledge that the FTW has updated.

The DDSC_SEL_SIDEBAND bit (Register 0x130, 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.

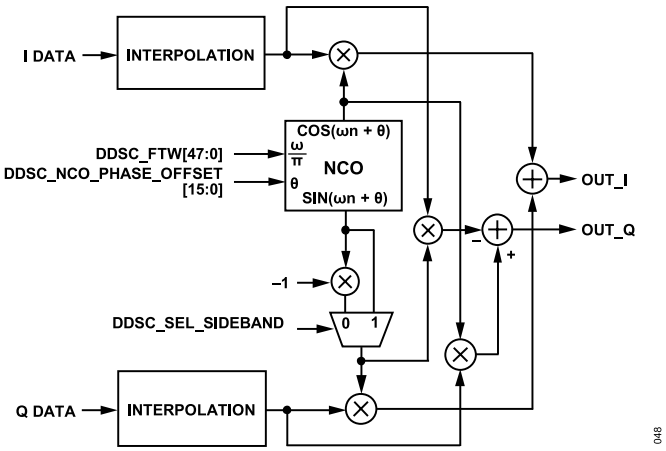


*Figure 47. NCO Modulator Block Diagram*

### Channel Modulus NCO Mode (Direct Digital Synthesis (DDS))

Each of the channel 48-bit NCOs 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 DDSC_MODULUS_EN bit in the DDSC_DATAPATH_CFG register to 1 (Register 0x130, 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 AD9171 is such that the fraction, M/N, is expressible by the following equation. 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.

$$\frac{f_{CARRIER}}{f_{NCO}} = \frac{M}{N} = \frac{X + \frac{A}{B}}{2^{48}}$$

where:
*X* is programmed in Register 0x132 to Register 0x137.
*A* is programmed in Register 0x140 to Register 0x145.
*B* is programmed in Register 0x13A to Register 0x13F.

### Programmable Modulus Example

Consider the case in which $f_{NCO}$ = 500 MHz and the desired value of $f_{CARRIER}$ is 50 MHz. This scenario synthesizes an output frequency that is not a power of two submultiple of the sample rate, namely $f_{CARRIER}$ = (1/10) $f_{NCO}$, which is not possible with a typical accumulator-based DDS. The frequency ratio, $f_{CARRIER}/f_{NCO}$, leads directly to M and N, which are determined by reducing the fraction (50,000,000/500,000,000) to its lowest terms, that is,

*M*/*N* = 50,000,000/500,000,000 = 1/10

Therefore, M = 1 and N = 10.

After calculation, X = 28,147,497,671,065, A = 3, and B = 5. Programming these values into the registers for X, A, and B (X is programmed in Register 0x132 to Register 0x137 for DDSC_FTWx, B is programmed in Register 0x13A to Register 0x13F for DDSC_ACC_MODULUSx, and A is programmed in Register 0x140 to Register 0x145 for DDSC_ACC_DELTAx) causes the NCO to produce an output frequency of exactly 50 MHz given a 500 MHz sampling clock. For more details, refer to the AN-953 Application Note.

### 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 or by the SYSREF± signal.

## DIGITAL DATAPATH

### MAIN DIGITAL DATAPATH

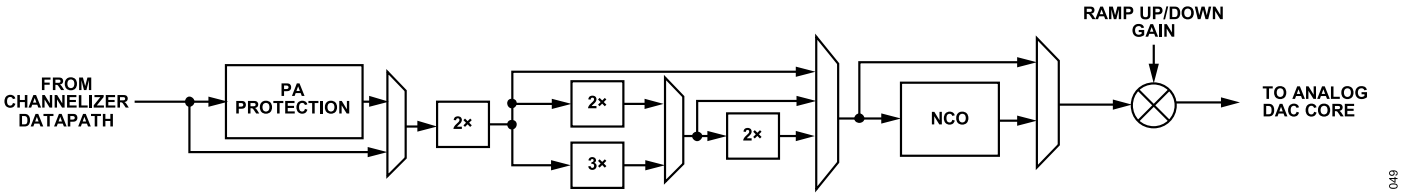


*Figure 48. Block Diagram of the Main Digital Datapath per Main DAC Output*

Each main DAC digital datapath consists of a power amplifier (PA) protection block, a complex main datapath interpolation block, a 48-bit complex main datapath NCO, and a ramp-up/ ramp-down gain block. The interpolation selection is set the same for all main datapaths; however, the PA protection block and complex NCO values can all be independently configured.

The controls for these blocks are paged by the main DAC datapath paging mask, MAINDAC_PAGE (Register 0x008, Bits[7:6]), as listed in Table 36. Each bit of the page mask corresponds to a main DAC datapath. These datapaths can be paged individually to set the values for each DAC uniquely or can be paged simultaneously to set all channels to the same value for the control being configured.

*Table 36. Main DAC Datapath Page Mask*

| MAINDAC_PAGE (Register 0x008, Bits[7:6]) | DAC Paged | DAC Datapath Updated |
|---|---|---|
| 0x40 (Bit 6) | DAC0 | DAC0 |
| 0x80 (Bit 7) | DAC1 | DAC1 |

Each of the digital blocks in the main datapaths is described in more detail in the following sections.

#### Downstream Protection (PA Protection)

The AD9171 has several blocks designed to protect the PA of the system, as well as other downstream blocks, by preventing transients out of the DAC outputs. The DAC output can be triggered to turn on and off by the following signals, shown in Figure 49:

- PDP_PROTECT. This signal asserts when the calculated digital vector power exceeds a programmable threshold.
- INTERFACE_PROTECT. This signal asserts when certain JESD204B errors occur.
- SPI_PROTECT. This signal asserts when the user writes the SPI trigger register control.
- BSM_PROTECT. This signal asserts and the blanking state machine (BSM) module flushes the datapath on the rising edge of the TXEN0 or TXEN1 signals, which may come from a SPI write or the external TXEN0 or TXEN1 pin.

The flags of these events can also be routed out on the pins of IRQ ($\overline{IRQ0}$ and $\overline{IRQ1}$) to shut down other external downstream components. The DAC output on/off is implemented through a feedforward trigger signal to the ramp-up/ramp-down digital gain block at the end of the main digital datapath, before the analog DAC core, which allows the DAC to be turned on or off gradually with the ramp on/off block.

## DIGITAL DATAPATH

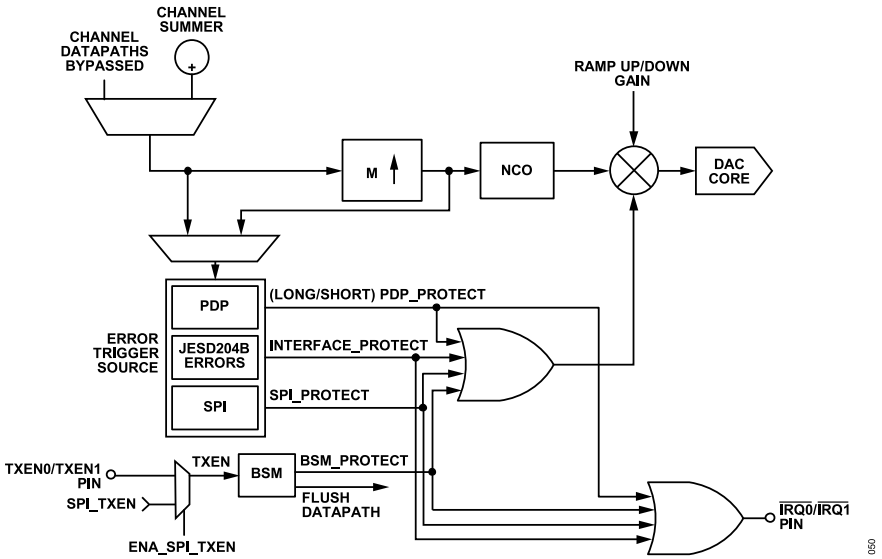


*Figure 49. Block Diagram of Downstream Protection Triggers*

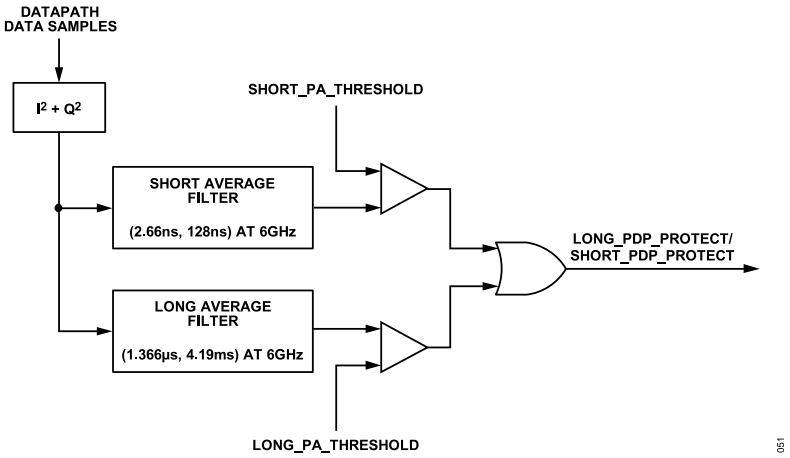


*Figure 50. PDP Block Diagram*

### Power Detection and Protection (PDP) Block

The PDP block detects the average power of the DAC input signal and prevents overrange signals from being passed to the next stage, which may potentially cause destructive breakdown on power sensitive devices, such as PAs. The protection function provides a signal, PDP_PROTECT, which can shut down the DAC outputs or be routed externally to shut down a PA.

The PDP block uses a separate path with a shorter latency than the datapath to ensure that PDP_PROTECT is triggered before the overrange signal reaches the analog DAC cores. The sum of the $I^2$ and $Q^2$ are calculated as a representation of the input signal power (only the top six MSBs of data samples are used). The calculated sample power numbers are accumulated through a moving average filter with an output that is the average of the input signal power in a certain number of samples.

There are two types of average filters with different lengths: the short filter detects pulses with high power in the digital domain and the long filter detects high power in the digital domain, which lasts longer than the thermal constant of the PA.

When the output of the averaging filter is larger than the threshold, the internal signal, PDP_PROTECT, goes high, which can optionally be configured to trigger an IRQ flag and turn off the DAC output through the ramp-up/ramp-down.

The PDP block is configured as shown in Figure 50.

The long and short averaging time is configured by the LONG_PA_AVG_TIME (Register 0x585, Bits [3:0]) and the SHORT_PA_AVG_TIME (Register 0x58A, Bits [1:0]) controls. Use the following calculations to determine the average window size times, in units of PA clock periods:

$$Length\ of\ Long\ Average\ Window = 2^{LONG_PP_AVG_TIME + 9}$$

## DIGITAL DATAPATH

$$Length\ of\ Short\ Average\ Window = 2^{SHORT_PA_AVG_TIME}$$

When the average calculation value exceeds a specified threshold, the ramp-down signal is triggered to ramp down the output. The thresholds for the long and short average options are programmed in the registers listed in Table 37, along with their respective detected power calculation readbacks.

*Table 37. PDP Threshold and Power Calculation Controls*

| Register | Bits | Control |
|---|---|---|
| 0x583 | [7:0] | LONG_PA_THRESHOLD[7:0] |
| 0x584 | [4:0] | LONG_PA_THRESHOLD[12:8] |
| 0x586 | [7:0] | LONG_PA_POWER[7:0] |
| 0x587 | [4:0] | LONG_PA_POWER[12:8] |
| 0x588 | [7:0] | SHORT_PA_THRESHOLD[7:0] |
| 0x589 | [4:0] | SHORT_PA_THRESHOLD[12:8] |
| 0x58B | [7:0] | SHORT_PA_POWER[7:0] |
| 0x58C | [4:0] | SHORT_PA_POWER[12:8] |

### Main Datapath Interpolation

The main digital datapath interpolation options available are 6× and 8×. Each of the half-band filters used for interpolation have 80% bandwidths with 85 dB of stop band rejection. The channel half-band cascaded configuration is shown in Figure 51, with each of the useable bandwidths of the channel interpolation filters listed in Table 38.

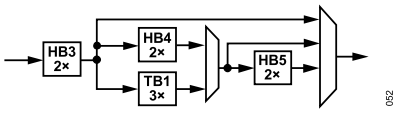


*Figure 51. Main Datapath Interpolation Half-Band Filter Block Diagram*

*Table 38. Main Datapath Interpolation Useable Bandwidths and Rejection*

| Half-Band Filter | Bandwidth ($\times f_{IN_FILTER}$) | Stop Band Rejection (dB) |
|---|---|---|
| HB3 | 80% | 85 |
| HB4 | 40% | 85 |
| TB1 | 27% | 85 |
| HB5 | 20% | 85 |

### Main Digital Modulation

The AD9171 has digital modulation features to modulate the baseband quadrature signal to a desired frequency. There are two stages of complex digital modulation available in the AD9171; channel modulation and main modulation. Each main DAC has a final main NCO that can individually modulate each main DAC datapath with a unique frequency and phase offset. The AD9171 is equipped with several NCO modes. The default NCO is a 48-bit, integer NCO. There is an additional modulus option for each main DAC datapath NCO where 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 39. These controls are paged per the main DAC page masks, MAINDAC_PAGE (Register 0x008, Bits [7:6]).

*Table 39. Main Modulation Mode Selection*

| | Modulation Type | |
|---|---|---|
| Modulation Mode | Register 0x112, Bit 3 | Register 0x112, Bit 2 |
| None | 0b0 | 0b0 |
| 48-Bit Integer NCO | 0b1 | 0b0 |
| 48-Bit Dual Modulus NCO | 0b1 | 0b1 |

The main NCO blocks also contain sideband selection controls as well as options for how the FTW and phase offset controls are updated.

Calculate the phase offset word control as follows:

$$-180° \le Degrees\ Offset \le +180°$$

$$Degrees\ Offset = 180° \times (DDSM_NCO_PHASE_OFFSET/2^{15})$$

where *DDSM_NCO_PHASE_OFFSET* is a 16-bit twos complement value programmed in the registers listed in Table 40.

*Table 40. Main Datapath NCO Phase Offset Registers*

| Address | Value | Description |
|---|---|---|
| 0x11C | DDSM_NCO_PHASE_OFFSET[7:0] | 8 LSBs of phase offset |
| 0x11D | DDSM_NCO_PHASE_OFFSET [15:8] | 8 MSBs of phase offset |

### 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 47. This configuration allows output signals to be placed anywhere in the output spectrum up to $\pm f_{NCO}/2$ 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 47.

### Integer NCO Mode

The channel 48-bit NCOs can be used as integer NCOs. The FTW for the channel NCOs depends on the speed at which the main NCO block is running ($f_{DAC}$), which is the same rate as the DAC sample rate (6.2 GSPS, maximum) and can be calculated by using the following formula:

$$f_{DAC} = f_{DATA} \times Channel\ Interpolation \times Main\ Interpolation$$

The FTWs for each individual channel can be programmed separately and are calculated using the following formula:

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

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

## DIGITAL DATAPATH

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

The frequency tuning word is set as shown in Table 41.

***Table 41. Main Datapath NCO FTW Registers***

| Address | Value | Description |
|---|---|---|
| 0x114 | DDSM_FTW[7:0] | 8 LSBs of FTW |
| 0x115 | DDSM_FTW[15:8] | Next 8 bits of FTW |
| 0x116 | DDSM_FTW[23:16] | Next 8 bits of FTW |
| 0x117 | DDSM_FTW[31:24] | Next 8 bits of FTW |
| 0x118 | DDSM_FTW[39:32] | Next 8 bits of FTW |
| 0x119 | DDSM_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 DDSM_FTW_LOAD_REQ (Register 0x113, Bit 0). After an update request, DDSM_FTW_LOAD_ACK (Register 0x113, Bit 1) must be high to acknowledge that the FTW updated.

The DDSM_SEL_SIDEBAND bit (Register 0x112, 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.

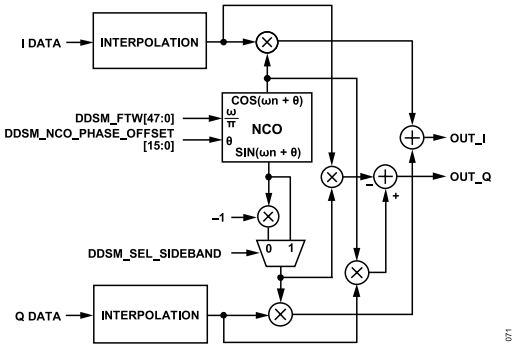


***Figure 52. NCO Modulator Block Diagram***

### Main Datapath Modulus NCO Mode (Direct Digital Synthesis (DDS))

Each of the main datapath 48-bit NCOs 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 DDSM_MODULUS_EN bit in the DDSM_DATAPATH_ CFG register to 1 (Register 0x112, Bit 2 = 0b1).

The frequency ratio for the programmable modulus 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 AD9171 is such that the fraction, M/N, is expressible by the following equation. 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.

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

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 $f_{DAC}$ = 6 GHz and the desired value of $f_{CARRIER}$ is 600 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 (600,000,000/6,000,000,000) to its lowest terms, that is,

*M*/*N* = 600,000,000/6,000,000,000 = 1/10

Therefore, M = 1 and N = 10.

After calculation, X = 28,147,497,671,065, 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 for DDSM_FTWx, B is programmed in Register 0x124 to Register 0x129 for DDSM_ACC_MODULUSx, and A is programmed in Register 0x12A to Register 0x12F for DDSM_ACC_DELTAx) causes the NCO to produce an output frequency of exactly 600 MHz given a 6 GHz sampling clock. For more details, refer to the AN-953 Application Note.

### 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 via an SPI write or by the SYSREF± signal.

### Calibration NCO

There is an additional 32-bit calibration NCO option as part of the main DAC NCO block, shown in Figure 44. Register 0x1E6, Bit 0 controls whether the 32-bit calibration NCO is in use in the main datapath, or if the normal 48-bit main NCO is used. First, enable the calibration NCO accumulator by setting Register 0x1E6, Bit 2 = 1. Then, program the calibration NCO FTW in Register 0x1E2 to Register 0x1E5 and update the FTW to take effect by toggling Register 0x113, Bit 0 from 0 to 1. Select the calibration NCO to be used instead of the main NCO by setting Register 0x1E6, Bit 0 = 1.

## DIGITAL DATAPATH

If using the calibration NCO along with the internal dc amplitude level, enable the dc amplitude injection feature by setting Register 0x1E6, Bit 1 = 1.

### Modulator Switch

The final DAC NCO block has an added functionality of a modulator switch that allows the user to select the desired outputs of the NCO block. There are four mode functionalities available, as shown in Figure 53 to Figure 56.

Some modes bypass the main datapath NCO for each DAC and use the complex I and Q data from each datapath, whereas other modes only use the I data output from each of the main datapath NCO blocks and select which DAC is used to output the data. These modes are programmed in Register 0x112, Bits[5:4] and are paged by the MAINDAC_PAGE register control.

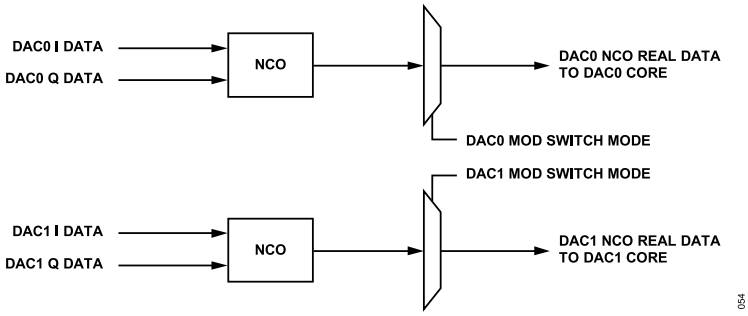


*Figure 53. Mode 0—DAC0 = I0, DAC1 = I1*

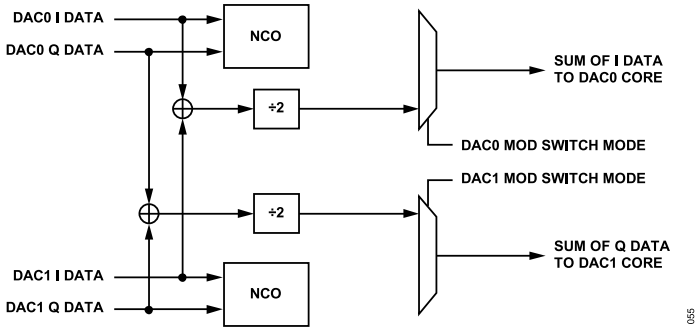


*Figure 54. Mode 1—DAC0 = I0 + I1, DAC1 = Q0 + Q1*

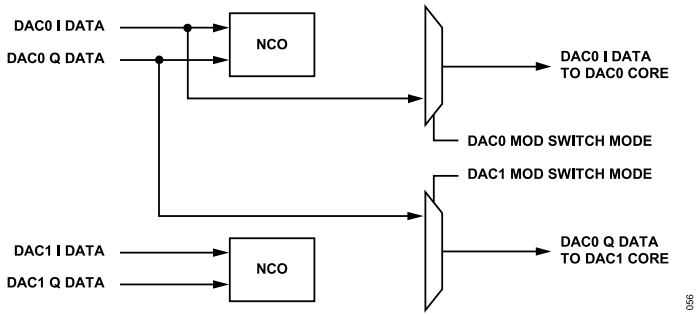


*Figure 55. Mode 2—DAC0 = I0, DAC1 = Q0*

## DIGITAL DATAPATH

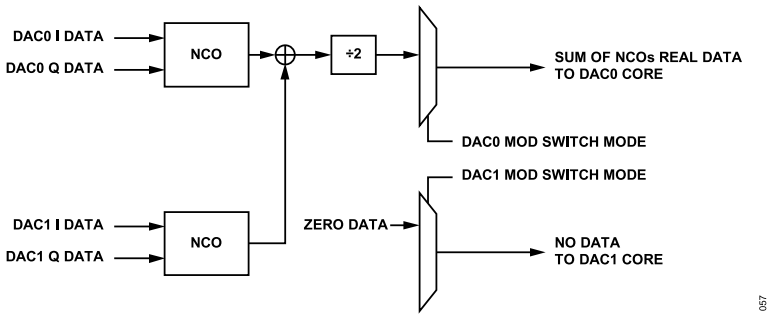


*Figure 56. Mode 3—DAC0 = I0 + I1, DAC1 = 0*

### Ramp Up/Down Gain

In each of the main datapaths, the ramp-up and ramp-down gain block is immediately before the digital datapath enters the analog DAC core for decoding. This block works in conjunction with the power detection and protection (PDP) block to protect any down-stream components from large power peaks or long, high average power.

Various trigger signals can be configured in the PA protection block to trigger a gain ramp-down to mute the data being trans-mitted out of the DAC, as shown in Figure 49. The ramp-up and ramp-down steps can be configured via the SPI in Register 0x580, Bits[2:0]. The equation for the ramp-up and ramp-down occurs in 32 steps over $2^{(CODE + 8)}$ DAC clock periods. This control can be configured individually for each of the DAC ramp blocks via the MAINDAC_PAGE control in Register 0x008.

After the data is ramped down from one of the triggers, it can be ramped back up in two of the following ways after the error condition is cleared. If the SPI protection control bit triggered the interrupt for a ramp-down, the SPI is then also used to ramp the data back up by toggling Register 0x582, Bit 7 from 0 to 1, and then back to 0.

Additionally, an option exists to mute the digital data during a digital clock rotation if the ROTATE_SOFT_OFF_EN control in Register 0x581, Bit 2 is set to 1. When this bit is set, the synchronization logic rotation triggers the DAC ramp-down block, rotates the digital clocks, and ramps back up. These actions only occur if Bit 1 of the ROTATION_MODE control in Register 0x03B is set to 1 to enable a datapath clock rotation when the synchronization logic rotates.

## INTERRUPT REQUEST OPERATION

The AD9171 provides an interrupt request output signal ($\overline{IRQ}$) on Ball D9 ($\overline{IRQ0}$) and Ball E9 ($\overline{IRQ1}$) that can be used to notify an external host processor of significant device events. The $\overline{IRQ}$ output can be switched between the $\overline{IRQ0}$ pin or the $\overline{IRQ1}$ pin by setting the corresponding bit for the $\overline{IRQ}$ signal in Register 0x028, Register 0x029, Register 0x02A, and Register 0x02B. Upon assertion of the interrupt, query the device to determine the precise event that occurred. The $\overline{IRQx}$ pins are open-drain, active low outputs. Pull the $\overline{IRQx}$ pins high, external to the device. These pins can be tied to the interrupt pins of other devices with open-drain outputs to wire; OR these pins together.

Figure 57 shows a simplified block diagram of how the $\overline{IRQx}$ 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 $\overline{IRQx}$ 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, EVENT_STATUS reads back an event signal or an INTERRUPT_SOURCE signal. The AD9171 has several $\overline{IRQ}$ register blocks that can monitor up to 86 events, depending on the device configuration. Certain details vary by $\overline{IRQ}$ register block, as described in Table 42. Table 43 shows the source registers of the IRQ_EN, IRQ_RESET, and STATUS_MODE signals in Figure 57, as well as the address where EVENT_STATUS is read back.

*Table 42. $\overline{IRQ}$ Register Block Details*

| Register Block | Event Reported | EVENT_STATUS |
|---|---|---|
| 0x020 to 0x27 | Per chip | INTERRUPT_SOURCE if IRQ is enabled; if not, it is an event |
| 0x4B8 to 0x4BB; 0x470 to 0x473 | Per link and lane | INTERRUPT_SOURCE if 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.

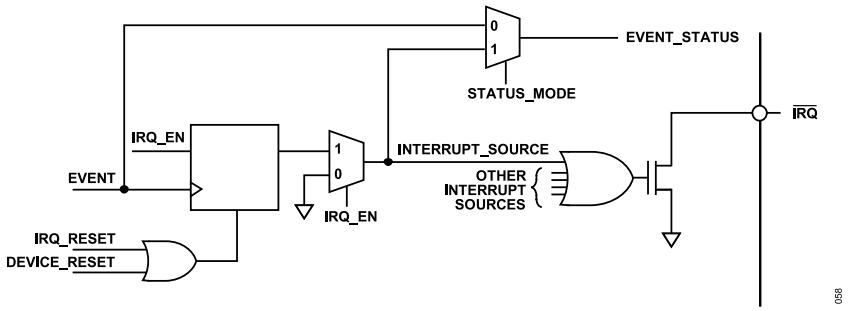


*Figure 57. Simplified Schematic of $\overline{IRQx}$ Circuitry*

*Table 43. $\overline{IRQ}$ Register Block Address of $\overline{IRQ}$ Signal Details*

| | Address of IRQ Signals[1] | | | |
|---|---|---|---|---|
| Register Block | IRQ_EN | IRQ_RESET | STATUS_MODE | EVENT_STATUS |
| 0x020 to 0x023 | 0x020 to 0x023; R/W per chip | 0x024 to 0x027; per chip | STATUS_MODE= IRQ_EN | 0x024 to 0x027; R per chip |
| 0x4B8 to 0x4BB | 0x4B8, 0x4B9; W per error type | 0x4BA, 0x4BB; W per error type | Not applicable, STATUS_MODE = 1 | 0x4BA, 0x4BB; W per chip |
| 0x470 to 0x473 | 0x470 to 0x473; W per error type | 0x470 to 0x473; W per link | Not applicable, STATUS_MODE = 1 | 0x470 to 0x473; W per link |

[1] R is read, W is write, and R/W is read/write.

## APPLICATIONS INFORMATION

### HARDWARE CONSIDERATIONS

#### Power Supply Recommendations

All the AD9171 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 V rms.

An LC filter on the output of the power supply is recommended to attenuate the noise, and must be placed as close to the AD9171 as possible. The AVDD1.0 supply, which supplies the clock receiver and DAC analog core circuitry, and the AVDD1.8 supply, which powers the DAC output and DAC PLL blocks, are the most noise sensitive supplies on the device. It is highly recommended that AVDD1.0 and AVDD1.8 be supplied separately with ultralow noise regulators, such as the ADP1763 and ADM7154 or better to achieve the best phase noise performance possible. Noisier regulators impose phase noise onto the DAC output.

The DVDD1.0 supply provides power to the digital datapath blocks and the SVDD1.0 supply powers the SERDES circuitry on the chip. The DVDD1.8 supply powers circuitry blocks related to the SPI, $\overline{\text{SYNCOUTx}}$± transmitter, SYSREF receiver, $\overline{\text{IRQx}}$, $\overline{\text{RESET}}$, and TXENx circuitry.

Take note of the maximum power consumption numbers shown 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 AD9171 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 or 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 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.

#### 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 29). The AD9171 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 AD9171 as close to 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 AD9171 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 30 and Figure 31) 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.

## APPLICATIONS INFORMATION

- 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 58).
- For each via pair, place a pair of ground vias adjacent to them to minimize the impedance discontinuity (see Figure 58).

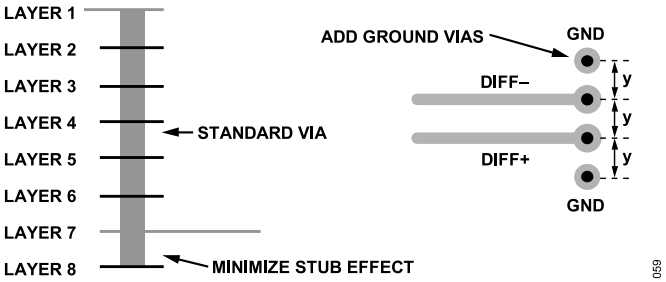


*Figure 58. Minimizing Stub Effect and Adding Ground Vias for Differential Stripline Traces*

### 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, make every effort to maintain a continuous impedance on the transmission line between the transmitting logic device and the AD9171. Minimizing the use of vias, or eliminating them entirely, 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 AD9171 handles this matching 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.

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.

### 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 10 mm (calculated by 12.5 mm × (12.5 Gbps/15 Gbps)) is adequate for operating the JESD204B link at speeds of up to 15 Gbps. This amount of channel length match is equivalent to about 85% UI on the AD9171 evaluation board. Managing the interconnect skew within a single link is straightforward.

Managing multiple links across multiple devices is more complex. However, follow the 10 mm guideline for length matching. The AD9171 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 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. Broadside vs. coplanar differential Tx lines are shown in Figure 59.

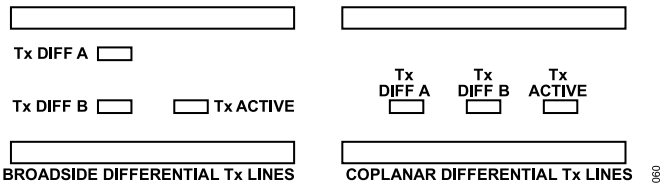


*Figure 59. Broadside vs. Coplanar Differential Stripline Routing Techniques*

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 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 60.

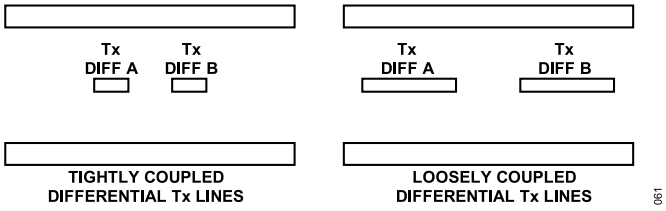


*Figure 60. Tightly Coupled vs. Loosely Coupled Differential Traces*

### AC Coupling Capacitors

The AD9171 requires that the JESD204B input signals be ac-coupled to the source. These capacitors must be 100 nF and placed as

## APPLICATIONS INFORMATION

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.

### SYNCOUT±, SYSREF±, and CLK± Signals

The SYSREF± signal on the AD9171 is a low speed, LVDS, differential signal. The SYNCOUTx± signals are LVDS or CMOS selectable. When LVDS mode is selected, use controlled impedance traces routed as 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 SYNCOUTx± signal from other noisy signals because noise on the SYNCOUTx± 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 61). If using a clock chip that can tightly control the phase of CLK± and SYSREF±, the trace length matching requirements are greatly reduced. The AD9516-0, ADCLK925, and HMC7044 are a few examples of possible clocking solutions depending on system requirements.

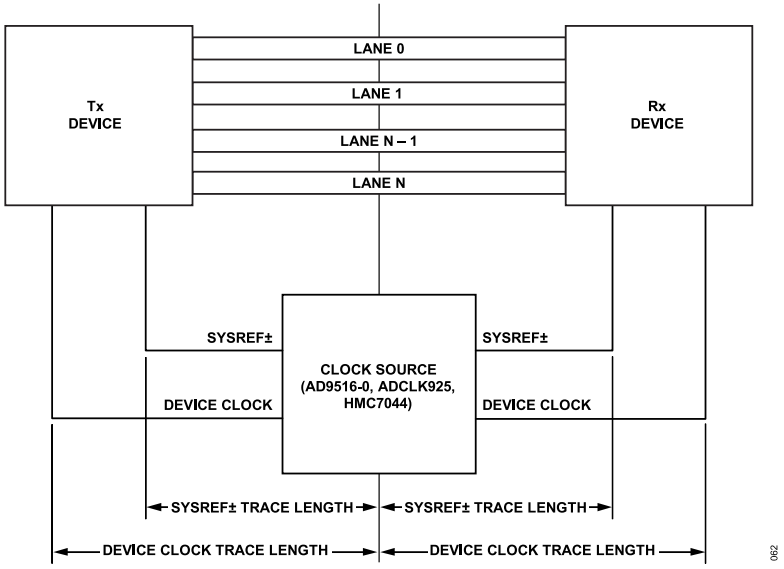


*Figure 61. SYSREF± Signal and Device Clock Trace Length*

## ANALOG INTERFACE CONSIDERATIONS

### DAC INPUT CLOCK CONFIGURATIONS

The AD9171 DAC sample clock or device clock (DACCLK) can be sourced directly through CLKIN± (Pin H12 and Pin J12) or by using the on-chip, integrated, integer PLL VCO with the same CLKIN± differential input serving as the reference. Clock multiplying employs the on-chip DAC PLL that accepts a reference clock operating at a submultiple of the desired DACCLK rate. The PLL then multiplies the reference clock up to the desired DACCLK frequency, which generates all the clocks within the AD9171, as shown in Figure 64.

The AD9171 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 100 Ω, it is recommended that the clock source be ac-coupled to the CLKIN± input pins. Improved phase noise performance can be achieved with a higher clock input level. The quality of the clock source, as well as its interface to the AD9171 clock input, directly impacts ac performance. Select the phase noise and spur characteristics of the clock source to meet the target application requirements. The typical phase noise performance when directly clocking the AD9171 with the input clock duty cycle correction on (enabled by default) is shown in Figure 63 compared to the on-chip PLL phase noise.

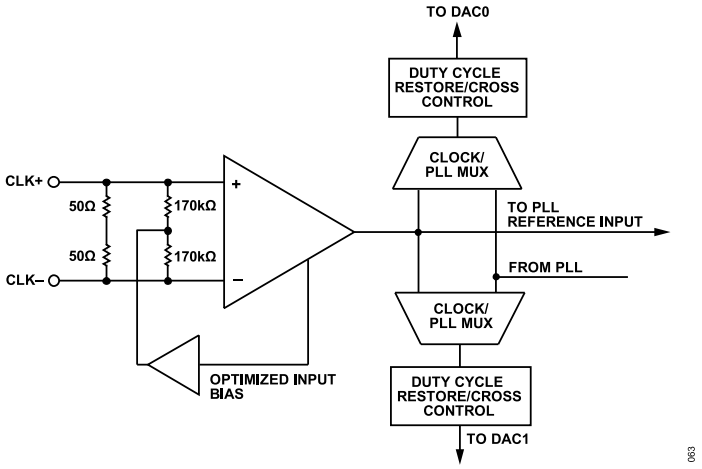


*Figure 62. Clock Receiver Input Simplified Equivalent Circuit*

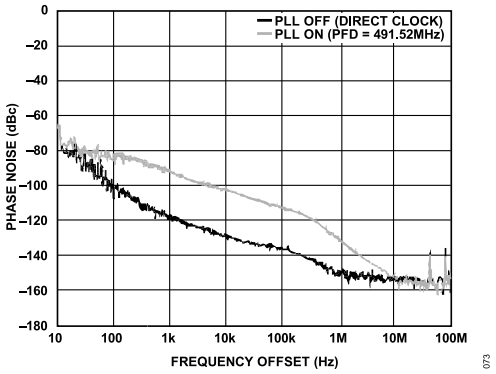


*Figure 63. Phase Noise vs. Frequency Offset; Direct Clock and PLL Phase Noise, 12 GHz DAC Sample Rate, 1.65 GHz Output Frequency*

## ANALOG INTERFACE CONSIDERATIONS

### DAC On-Chip PLL

When using the on-chip PLL, the predivider setting is selected (via Register 0x793, Bits[1:0]) to divide the clock receiver frequency internally to be within the range of 25 MHz to 770 MHz for the phase frequency detector (PFD) circuitry block input.

The AD9171 DAC VCO operates over a frequency range of 8.74 GHz to 12.4 GHz. Enable the DAC PLL synthesizer by setting Register 0x095, Bit 0 to 0. The DAC clock rate is user configurable to be the VCO frequency divided by 2 (4.37 GHz to 6.2 GHz) or the VCO frequency divided by 3 (2.92 GHz to 4.1 GHz) by setting Register 0x094, Bits[1:0]. See the Start-Up Sequence section for instructions on how to program the PLL.

The AD9171 DAC PLL requires an external loop filter. The recommended filter is a standard passive filter, as shown in Figure 64. The user can customize the filter according to the PFD frequency, reference clock phase noise, and DAC output phase noise requirements. For lowest jitter applications, use the highest possible PFD frequency to minimize the contribution of in-band noise from the PLL. However, when operating with a VCO frequency from 9.96 GHz to 10.87 GHz, set the PFD frequency to less than 225 MHz to ensure optimized stability over temperature.

Set the PLL filter bandwidth such that the in-band noise of the PLL intersects with the open-loop noise of the VCO, minimizing the contribution of both to the overall noise.

The DAC PLL uses an integer type synthesizer to achieve the DAC sampling clock. The relation between DAC clock and the reference clock is as follows:

$$f_{DAC} = (8 \times N \times f_{REF})/M/(Register\ 0x094,\ Bits[1:0] + 1)$$

where:
$f_{DAC}$ is the desired DAC clock rate.
$N$ is the VCO feedback divider ratio, ranging from 2 to 50.
$f_{REF}$ is the reference clock.
$M$ is the reference clock divider ratio; the valid values for reference clock divider (predivider) are 1, 2, 3, or 4 by setting Register 0x793, Bits[1:0].

The VCO automatic calibration is triggered by the falling edge of Register 0x792, Bit 1 transitioning from a logic high to logic low. A lock detector bit (Register 0x7B5, Bit 0) is provided to indicate that the DAC PLL achieved lock. If Register 0x7B5, Bit 0 = 1, the PLL has locked.

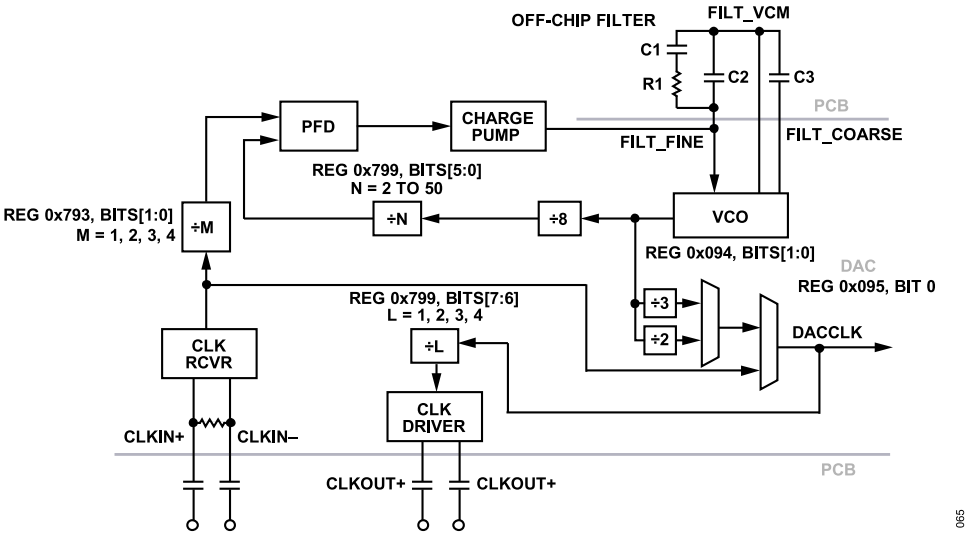


*Figure 64. DAC PLL and Clock Path Block Diagram*

## ANALOG INTERFACE CONSIDERATIONS

### CLOCK OUTPUT DRIVER

The AD9171 is capable of generating and outputting a high performance divide down clock that can be used to clock a companion analog-to-digital converter (ADC). The integer clock divider supports divide ratios 1, 2, 3, and 4 and can be programmed by Register 0x799, Bits[7:6] to set the desired output frequency. The output frequency ranges from 727.5 MHz to 3 GHz.

### ANALOG OUTPUTS

The AD9171 provides complementary current outputs, DAC0± and DAC1±. Figure 65 shows an equivalent output circuit for the DAC. The AD9171 features an internal 100 Ω termination resistor ($R_{INT}$) that eliminates the need to terminate the DAC current output externally on the PCB board. Two RF chokes, one for each DAC output leg, are required to provide a dc current path for the DAC output. The value of the choke depends on the desired output frequency range. In general, a larger choke provides a lower cutoff output frequency. A 2:1 balun is recommended to transform the differential DAC output to a single-ended signal.

$I_{OUTFS}$ is the full-scale current of each of the two DAC outputs. It is a 10-bit value stored in Register 0x05A and Register 0x059. The full-scale current of the DAC is typically 19.91 mA. The DAC full-scale current can be adjusted typically from 16 mA to 26 mA by programming the appropriate value in Register 0x05A.

$$I_{OUTFS} = 16\ mA + FSC_CTRL \times (25/1024)\ (mA)$$

The example shown in Figure 65 can be modeled as a pair of dc current sources that source half of the $I_{OUTFS}$ current to each output. This differential ac current source models the signal (that is, a digital code) dependent nature of the DAC output.

The ac differential current measured at DACx+ and DACx− is as follows:

$$I_{AC_DIFF}\ (mA) = DACCODE/DACCODEMAX \times I_{OUTFS} \times R_{INT}/(R_{LOAD} + R_{INT})$$

where:
*DACCODE* = 0 to 65,535 (decimal).
$R_{LOAD}$ is the load impedance seen on DACx+ and DACx−.

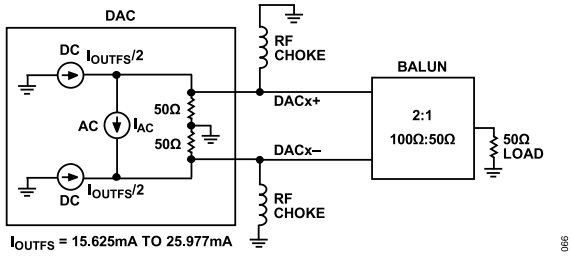


*Figure 65. Equivalent DAC Output Circuit and Recommended DAC Output Network*

## START-UP SEQUENCE

Several steps are required to program the AD9171 to the proper operating state after the device is powered up. This sequence is divided into several steps, and is listed in Table 44 to Table 53, along with an explanation of the purpose of each step. Private registers are reserved but must be written for proper operation. Blank cells or cells with a variable or bit field name (in all capital letters) in Table 44 to Table 53 indicate that the value depends on the result as described in the Description column.

*Table 44. Power-Up and Required Register Writes*

| R/W | Register | Bits | Value | Description |
|---|---|---|---|---|
| W | 0x000 | [7:0] | 0x81 | Soft reset. |
| W | 0x000 | [7:0] | 0x3C | Release reset and set to 4-wire SPI (optional; leave at the default of the 3-wire SPI). |
| W | 0x091 | [7:0] | 0x00 | Power up clock receiver. |
| W | 0x206 | [7:0] | 0x01 | Take PHYs out of reset. |
| W | 0x705 | [7:0] | 0x01 | Enable boot loader. |
| | Pause | | | Wait 1 ms. |
| W | 0x090 | [7:0] | 0x00 | Power on DACs and bias circuitry. |

*Table 45. DAC PLL Configuration*

| R/W | Register | Bits | Value | Description |
|---|---|---|---|---|
| W | 0x095 | [7:0] | 0x00 or 0x01 | Bypass PLL. Set to 0x00 to use internal DAC PLL. If the user plans to supply the DAC clock directly, set this register to 0x01 and execute the following two register writes; then, skip the remaining writes in this table. |
| W | 0x790 | [7:0] | 0xFF or 0x00 | Write this register to 0xFF if bypassing the PLL (Register 0x095 = 0x01). If using the PLL, write this register to 0x00. |
| W | 0x791 | [7:0] | 0x1F or 0x00 | Write this register to 0xFF if bypassing the PLL (Register 0x095 = 0x01) and then skip the remaining register writes in this table and continue to Table 46. If using the PLL, write this register to 0x00 as well as the remainder of the register writes in this table. |
| W | 0x796 | [7:0] | 0xE5 | DAC PLL required write. |
| W | 0x7A0 | [7:0] | 0xBC | DAC PLL required write. |
| W | 0x794 | [5:0] | DACPLL_CP | Set DAC PLL charge pump current. The recommended setting is 0x08, but can range from 0x04 to 0x10 for different phase noise performance targets. |
| W | 0x797 | [7:0] | 0x10 | DAC PLL required write. |
| W | 0x797 | [7:0] | 0x20 | DAC PLL required write. |
| W | 0x798 | [7:0] | 0x10 | DAC PLL required write. |
| W | 0x7A2 | [7:0] | 0x7F | DAC PLL required write. |
| | Pause | | | Wait 100 ms. |
| W | 0x799 | | | DAC PLL divider settings. |
| | | [7:6] | ADC_CLK_DIVIDER | ADC driver/clock output divide ratio.<br>0b00 = ÷1.<br>0b01 = ÷2.<br>0b10 = ÷3.<br>0b11 = ÷4. |
| | | [5:0] | N_DIVIDER | Programmable N divider. N_DIVIDER = ($f_{VCO}$ × M_DIVIDER)/(8 × reference clock). |
| W | 0x793 | | | DAC PLL divider settings. |
| | | [7:2] | 0x06 | Keep default value for these bits. |
| | | [1:0] | M_DIVIDER-1 | Programmable predivider M_DIVIDER-1 (in n − 1 notation). The relevant calculation is as follows:<br>PFD Frequency = reference clock/M_DIVIDER, where 25 MHz ≤ PFD frequency ≤ 770 MHz.<br>If 9.96 GHz ≤ PLL VCO frequency ≤ 10.87 GHz, must keep 25 MHz ≤ PFD frequency ≤ 225 MHz.<br>0b00 = ÷1.<br>0b01 = ÷2.<br>0b10 = ÷3.<br>0b11 = ÷4. |
| W | 0x094 | [7:2] | 0x00 | Keep default value for these bits. |
| | | 1 | PLL_VCO_DIV3_EN | Enable PLL output clock to be divided by 3. If this bit is set to 1, DAC clock = PLL VCO frequency/3. |

## START-UP SEQUENCE

*Table 45. DAC PLL Configuration*

| R/W | Register | Bits | Value | Description |
|---|---|---|---|---|
| | | 0 | PLL_VCO_DIV2_EN | Enable PLL output clock to be divided by 2. Either this bit or Bit 1 in this register can be set to 1, but both bits cannot be set at the same time (there is no divide by 6 option).<br>0b1: DAC clock = PLL VCO frequency/2. |
| W | 0x792 | [7:0] | 0x02 | Reset VCO. |
| W | 0x792 | [7:0] | 0x00 | |
| | Pause | | | Wait 100 ms for PLL to lock. |
| R | 0x7B5 | 0 | 0b1 | Ensure PLL is locked by reading back a value of 1 for bit 0 of this register. |

*Table 46. Delay Lock Loop (DLL) Configuration*

| R/W | Register | Bits | Value | Description |
|---|---|---|---|---|
| W | 0x0C0 | [7:0] | 0x00 | Power-up delay line. |
| W | 0x0DB | [7:0] | 0x00 | |
| W | 0x0DB | [7:0] | 0x01 | Update DLL settings to circuitry. |
| W | 0x0DB | [7:0] | 0x00 | |
| W | 0x0C1 | [7:0] | 0x68 or 0x48 | Set DLL search mode. If $f_{DAC}$ is < 4.5 GHz, set this register to 0x48. Otherwise, set this register to 0x68. |
| W | 0x0C1 | [7:0] | 0x69 or 0x49 | Set DLL search mode. If $f_{DAC}$ is < 4.5 GHz, set this register to 0x49. Otherwise, set this register to 0x69. |
| W | 0x0C7 | [7:0] | 0x01 | Enable DLL read status. |
| R | 0x0C3 | 0 | 0b1 | Ensure DLL is locked by reading back a value of 1 for Bit 0 of this register. |

*Table 47. Calibration*

| R/W | Register | Bits | Value | Description |
|---|---|---|---|---|
| W | 0x050 | [7:0] | 0x2A | Optimized calibration setting register write. |
| W | 0x061 | [7:0] | 0x68 | Required calibration control register write. |
| W | 0x051 | [7:0] | 0x82 | Optimized calibration setting register write. |
| | Pause | | | Wait 1 ms. |
| W | 0x051 | [7:0] | 0x83 | Required calibration control register write. |
| W | 0x081 | [7:0] | 0x03 | Required calibration control register write. |

*Table 48. JESD204B Mode Setup*

| R/W | Register | Bits | Value | Description |
|---|---|---|---|---|
| W | 0x100 | [7:0] | 0x00 | Power up digital datapath clocks when internal clocks are stable. |
| W | 0x110 | [5:0] | JESD_MODE | Bit 5 of the JESD_MODE bit field determines whether the device is operating in single link or dual link modes. 0 = single-link mode; 1 = dual-link mode. Bits[4:0] determine the SERDES JESD204B mode of operation chosen from the appropriate single-link or dual-link modes in Table 14. |
| W | 0x111 | [7:4] | DP_INTERP_MODE | Main datapath interpolation mode. The valid interpolation options for this control is based on the JESD_MODE selected in Register 0x110. Bit 7 of Register 0x110 equals 1 if the JESD_MODE, DP_INTERP_MODE, and CH_INTERP_MODE settings are not a valid combination. |
| | | [3:0] | CH_INTERP_MODE | Channel datapath interpolation mode. The valid interpolation options for this control is based on the JESD_MODE selected in Register 0x110. Bit 7 of Register 0x110 equals 1 if the JESD_MODE, DP_INTERP_MODE, and CH_INTERP_MODE settings are not a valid combination. |
| W | 0x084 | 6 | SYSREF_INPUTMODE | SYSREF± signal input mode selection.<br>0b0 = ac-coupled.<br>0b1 = dc-coupled. |
| | | 0 | SYSREF_PD | If using Subclass 0, this bit can be set to 1 to power down the SYSREF± receiver. If using Subclass 1, keep at the default of 0. |
| W | 0x312 | [7:4] | | Set $\overline{\text{SYNCOUTx}}$± error duration, depending on the selected mode. |
| W | 0x300 | 3 | LINK_MODE | Corresponds to the mode selection made in Register 0x110.<br>0b0 = single-link mode.<br>0b1 = dual-link mode. |

## START-UP SEQUENCE

*Table 48. JESD204B Mode Setup*

| R/W | Register | Bits | Value | Description |
|---|---|---|---|---|
| | | 2 | 0b0 | Select Link 0 for setup. This bit selects the link QBD being paged.<br>0b0 = Link 0 (QBD0).<br>0b1 = Link 1 (QBD1). |
| | | [1:0] | 0b00 | Keep links disabled until end of routine. |
| W | 0x475 | [7:0] | 0x09 | Soft reset the JESD204B quad-byte deframer. |
| W | 0x453 | 7 | SCR | Set scrambling option for SERDES data.<br>0 = disable scrambling.<br>1 = enable scrambling. |
| | | [4:0] | L−1 | Write the L value (in n − 1 notation) for the selected JESD_MODE. |
| W | 0x458 | [7:5] | SUBCLASSV | For Subclass 0, set this bit to 0. For Subclass 1, set this bit to 1. |
| | | [4:0] | NP−1 | Write the NP value (in n − 1 notation) for the selected JESD_MODE. |
| W | 0x475 | [7:0] | 0x01 | Bring the JESD204B quad-byte deframer out of reset. |
| W | 0x300 | | | If running in dual link mode, repeat writes for Link 1 as follows. If running in single-link mode, skip the remaining steps in this table. |
| | | 3 | LINK_MODE | Corresponds to the mode selection made in Register 0x110.<br>0b0 = single-link mode.<br>0b1 = dual link mode. |
| | | 2 | 0b1 | Select Link 1 for setup. This bit selects which link QBD is being paged.<br>0b0 = Link 0 (QBD0).<br>0b1 = Link 1 (QBD1). |
| | | [1:0] | 0b00 | Keep links disabled until end of routine. |
| W | 0x475 | [7:0] | 0x09 | Soft reset the JESD204B quad-byte deframer. |
| W | 0x453 | 7 | SCR | Set scrambling option for SERDES data.<br>0 = disable scrambling.<br>1 = enable scrambling. |
| W | 0x458 | [4:0] | L−1 | Write the L value (in n − 1 notation) for the selected JESD_MODE. |
| | | [7:5] | SUBCLASSV | For Subclass 0, set this bit to 0. For Subclass 1, set this bit to 1. |
| | | [4:0] | NP−1 | Write the NP value (in n − 1 notation) for the selected JESD_MODE. |
| W | 0x475 | [7:0] | 0x01 | Bring the JESD204B quad-byte deframer out of reset. |

Table 49 lists optional registers to configure the channel datapaths if they are being configured for a specific application.

*Table 49. Channel Datapath Setup: Digital Gain and Channel NCOs*

| R/W | Register | Bits | Value | Description |
|---|---|---|---|---|
| W | 0x008 | [5:0] | | CHANNEL_PAGE. Select the channels to be programmed at the same time (or repeat this block for each channel to independently program values). Bit 0 of this control corresponds to the Channel 0 datapath. Bit 3 of this control corresponds to the Channel 1 datapath. |
| W | 0x146 | [7:0] | | CHNL_GAIN[7:0]. Write LSBs of channel digital gain. Configure digital gain for selected channels in Paging Register 0x008. Calculation: CHNL_GAIN = $2^{11}$ × 10(dBGain/20) where dBGain is the gain value in dB for the channel gain desired. |
| W | 0x147 | [7:0] | | CHNL_GAIN[11:8]. Write MSBs of channel digital gain. Calculations shown in Register 0x146. |
| W | 0x130 | 6 | | Enable NCO for selected channels in paging Register 0x008.<br>0b0 = disable NCO.<br>0b1 = enable NCO. |
| | | 2 | | Enable NCO modulus for selected channels in paging Register 0x008.<br>0b0 = disable NCO modulus.<br>0b1 = enable NCO modulus. |
| | | 1 | | Select sideband from modulation result.<br>0b0 = upper sideband. |

## START-UP SEQUENCE

*Table 49. Channel Datapath Setup: Digital Gain and Channel NCOs*

| R/W | Register | Bits | Value | Description |
|---|---|---|---|---|
| | | | | 0b1 = lower sideband (spectral flip). |
| | | 0 | | If dc test mode or NCO test mode is desired, set this bit to 1 to enable the test tone generation. Otherwise, set this bit to the default value of 0.<br>Integer NCO mode calculation: DDSC_FTW = ($f_{CARRIER}/f_{NCO}$) × $2^{48}$, where $f_{NCO}$ = $f_{DATA}$/CH_INTERP_MODE. |
| W | 0x132 | [7:0] | | Write DDSC_FTW[7:0]. |
| W | 0x133 | [7:0] | | Write DDSC_FTW[15:8]. |
| W | 0x134 | [7:0] | | Write DDSC_FTW[23:16]. |
| W | 0x135 | [7:0] | | Write DDSC_FTW[31:24]. |
| W | 0x136 | [7:0] | | Write DDSC_FTW[39:32]. |
| W | 0x137 | [7:0] | | Write DDSC_FTW[47:40]. |
| W | 0x138 | [7:0] | | Write DDSC_NCO_PHASE_OFFSET[7:0]. Calculation: DDSC_NCO_PHASE_OFFSET = (Degrees Offset/180) × $2^{15}$. |
| W | 0x139 | [7:0] | | Write DDSC_NCO_PHASE_OFFSET[15:8].<br>If using NCO modulus mode, also program modulus parameters. If not, skip this section. For modulus NCO mode: ($f_{CARRIER}/f_{NCO}$) = (X + (A/B))/$2^{48}$ where DDSC_ACC_DELTA = A, DDSC_ACC_MODULUS = B, and DDSC_FTW = X. |
| W | 0x13A | [7:0] | | Write DDSC_ACC_MODULUS[7:0]. |
| W | 0x13B | [7:0] | | Write DDSC_ACC_MODULUS[15:8]. |
| W | 0x13C | [7:0] | | Write DDSC_ACC_MODULUS[23:16]. |
| W | 0x13D | [7:0] | | Write DDSC_ACC_MODULUS[31:24]. |
| W | 0x13E | [7:0] | | Write DDSC_ACC_MODULUS[39:32]. |
| W | 0x13F | [7:0] | | Write DDSC_ACC_MODULUS[47:40]. |
| W | 0x140 | [7:0] | | Write DDSC_ACC_DELTA[7:0]. |
| W | 0x141 | [7:0] | | Write DDSC_ACC_DELTA[15:8]. |
| W | 0x142 | [7:0] | | Write DDSC_ACC_DELTA[23:16]. |
| W | 0x143 | [7:0] | | Write DDSC_ACC_DELTA[31:24]. |
| W | 0x144 | [7:0] | | Write DDSC_ACC_DELTA[39:32]. |
| W | 0x145 | [7:0] | | Write DDSC_ACC_DELTA[47:40]. |
| W | 0x131 | 0 | 0b1 | Update all NCO phase and FTW words. |

Table 50 lists optional registers to configure the main DAC datapaths if they are being configured for a specific application.

*Table 50. Main DAC Datapath Setup: PA Protect and Main NCOs*

| R/W | Register | Bits | Value | Description |
|---|---|---|---|---|
| W | 0x008 | [7:6] | | MAINDAC_PAGE. Select the main DAC datapath to be programmed at the same time (or repeat this block for each DAC datapath to independently program values). Bit x of this control corresponds to the DAC x datapath. |
| W | 0x112 | 3 | | Enable NCO for selected channels in paging Register 0x008.<br>0b0 = disable NCO.<br>0b1 = enable NCO. |
| | | 2 | | Enable NCO modulus for selected channels in paging Register 0x008.<br>0b0 = disable NCO modulus.<br>0b1 = enable NCO modulus. |
| | | 1 | | Select sideband from modulation result.<br>0b0 = upper sideband.<br>0b1 = lower sideband (spectral flip). |
| | | 0 | | Set this bit to 0.<br>Integer NCO mode calculation: DDSM_FTW = ($f_{CARRIER}/f_{DAC}$) × $2^{48}$. |
| W | 0x114 | [7:0] | | Write DDSM_FTW[7:0]. |
| W | 0x115 | [7:0] | | Write DDSM_FTW[15:8]. |
| W | 0x116 | [7:0] | | Write DDSM_FTW[23:16]. |

## START-UP SEQUENCE

*Table 50. Main DAC Datapath Setup: PA Protect and Main NCOs*

| R/W | Register | Bits | Value | Description |
|---|---|---|---|---|
| W | 0x117 | [7:0] | | Write DDSM_FTW[31:24]. |
| W | 0x118 | [7:0] | | Write DDSM_FTW[39:32]. |
| W | 0x119 | [7:0] | | Write DDSM_FTW[47:40]. |
| W | 0x11C | [7:0] | | Write DDSM_NCO_PHASE_OFFSET[7:0]. Calculation: DDSM_NCO_PHASE_OFFSET = (degrees offset/180) × $2^{15}$. |
| W | 0x11D | [7:0] | | Write DDSM_NCO_PHASE_OFFSET[15:8].<br>If using NCO modulus mode, also program modulus parameters. If not, skip this section. For modulus NCO mode: ($f_{CARRIER}/f_{DAC}$) = (X + (A/B))/$2^{48}$, where DDSM_ACC_DELTA = A, DDSM_ACC_MODULUS = B, and DDSM_FTW = X. |
| W | 0x124 | [7:0] | | Write DDSM_ACC_MODULUS[7:0]. |
| W | 0x125 | [7:0] | | Write DDSM_ACC_MODULUS[15:8]. |
| W | 0x126 | [7:0] | | Write DDSM_ACC_MODULUS[23:16]. |
| W | 0x127 | [7:0] | | Write DDSM_ACC_MODULUS[31:24]. |
| W | 0x128 | [7:0] | | Write DDSM_ACC_MODULUS[39:32]. |
| W | 0x129 | [7:0] | | Write DDSM_ACC_MODULUS[47:40]. |
| W | 0x12A | [7:0] | | Write DDSM_ACC_DELTA[7:0]. |
| W | 0x12B | [7:0] | | Write DDSM_ACC_DELTA[15:8]. |
| W | 0x12C | [7:0] | | Write DDSM_ACC_DELTA[23:16]. |
| W | 0x12D | [7:0] | | Write DDSM_ACC_DELTA[31:24]. |
| W | 0x12E | [7:0] | | Write DDSM_ACC_DELTA[39:32]. |
| W | 0x12F | [7:0] | | Write DDSM_ACC_DELTA[47:40]. |
| W | 0x113 | 0 | 0b1 | Update all NCO phase and FTW words. |

*Table 51. JESD204B SERDES Required Interface Setup*

| R/W | Register | Bits | Value | Description |
|---|---|---|---|---|
| W | 0x240 | [7:0] | 0xAA or 0xFF | EQ settings determined by amount of insertion loss according to Table 18. For insertion loss ≤ 11 dB, set to 0xAA; otherwise, set to 0xFF. |
| W | 0x241 | [7:0] | 0xAA or 0xFF | EQ settings determined by amount of insertion loss according to Table 18. For insertion loss ≤ 11 dB, set to 0xAA; otherwise, set to 0xFF. |
| W | 0x242 | [7:0] | 0x55 or 0xFF | EQ settings determined by amount of insertion loss according to Table 18. For insertion loss ≤ 11 dB, set to 0x55; otherwise, set to 0xFF. |
| W | 0x243 | [7:0] | 0x55 or 0xFF | EQ settings determined by amount of insertion loss according to Table 18. For insertion loss ≤ 11 dB, set to 0x55; otherwise, set to 0xFF. |
| W | 0x244 | [7:0] | 0x1F | EQ settings. |
| W | 0x245 | [7:0] | 0x1F | EQ settings. |
| W | 0x246 | [7:0] | 0x1F | EQ settings. |
| W | 0x247 | [7:0] | 0x1F | EQ settings. |
| W | 0x248 | [7:0] | 0x1F | EQ settings. |
| W | 0x249 | [7:0] | 0x1F | EQ settings. |
| W | 0x24A | [7:0] | 0x1F | EQ settings. |
| W | 0x24B | [7:0] | 0x1F | EQ settings. |
| W | 0x201 | [7:0] | | Power down unused PHYs. Bit x corresponds to SERDINx± pin power-down. |
| W | 0x203 | | | If in single-link mode, set to 0x01. If in dual-link mode and using both $\overline{\text{SYNCOUTx±}}$ signals, set to 0x00. |
| | | 1 | 0b0 | Power up $\overline{\text{SYNCOUT0±}}$ driver by setting this bit to 0. |
| | | 0 | | Power up $\overline{\text{SYNCOUT1±}}$ driver by setting this bit to 0 if using dual link and both $\overline{\text{SYNCOUTx±}}$ signals. |
| W | 0x253 | [7:0] | 0x01 | Set $\overline{\text{SYNCOUT0±}}$ to be LVDS output. For CMOS output on $\overline{\text{SYNCOUT0±}}$, set Bit 0 to 0. |
| W | 0x254 | [7:0] | 0x01 | Set $\overline{\text{SYNCOUT1±}}$ to be LVDS output. For CMOS output on $\overline{\text{SYNCOUT1±}}$, set Bit 0 to 0. |
| W | 0x210 | [7:0] | 0x16 | SERDES required register write. |
| W | 0x216 | [7:0] | 0x05 | SERDES required register write. |
| W | 0x212 | [7:0] | 0xFF | SERDES required register write. |

## START-UP SEQUENCE

*Table 51. JESD204B SERDES Required Interface Setup*

| R/W | Register | Bits | Value | Description |
|---|---|---|---|---|
| W | 0x212 | [7:0] | 0x00 | SERDES required register write. |
| W | 0x210 | [7:0] | 0x87 | SERDES required register write. |
| W | 0x216 | [7:0] | 0x11 | SERDES required register write. |
| W | 0x213 | [7:0] | 0x01 | SERDES required register write. |
| W | 0x213 | [7:0] | 0x00 | SERDES required register write. |
| W | 0x200 | [7:0] | 0x00 | Power up the SERDES circuitry blocks. |
| | Pause | | | Wait 100 ms. |
| W | 0x210 | [7:0] | 0x86 | SERDES required register write. |
| W | 0x216 | [7:0] | 0x40 | SERDES required register write. |
| W | 0x213 | [7:0] | 0x01 | SERDES required register write. |
| W | 0x213 | [7:0] | 0x00 | SERDES required register write. |
| W | 0x210 | [7:0] | 0x86 | SERDES required register write. |
| W | 0x216 | [7:0] | 0x00 | SERDES required register write. |
| W | 0x213 | [7:0] | 0x01 | SERDES required register write. |
| W | 0x213 | [7:0] | 0x00 | SERDES required register write. |
| W | 0x210 | [7:0] | 0x87 | SERDES required register write. |
| W | 0x216 | [7:0] | 0x01 | SERDES required register write. |
| W | 0x213 | [7:0] | 0x01 | SERDES required register write. |
| W | 0x213 | [7:0] | 0x00 | SERDES required register write. |
| W | 0x280 | [7:0] | 0x05 | SERDES required register write. |
| W | 0x280 | [7:0] | 0x01 | Start up SERDES PLL circuitry blocks and initiate SERDES PLL calibration. |
| R | 0x281 | 0 | 0b1 | Ensure Bit 0 of this register reads back 1 to indicate the SERDSES PLL is locked. |

Crossbar mapping writes the SERDINx input pin that is the source for each given logical lane in these registers. A value of x corresponds to mapping data from the SERDINx± pin to the logical lane of the control bit field. These values in Table 52 vary with different PCB layout routing.

*Table 52. Transport Layer Setup, Synchronization, and Enable Links*

| R/W | Register | Bits | Value | Description |
|---|---|---|---|---|
| W | 0x308 | [7:0] | | Crossbar setup. Program the physical lane value that is providing data (the source) for each of the logical lanes. [5:3] = Logical Lane 1 source, [2:0] = Logical Lane 0 source. |
| W | 0x309 | [7:0] | | [5:3] = Logical Lane 3 source, [2:0] = Logical Lane 2 source. |
| W | 0x30A | [7:0] | | [5:3] = Logical Lane 5 source, [2:0] = Logical Lane 4 source. |
| W | 0x30B | [7:0] | | [5:3] = Logical Lane 7 source, [2:0] = Logical Lane 6 source. |
| W | 0x306 | [7:0] | 0x0C | If operating in Subclass 0, this register write is not needed. |
| W | 0x307 | [7:0] | 0x0C | If operating in Subclass 0, this register write is not needed. |
| W | 0x304 | [7:0] | | If operating in Subclass 0, this register write is not needed. For Subclass 1, these values must be determined by following one of the deterministic latency methods (with or without known delays), as mentioned in the Link Delay section. |
| W | 0x305 | [7:0] | | If operating in Subclass 0, this register write is not needed. For Subclass 1, these values must be determined by following one of the deterministic latency methods (with or without known delays), as mentioned in the Link Delay section. |
| W | 0x03B | [7:0] | 0xF1 | Enable the sync logic, and set the rotation mode to reset the synchronization logic upon a sync reset trigger. |
| W | 0x03A | [7:0] | 0x02 | Set up sync for one-shot sync mode. |
| | SYSREF± | | | If operating in Subclass 1, send SYSREF± pulse edges to the device for synchronization alignment. |
| W | 0x300 | 3 | LINK_MODE | Corresponds to the mode selection made in Register 0x110.<br>0b0 = single-link mode.<br>0b1 = dual-link mode. |
| | | 2 | 0b0 | Select Link 0 for setup. This bit selects which link QBD is being paged. |

## START-UP SEQUENCE

*Table 52. Transport Layer Setup, Synchronization, and Enable Links*

| R/W | Register | Bits | Value | Description |
|---|---|---|---|---|
| | | | | 0b0 = Link 0 (QBD0).<br>0b1 = Link 1 (QBD1). |
| | | [1:0] | LINK_EN | Enables the links.<br>0b01 = single-link mode<br>0b11 = dual-link mode. |

*Table 53. Cleanup Registers*

| R/W | Register | Bits | Value | Description |
|---|---|---|---|---|
| W | 0x1DE | [7:0] | 0x00 | Disable analog SPI. To debug and continue readback capability, write 0x03. |
| W | 0x008 | [7:0] | 0xC0 | Page all main DACs for TXEN control update. |
| W | 0x596 | [7:0] | 0x0C | SPI turn on TXENx feature. |

## REGISTER SUMMARY

*Table 54. Register Summary*

| Reg | Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Reset | RW |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x000 | SPI_INTFCONFA | SOFTRESET_M | LSBFIRST_M | ADDRINC_M | SDOACTIVE_M | SDOACTIVE | ADDRINC | LSBFIRST | SOFTRESET | 0x00 | R/W |
| 0x001 | SPI_INTFCONFB | SINGLEINS | $\overline{CS}$STALL | RESERVED | | | | | | 0x00 | R/W |
| 0x003 | SPI_CHIPTYPE | CHIP_TYPE | | | | | | | | 0x04 | R |
| 0x004 | SPI_PRODIDL | PROD_ID[7:0] | | | | | | | | 0x71 | R |
| 0x005 | SPI_PRODIDH | PROD_ID[15:8] | | | | | | | | 0x91 | R |
| 0x006 | SPI_CHIPGRADE | PROD_GRADE | | | | DEV_REVISION | | | | 0x02 | R |
| 0x008 | SPI_PAGEINDX | MAINDAC_PAGE | | CHANNEL_PAGE | | | | | | 0xFF | R/W |
| 0x00A | SPI_SCRATCHPAD | SCRATCHPAD | | | | | | | | 0x00 | R/W |
| 0x010 | CHIP_ID_L | CHIP_ID[7:0] | | | | | | | | 0x00 | R |
| 0x011 | CHIP_ID_M1 | CHIP_ID[15:8] | | | | | | | | 0x00 | R |
| 0x012 | CHIP_ID_M2 | CHIP_ID[23:16] | | | | | | | | 0x00 | R |
| 0x013 | CHIP_ID_H | CHIP_ID[31:24] | | | | | | | | 0x00 | R |
| 0x020 | IRQ_ENABLE | RESERVED | | | EN_SYSREF_JITTER | EN_DATA_READY | EN_LANE_FIFO | EN_PRBSQ | EN_PRBSI | 0x00 | R/W |
| 0x021 | IRQ_ENABLE0 | RESERVED | | | | EN_DAC0_CAL_DONE | RESERVED | | EN_PAERR0 | 0x00 | R/W |
| 0x022 | IRQ_ENABLE1 | RESERVED | | | | EN_DAC1_CAL_DONE | RESERVED | | EN_PAERR1 | 0x00 | R/W |
| 0x023 | IRQ_ENABLE2 | RESERVED | | EN_DLL_LOST | EN_DLL_LOCK | RESERVED | | EN_PLL_LOST | EN_PLL_LOCK | 0x00 | R/W |
| 0x024 | IRQ_STATUS | RESERVED | | | IRQ_SYSREF_JITTER | IRQ_DATA_READY | IRQ_LANE_FIFO | IRQ_PRBSQ | IRQ_PRBSI | 0x00 | R/W |
| 0x025 | IRQ_STATUS0 | RESERVED | | | | IRQ_DAC0_CAL_DONE | RESERVED | | IRQ_PAERR0 | 0x00 | R/W |
| 0x026 | IRQ_STATUS1 | RESERVED | | | | IRQ_DAC1_CAL_DONE | RESERVED | | IRQ_PAERR1 | 0x00 | R/W |
| 0x027 | IRQ_STATUS2 | RESERVED | | IRQ_DLL_LOST | IRQ_DLL_LOCK | RESERVED | | IRQ_PLL_LOST | IRQ_PLL_LOCK | 0x00 | R/W |
| 0x028 | IRQ_OUTPUT_MUX | RESERVED | | | MUX_SYSREF_JITTER | MUX_DATA_READY | MUX_LANE_FIFO | MUX_PRBSQ | MUX_PRBSI | 0x00 | R/W |
| 0x029 | IRQ_OUTPUT_MUX0 | RESERVED | | | | MUX_DAC0_CAL_DONE | RESERVED | | MUX_PAERR0 | 0x00 | R/W |
| 0x02A | IRQ_OUTPUT_MUX1 | RESERVED | | | | MUX_DAC1_CAL_DONE | RESERVED | | MUX_PAERR1 | 0x00 | R/W |
| 0x02B | IRQ_OUTPUT_MUX2 | RESERVED | | MUX_DLL_LOST | MUX_DLL_LOCK | RESERVED | | MUX_PLL_LOST | MUX_PLL_LOCK | 0x00 | R/W |
| 0x02C | IRQ_STATUS_ALL | RESERVED | | | | | | | IRQ_STATUS_ALL | 0x00 | R/W |
| 0x036 | SYSREF_COUNT | SYSREF_COUNT | | | | | | | | 0x00 | R/W |
| 0x039 | SYSREF_ERR_WINDOW | RESERVED | SYSREF_ERR_WINDOW | | | | | | | 0x00 | R/W |
| 0x03A | SYSREF_MODE | RESERVED | | | SYNC_ROTATION_ | RESERVED | | SYSREF_MODE_ | RESERVED | 0x10 | R/W |

## REGISTER SUMMARY

*Table 54. Register Summary*

| Reg | Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Reset | RW |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | DONE | | | ONESHOT | | | |
| 0x03B | ROTATION_ MODE | SYNCLOGIC_ | RESERVED | PERIODIC_ RST_EN | NCORST_ AFTER_ROT_ EN | RESERVED | | ROTATION_MODE | | 0xB0 | R/W |
| 0x03F | TX_ENABLE | RESERVED | | TXEN_ DATAPATH_ DAC1 | TXEN_ DATAPATH_ DAC0 | RESERVED | | | | 0x00 | R/W |
| 0x050 | CAL_CLK_DIV | RESERVED | | | | CAL_CLK_DIV | | | | 0x28 | R/W |
| 0x051 | CAL_CTRL | CAL_CTRL0 | RESERVED | | | | CAL_CTRL1 | | CAL_START | 0x82 | R/W |
| 0x052 | CAL_STAT | RESERVED | | | | | CAL_ ACTIVE | CAL_FAIL_ SEARCH | CAL_FINISH | 0x00 | R/W |
| 0x059 | FSC0 | RESERVED | | | | | | FSC_CTRL[1:0] | | 0xA0 | R/W |
| 0x05A | FSC1 | FSC_CTRL[9:2] | | | | | | | | 0x28 | R/W |
| 0x061 | CAL_DEBUG0 | RESERVED | CAL_CTRL2 | CAL_CTRL3 | RESERVED | CAL_CTRL4 | RESERVED | | | 0x60 | R/W |
| 0x081 | CLK_CTRL | RESERVED | | | | | | CAL_CLK_ PD1 | CAL_CLK_ PD0 | 0x00 | R/W |
| 0x083 | NVM_CTRL0 | NVM_CTRL0A | RESERVED | | | | | NVM_CTRL0B | | 0x02 | R/W |
| 0x084 | SYSREF_CTRL | RESERVED | SYSREF_ INPUTMODE | RESERVED | | | | | SYSREF_PD | 0x00 | R/W |
| 0x085 | NVM_CTRL1 | RESERVED | NVM_CTRL1A | | | RESERVED | | NVM_ CTRL1B | NVM_ CTRL1C | 0x13 | R/W |
| 0x08D | ADC_CLK_ CTRL0 | RESERVED | | | CLKOUT_SWING | | | | | 0x00 | R/W |
| 0x08F | ADC_CLK_ CTRL2 | RESERVED | | | | | | | PD_ CLKOUT_ DRIVER | 0x00 | R/W |
| 0x090 | DAC_ POWERDOWN | RESERVED | | | | | | DAC_PD1 | DAC_PD0 | 0x03 | R/W |
| 0x091 | ACLK_CTRL | RESERVED | | | | | | | ACLK_ POWER-DOWN | 0x01 | R/W |
| 0x094 | PLL_CLK_DIV | RESERVED | | | | | | PLL_VCO_ DIV3_EN | PLL_VCO_ DIV2_EN | 0x00 | R/W |
| 0x095 | PLL_BYPASS | RESERVED | | | | | | | PLL_ BYPASS | 0x00 | R/W |
| 0x09A | NVM_CTRL | PD_BGR | RESERVED | | | | | | | 0x00 | R/W |
| 0x0C0 | DELAY_LINE_ PD | RESERVED | | DLL_CTRL0B | DLL_CTRL0A | RESERVED | | | DLL_PD | 0x31 | R/W |
| 0x0C1 | DLL_CTRL0 | DLL_CTRL1C | | DLL_CTRL1B | DLL_CTRL1A | | RESERVED | | DLL_ ENABLE | 0x70 | R/W |
| 0x0C3 | DLL_STATUS | RESERVED | | | | | | | DLL_LOCK | 0x00 | R/W |
| 0x0C7 | DLL_READ | RESERVED | | | | | | | DLL_READ_ EN | 0x00 | R/W |
| 0x0CC | DLL_FINE_ DELAY0 | RESERVED | | DLL_FINE_DELAY0 | | | | | | 0x00 | R/W |
| 0x0CD | DLL_FINE_ DELAY1 | RESERVED | | DLL_FINE_DELAY1 | | | | | | 0x00 | R/W |
| 0x0DB | DLL_UPDATE | RESERVED | | | | | | | DLL_ DELAY_ UPDATE | 0x00 | R/W |
| 0x100 | DIG_RESET | RESERVED | | | | | | | DIG_ | 0x01 | R/W |

## REGISTER SUMMARY

*Table 54. Register Summary*

| Reg | Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Reset | RW |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | DATAPATH_PD | | |
| 0x110 | JESD_MODE | MODE_NOT_IN_TABLE | COM_SYNC | JESD_MODE | | | | | | 0x20 | R/W |
| 0x111 | INTRP_MODE | DP_INTERP_MODE | | | | CH_INTERP_MODE | | | | 0x84 | R/W |
| 0x112 | DDSM_DATAPATH_CFG | RESERVED | | DDSM_MODE | | DDSM_NCO_EN | DDSM_MODULUS_EN | DDSM_SEL_SIDEBAND | EN_SYNC_ALL_CHNL_NCO_RESETS | 0x01 | R/W |
| 0x113 | DDSM_FTW_UPDATE | RESERVED | DDSM_FTW_REQ_MODE | | | RESERVED | DDSM_FTW_LOAD_SYSREF | DDSM_FTW_LOAD_ACK | DDSM_FTW_LOAD_REQ | 0x00 | R/W |
| 0x114 | DDSM_FTW0 | DDSM_FTW[7:0] | | | | | | | | 0x00 | R/W |
| 0x115 | DDSM_FTW1 | DDSM_FTW[15:8] | | | | | | | | 0x00 | R/W |
| 0x116 | DDSM_FTW2 | DDSM_FTW[23:16] | | | | | | | | 0x00 | R/W |
| 0x117 | DDSM_FTW3 | DDSM_FTW[31:24] | | | | | | | | 0x00 | R/W |
| 0x118 | DDSM_FTW4 | DDSM_FTW[39:32] | | | | | | | | 0x00 | R/W |
| 0x119 | DDSM_FTW5 | DDSM_FTW[47:40] | | | | | | | | 0x00 | R/W |
| 0x11C | DDSM_PHASE_OFFSET0 | DDSM_NCO_PHASE_OFFSET[7:0] | | | | | | | | 0x00 | R/W |
| 0x11D | DDSM_PHASE_OFFSET1 | DDSM_NCO_PHASE_OFFSET[15:8] | | | | | | | | 0x00 | R/W |
| 0x124 | DDSM_ACC_MODULUS0 | DDSM_ACC_MODULUS[7:0] | | | | | | | | 0x00 | R/W |
| 0x125 | DDSM_ACC_MODULUS1 | DDSM_ACC_MODULUS[15:8] | | | | | | | | 0x00 | R/W |
| 0x126 | DDSM_ACC_MODULUS2 | DDSM_ACC_MODULUS[23:16] | | | | | | | | 0x00 | R/W |
| 0x127 | DDSM_ACC_MODULUS3 | DDSM_ACC_MODULUS[31:24] | | | | | | | | 0x00 | R/W |
| 0x128 | DDSM_ACC_MODULUS4 | DDSM_ACC_MODULUS[39:32] | | | | | | | | 0x00 | R/W |
| 0x129 | DDSM_ACC_MODULUS5 | DDSM_ACC_MODULUS[47:40] | | | | | | | | 0x00 | R/W |
| 0x12A | DDSM_ACC_DELTA0 | DDSM_ACC_DELTA[7:0] | | | | | | | | 0x00 | R/W |
| 0x12B | DDSM_ACC_DELTA1 | DDSM_ACC_DELTA[15:8] | | | | | | | | 0x00 | R/W |
| 0x12C | DDSM_ACC_DELTA2 | DDSM_ACC_DELTA[23:16] | | | | | | | | 0x00 | R/W |
| 0x12D | DDSM_ACC_DELTA3 | DDSM_ACC_DELTA[31:24] | | | | | | | | 0x00 | R/W |
| 0x12E | DDSM_ACC_DELTA4 | DDSM_ACC_DELTA[39:32] | | | | | | | | 0x00 | R/W |
| 0x12F | DDSM_ACC_DELTA5 | DDSM_ACC_DELTA[47:40] | | | | | | | | 0x00 | R/W |
| 0x130 | DDSC_DATAPATH_CFG | RESERVED | DDSC_NCO_EN | RESERVED | | | DDSC_MODULUS_EN | DDSC_SEL_SIDEBAND | DDSC_EN_DC_INPUT | 0x00 | R/W |
| 0x131 | DDSC_FTW_ | RESERVED | | | | | DDSC_ | DDSC_ | DDSC_ | 0x00 | R/W |

## REGISTER SUMMARY

*Table 54. Register Summary*

| Reg | Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Reset | RW |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | UPDATE | | | | | | FTW_ LOAD_ SYSREF | FTW_ | FTW_ LOAD_REQ | | |
| 0x132 | DDSC_FTW0 | DDSC_FTW[7:0] | | | | | | | | 0x00 | R/W |
| 0x133 | DDSC_FTW1 | DDSC_FTW[15:8] | | | | | | | | 0x00 | R/W |
| 0x134 | DDSC_FTW2 | DDSC_FTW[23:16] | | | | | | | | 0x00 | R/W |
| 0x135 | DDSC_FTW3 | DDSC_FTW[31:24] | | | | | | | | 0x00 | R/W |
| 0x136 | DDSC_FTW4 | DDSC_FTW[39:32] | | | | | | | | 0x00 | R/W |
| 0x137 | DDSC_FTW5 | DDSC_FTW[47:40] | | | | | | | | 0x00 | R/W |
| 0x138 | DDSC_PHASE_ OFFSET0 | DDSC_NCO_PHASE_OFFSET[7:0] | | | | | | | | 0x00 | R/W |
| 0x139 | DDSC_PHASE_ OFFSET1 | DDSC_NCO_PHASE_OFFSET[15:8] | | | | | | | | 0x00 | R/W |
| 0x13A | DDSC_ACC_ MODULUS0 | DDSC_ACC_MODULUS[7:0] | | | | | | | | 0x00 | R/W |
| 0x13B | DDSC_ACC_ MODULUS1 | DDSC_ACC_MODULUS[15:8] | | | | | | | | 0x00 | R/W |
| 0x13C | DDSC_ACC_ MODULUS2 | DDSC_ACC_MODULUS[23:16] | | | | | | | | 0x00 | R/W |
| 0x13D | DDSC_ACC_ MODULUS3 | DDSC_ACC_MODULUS[31:24] | | | | | | | | 0x00 | R/W |
| 0x13E | DDSC_ACC_ MODULUS4 | DDSC_ACC_MODULUS[39:32] | | | | | | | | 0x00 | R/W |
| 0x13F | DDSC_ACC_ MODULUS5 | DDSC_ACC_MODULUS[47:40] | | | | | | | | 0x00 | R/W |
| 0x140 | DDSC_ACC_ DELTA0 | DDSC_ACC_DELTA[7:0] | | | | | | | | 0x00 | R/W |
| 0x141 | DDSC_ACC_ DELTA1 | DDSC_ACC_DELTA[15:8] | | | | | | | | 0x00 | R/W |
| 0x142 | DDSC_ACC_ DELTA2 | DDSC_ACC_DELTA[23:16] | | | | | | | | 0x00 | R/W |
| 0x143 | DDSC_ACC_ DELTA3 | DDSC_ACC_DELTA[31:24] | | | | | | | | 0x00 | R/W |
| 0x144 | DDSC_ACC_ DELTA4 | DDSC_ACC_DELTA[39:32] | | | | | | | | 0x00 | R/W |
| 0x145 | DDSC_ACC_ DELTA5 | DDSC_ACC_DELTA[47:40] | | | | | | | | 0x00 | R/W |
| 0x146 | CHNL_GAIN0 | CHNL_GAIN[7:0] | | | | | | | | 0x00 | R/W |
| 0x147 | CHNL_GAIN1 | RESERVED | | | | | CHNL_GAIN[11:8] | | | 0x08 | R/W |
| 0x148 | DC_CAL_TONE0 | DC_TEST_INPUT_AMPLITUDE[7:0] | | | | | | | | 0x00 | R/W |
| 0x149 | DC_CAL_TONE1 | DC_TEST_INPUT_AMPLITUDE[15:8] | | | | | | | | 0x00 | R/W |
| 0x14B | PRBS | 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 | PRBS_COUNT_I | | | | | | | | 0x00 | R |
| 0x14D | PRBS_ERROR_ Q | PRBS_COUNT_Q | | | | | | | | 0x00 | R |
| 0x14E | PRBS_CHANSEL | RESERVED | | | | | PRBS_CHANSEL | | | 0x07 | R/W |
| 0x151 | DECODE_MODE | RESERVED | | MSB_MODE | | RESERVED | | | | 0x00 | R/W |
| 0x1DE | SPI_ENABLE | RESERVED | | | | | | SPI_EN1 | SPI_EN0 | 0x03 | R/W |
| 0x1E2 | DDSM_CAL_ | DDSM_CAL_FTW[7:0] | | | | | | | | 0x00 | R/W |

## REGISTER SUMMARY

*Table 54. Register Summary*

| Reg | Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Reset | RW |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | FTW0 | | | | | | | | | | |
| 0x1E3 | DDSM_CAL_FTW1 | DDSM_CAL_FTW[15:8] | | | | | | | | 0x00 | R/W |
| 0x1E4 | DDSM_CAL_FTW2 | DDSM_CAL_FTW[23:16] | | | | | | | | 0x00 | R/W |
| 0x1E5 | DDSM_CAL_FTW3 | DDSM_CAL_FTW[31:24] | | | | | | | | 0x00 | R/W |
| 0x1E6 | DDSM_CAL_MODE_DEF | RESERVED | | | | | DDSM_EN_CAL_ACC | DDSM_EN_CAL_DC_INPUT | DDSM_EN_CAL_FREQ_TUNE | 0x00 | R/W |
| 0x1E7 | DATAPATH_NCO_SYNC_CFG | RESERVED | | | | | | ALL_NCO_SYNC_ACK | START_NCO_SYNC | 0x00 | R/W |
| 0x200 | MASTER_PD | RESERVED | | | | | | | SERDES_MASTER_PD | 0x01 | R/W |
| 0x201 | PHY_PD | PHY_PD | | | | | | | | 0xEE | R/W |
| 0x203 | GENERIC_PD | RESERVED | | | | | | PD_SYNCOUT0 | PD_SYNCOUT1 | 0x01 | R/W |
| 0x206 | CDR_RESET | RESERVED | | | | | | | CDR_PHY_RESET | 0x00 | R/W |
| 0x210 | CBUS_ADDR | SERDES_CBUS_ADDR | | | | | | | | 0x00 | R/W |
| 0x212 | CBUS_WRSTROBE_PHY | SERDES_CBUS_WR0 | | | | | | | | 0x00 | R/W |
| 0x213 | CBUS_WRSTROBE_OTHER | RESERVED | | | | | | | SERDES_CBUS_WR1 | 0x00 | R/W |
| 0x216 | CBUS_WDATA | SERDES_CBUS_DATA | | | | | | | | 0x00 | R/W |
| 0x234 | CDR_BITINVERSE | SEL_IF_PARDATAINV_DES_RC_CH | | | | | | | | 0x66 | R/W |
| 0x240 | EQ_BOOST_PHY_3_0 | EQ_BOOST_PHY3 | | | EQ_BOOST_PHY2 | | EQ_BOOST_PHY1 | | EQ_BOOST_PHY0 | 0xFF | R/W |
| 0x241 | EQ_BOOST_PHY_7_4 | EQ_BOOST_PHY7 | | | EQ_BOOST_PHY6 | | EQ_BOOST_PHY5 | | EQ_BOOST_PHY4 | 0xFF | R/W |
| 0x242 | EQ_GAIN_PHY_3_0 | EQ_GAIN_PHY3 | | | EQ_GAIN_PHY2 | | EQ_GAIN_PHY1 | | EQ_GAIN_PHY0 | 0xFF | R/W |
| 0x243 | EQ_GAIN_PHY_7_4 | EQ_GAIN_PHY7 | | | EQ_GAIN_PHY6 | | EQ_GAIN_PHY5 | | EQ_GAIN_PHY4 | 0xFF | R/W |
| 0x244 | EQ_FB_PHY_0 | RESERVED | | | EQ_PHY_0 | | | | | 0x19 | R/W |
| 0x245 | EQ_FB_PHY_1 | RESERVED | | | EQ_PHY1 | | | | | 0x19 | R/W |
| 0x246 | EQ_FB_PHY_2 | RESERVED | | | EQ_PHY2 | | | | | 0x19 | R/W |
| 0x247 | EQ_FB_PHY_3 | RESERVED | | | EQ_PHY3 | | | | | 0x19 | R/W |
| 0x248 | EQ_FB_PHY_4 | RESERVED | | | EQ_PHY4 | | | | | 0x19 | R/W |
| 0x249 | EQ_FB_PHY_5 | RESERVED | | | EQ_PHY5 | | | | | 0x19 | R/W |
| 0x24A | EQ_FB_PHY_6 | RESERVED | | | EQ_PHY6 | | | | | 0x19 | R/W |
| 0x24B | EQ_FB_PHY_7 | RESERVED | | | EQ_PHY7 | | | | | 0x19 | R/W |
| 0x250 | LBT_REG_CNTRL_0 | EN_LBT_DES_RC_CH | | | | | | | | 0x00 | R/W |
| 0x251 | LBT_REG_CNTRL_1 | RESERVED | | | | | | EN_LBT_ | INIT_LBT_ | 0x02 | R/W |

## REGISTER SUMMARY

*Table 54. Register Summary*

| Reg | Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Reset | RW |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | HALFRATE_DES_RC | SYNC_DES_RC | | |
| 0x253 | $\overline{\text{SYNCOUT0}}$_CTRL | RESERVED | | | | | | | SEL_$\overline{\text{SYNCOUT0}}$_MODE | 0x00 | R/W |
| 0x254 | $\overline{\text{SYNCOUT1}}$_CTRL | RESERVED | | | | | | | SEL_$\overline{\text{SYNCOUT1}}$_MODE | 0x00 | R/W |
| 0x280 | PLL_ENABLE_CTRL | RESERVED | | | | | LOLSTICKY CLEAR_LCPLL_RC | LDSYNTH_LCPLL_RC | SERDES_PLL_STARTUP | 0x01 | R/W |
| 0x281 | PLL_STATUS | RESERVED | | | | | | | SERDES_PLL_LOCK | 0x00 | R |
| 0x300 | GENERAL_JRX_CTRL_0 | RESERVED | | | | LINK_MODE | LINK_PAGE | LINK_EN | | 0x00 | R/W |
| 0x302 | DYN_LINK_LATENCY_0 | RESERVED | | DYN_LINK_LATENCY_0 | | | | | | 0x00 | R |
| 0x303 | DYN_LINK_LATENCY_1 | RESERVED | | DYN_LINK_LATENCY_1 | | | | | | 0x00 | R |
| 0x304 | LMFC_DELAY_0 | RESERVED | | LMFC_DELAY_0 | | | | | | 0x00 | R/W |
| 0x305 | LMFC_DELAY_1 | RESERVED | | LMFC_DELAY_1 | | | | | | 0x00 | R/W |
| 0x306 | LMFC_VAR_0 | RESERVED | | LMFC_VAR_0 | | | | | | 0x3F | R/W |
| 0x307 | LMFC_VAR_1 | RESERVED | | LMFC_VAR_1 | | | | | | 0x3F | R/W |
| 0x308 | XBAR_LN_0_1 | RESERVED | | LOGICAL_LANE1_SRC | | | LOGICAL_LANE0_SRC | | | 0x08 | R/W |
| 0x309 | XBAR_LN_2_3 | RESERVED | | LOGICAL_LANE3_SRC | | | LOGICAL_LANE2_SRC | | | 0x1A | R/W |
| 0x30A | XBAR_LN_4_5 | RESERVED | | LOGICAL_LANE5_SRC | | | LOGICAL_LANE4_SRC | | | 0x2C | R/W |
| 0x30B | XBAR_LN_6_7 | RESERVED | | LOGICAL_LANE7_SRC | | | LOGICAL_LANE6_SRC | | | 0x3E | R/W |
| 0x30C | FIFO_STATUS_REG_0 | LANE_FIFO_FULL | | | | | | | | 0x00 | R |
| 0x30D | FIFO_STATUS_REG_1 | LANE_FIFO_EMPTY | | | | | | | | 0x00 | R |
| 0x311 | $\overline{\text{SYNCOUT}}$_GEN_0 | RESERVED | | | | EOMF_MASK_1 | EOMF_MASK_0 | EOF_MASK_1 | EOF_MASK_0 | 0x00 | R/W |
| 0x312 | $\overline{\text{SYNCOUT}}$_GEN_1 | $\overline{\text{SYNC}}$_ERR_DUR | | | | RESERVED | | | | 0x00 | R/W |
| 0x315 | PHY_PRBS_TEST_EN | PHY_TEST_EN | | | | | | | | 0x00 | R/W |
| 0x316 | PHY_PRBS_TEST_CTRL | RESERVED | PHY_SRC_ERR_CNT | | | PHY_PRBS_PAT_SEL | | PHY_TEST_START | PHY_TEST_RESET | 0x00 | R/W |
| 0x317 | PHY_PRBS_TEST_THRESHOLD_LOBITS | PHY_PRBS_THRESHOLD_LOBITS | | | | | | | | 0x00 | R/W |
| 0x318 | PHY_PRBS_TEST_THRESHOLD_MIDBITS | PHY_PRBS_THRESHOLD_MIDBITS | | | | | | | | 0x00 | R/W |
| 0x319 | PHY_PRBS_TEST_THRESHOLD_HIBITS | PHY_PRBS_THRESHOLD_HIBITS | | | | | | | | 0x00 | R/W |
| 0x31A | PHY_ | PHY_PRBS_ERR_CNT_LOBITS | | | | | | | | 0x00 | R |

## REGISTER SUMMARY

*Table 54. Register Summary*

| Reg | Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Reset | RW |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | PRBS_TEST_ERRCNT_LOBITS | | | | | | | | | | |
| 0x31B | PHY_PRBS_TEST_ERRCNT_MIDBITS | PHY_PRBS_ERR_CNT_MIDBITS | | | | | | | | 0x00 | R |
| 0x31C | PHY_PRBS_TEST_ERRCNT_HIBITS | PHY_PRBS_ERR_CNT_HIBITS | | | | | | | | 0x00 | R |
| 0x31D | PHY_PRBS_TEST_STATUS | PHY_PRBS_PASS | | | | | | | | 0xFF | R |
| 0x31E | PHY_DATA_SNAPSHOT_CTRL | RESERVED | | | | | | PHY_GRAB_MODE | PHY_GRAB_DATA | 0x00 | R/W |
| 0x31F | PHY_SNAPSHOT_DATA_BYTE0 | PHY_SNAPSHOT_DATA_BYTE0 | | | | | | | | 0x00 | R |
| 0x320 | PHY_SNAPSHOT_DATA_BYTE1 | PHY_SNAPSHOT_DATA_BYTE1 | | | | | | | | 0x00 | R |
| 0x321 | PHY_SNAPSHOT_DATA_BYTE2 | PHY_SNAPSHOT_DATA_BYTE2 | | | | | | | | 0x00 | R |
| 0x322 | PHY_SNAPSHOT_DATA_BYTE3 | PHY_SNAPSHOT_DATA_BYTE3 | | | | | | | | 0x00 | R |
| 0x323 | PHY_SNAPSHOT_DATA_BYTE4 | PHY_SNAPSHOT_DATA_BYTE4 | | | | | | | | 0x00 | R |
| 0x32C | SHORT_TPL_TEST_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 | SHORT_TPL_REF_SP_LSB | | | | | | | | 0x00 | R/W |
| 0x32E | SHORT_TPL_TEST_2 | SHORT_TPL_REF_SP_MSB | | | | | | | | 0x00 | R/W |
| 0x32F | SHORT_TPL_TEST_3 | RESERVED | SHORT_TPL_IQ_SAMPLE_SEL | RESERVED | | | | | SHORT_TPL_FAIL | 0x00 | R/W |
| 0x334 | JESD_BIT_INVERSE_CTRL | JESD_BIT_INVERSE | | | | | | | | 0x00 | R/W |
| 0x400 | DID_REG | DID_RD | | | | | | | | 0x00 | R |
| 0x401 | BID_REG | BID_RD | | | | | | | | 0x00 | R |
| 0x402 | LID0_REG | RESERVED | ADJDIR_RD | PHADJ_RD | LL_LID0 | | | | | 0x00 | R |
| 0x403 | SCR_L_REG | SCR_RD | RESERVED | | L_RD-1 | | | | | 0x00 | R |
| 0x404 | F_REG | F_RD-1 | | | | | | | | 0x00 | R |
| 0x405 | K_REG | RESERVED | | | K_RD-1 | | | | | 0x00 | R |
| 0x406 | M_REG | M_RD-1 | | | | | | | | 0x00 | R |
| 0x407 | CS_N_REG | CS_RD | | RESERVED | N_RD-1 | | | | | 0x00 | R |
| 0x408 | NP_REG | SUBCLASSV_RD | | | NP_RD-1 | | | | | 0x00 | R |
| 0x409 | S_REG | JESDV_RD-1 | | | S_RD-1 | | | | | 0x00 | R |

## REGISTER SUMMARY

*Table 54. Register Summary*

| Reg | Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Reset | RW |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x40A | HD_CF_REG | HD_RD | RESERVED | | | | CF_RD | | | 0x00 | R |
| 0x40B | RES1_REG | RES1_RD | | | | | | | | 0x00 | R |
| 0x40C | RES2_REG | RES2_RD | | | | | | | | 0x00 | R |
| 0x40D | CHECKSUM0_REG | LL_FCHK0 | | | | | | | | 0x00 | R |
| 0x40E | COMPSUM0_REG | LL_FCMP0 | | | | | | | | 0x00 | R |
| 0x412 | LID1_REG | RESERVED | | | | | LL_LID1 | | | 0x00 | R |
| 0x415 | CHECKSUM1_REG | LL_FCHK1 | | | | | | | | 0x00 | R |
| 0x416 | COMPSUM1_REG | LL_FCMP1 | | | | | | | | 0x00 | R |
| 0x41A | LID2_REG | RESERVED | | | | | LL_LID2 | | | 0x00 | R |
| 0x41D | CHECKSUM2_REG | LL_FCHK2 | | | | | | | | 0x00 | R |
| 0x41E | COMPSUM2_REG | LL_FCMP2 | | | | | | | | 0x00 | R |
| 0x422 | LID3_REG | RESERVED | | | | | LL_LID3 | | | 0x00 | R |
| 0x425 | CHECKSUM3_REG | LL_FCHK3 | | | | | | | | 0x00 | R |
| 0x426 | COMPSUM3_REG | LL_FCMP3 | | | | | | | | 0x00 | R |
| 0x42A | LID4_REG | RESERVED | | | | | LL_LID4 | | | 0x00 | R |
| 0x42D | CHECKSUM4_REG | LL_FCHK4 | | | | | | | | 0x00 | R |
| 0x42E | COMPSUM4_REG | LL_FCMP4 | | | | | | | | 0x00 | R |
| 0x432 | LID5_REG | RESERVED | | | | | LL_LID5 | | | 0x00 | R |
| 0x435 | CHECKSUM5_REG | LL_FCHK5 | | | | | | | | 0x00 | R |
| 0x436 | COMPSUM5_REG | LL_FCMP5 | | | | | | | | 0x00 | R |
| 0x43A | LID6_REG | RESERVED | | | | | LL_LID6 | | | 0x00 | R |
| 0x43D | CHECKSUM6_REG | LL_FCHK6 | | | | | | | | 0x00 | R |
| 0x43E | COMPSUM6_REG | LL_FCMP6 | | | | | | | | 0x00 | R |
| 0x442 | LID7_REG | RESERVED | | | | | LL_LID7 | | | 0x00 | R |
| 0x445 | CHECKSUM7_REG | LL_FCHK7 | | | | | | | | 0x00 | R |
| 0x446 | COMPSUM7_REG | LL_FCMP7 | | | | | | | | 0x00 | R |
| 0x450 | ILS_DID | DID | | | | | | | | 0x00 | R/W |
| 0x451 | ILS_BID | BID | | | | | | | | 0x00 | R/W |
| 0x452 | ILS_LID0 | RESERVED | ADJDIR | PHADJ | | | LID0 | | | 0x00 | R/W |
| 0x453 | ILS_SCR_L | SCR | RESERVED | | | | L-1 | | | 0x87 | R/W |
| 0x454 | ILS_F | F-1 | | | | | | | | 0x00 | R/W |
| 0x455 | ILS_K | RESERVED | | | | | K-1 | | | 0x1F | R/W |
| 0x456 | ILS_M | M-1 | | | | | | | | 0x01 | R/W |
| 0x457 | ILS_CS_N | CS | | RESERVED | | | N-1 | | | 0x0F | R/W |

## REGISTER SUMMARY

*Table 54. Register Summary*

| Reg | Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Reset | RW |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x458 | ILS_NP | SUBCLASSV | | | NP-1 | | | | | 0x0F | R/W |
| 0x459 | ILS_S | JESDV | | | S-1 | | | | | 0x01 | R/W |
| 0x45A | ILS_HD_CF | HD | RESERVED | | CF | | | | | 0x80 | R |
| 0x45B | ILS_RES1 | RES1 | | | | | | | | 0x00 | R/W |
| 0x45C | ILS_RES2 | RES2 | | | | | | | | 0x00 | R/W |
| 0x45D | ILS_CHECKSUM | FCHK0 | | | | | | | | 0x00 | R/W |
| 0x46C | LANE_DESKEW | ILD7 | ILD6 | ILD5 | ILD4 | ILD3 | ILD2 | ILD1 | ILD0 | 0x00 | R |
| 0x46D | BAD_DISPARITY | BDE7 | BDE6 | BDE5 | BDE4 | BDE3 | BDE2 | BDE1 | BDE0 | 0x00 | R |
| 0x46E | NOT_IN_TABLE | NIT7 | NIT6 | NIT5 | NIT4 | NIT3 | NIT2 | NIT1 | NIT0 | 0x00 | R |
| 0x46F | UNEXPECTED_KCHAR | UEK7 | UEK6 | UEK5 | UEK4 | UEK3 | UEK2 | UEK1 | UEK0 | 0x00 | R |
| 0x470 | CODE_GRP_SYNC | CGS7 | CGS6 | CGS5 | CGS4 | CGS3 | CGS2 | CGS1 | CGS0 | 0x00 | R |
| 0x471 | FRAME_SYNC | FS7 | FS6 | FS5 | FS4 | FS3 | FS2 | FS1 | FS0 | 0x00 | R |
| 0x472 | GOOD_CHECKSUM | CKS7 | CKS6 | CKS5 | CKS4 | CKS3 | CKS2 | CKS1 | CKS0 | 0x00 | R |
| 0x473 | INIT_LANE_SYNC | ILS7 | ILS6 | ILS5 | ILS4 | ILS3 | ILS2 | ILS1 | ILS0 | 0x00 | R |
| 0x475 | CTRLREG0 | RESERVED | | | | SOFTRST | FORCE-SYNCREQ | RESERVED | REPL_FRM_ENA | 0x01 | R/W |
| 0x476 | CTRLREG1 | RESERVED | | | QUAL_RDERR | RESERVED | | | FCHK_N | 0x14 | R/W |
| 0x477 | CTRLREG2 | ILS_MODE | RESERVED | REPDATATEST | QUETESTERR | AR_ECNTR | RESERVED | | | 0x00 | R/W |
| 0x478 | KVAL | KSYNC | | | | | | | | 0x01 | R/W |
| 0x47C | ERRORTHRES | ETH | | | | | | | | 0xFF | R/W |
| 0x47D | SYNC_ASSERT_MASK | RESERVED | | | | | SYNC_ASSERT_MASK | | | 0x07 | R/W |
| 0x480 | ECNT_CTRL0 | RESERVED | | ECNT_ENA0 | | | ECNT_RST0 | | | 0x3F | R/W |
| 0x481 | ECNT_CTRL1 | RESERVED | | ECNT_ENA1 | | | ECNT_RST1 | | | 0x3F | R/W |
| 0x482 | ECNT_CTRL2 | RESERVED | | ECNT_ENA2 | | | ECNT_RST2 | | | 0x3F | R/W |
| 0x483 | ECNT_CTRL3 | RESERVED | | ECNT_ENA3 | | | ECNT_RST3 | | | 0x3F | R/W |
| 0x484 | ECNT_CTRL4 | RESERVED | | ECNT_ENA4 | | | ECNT_RST4 | | | 0x3F | R/W |
| 0x485 | ECNT_CTRL5 | RESERVED | | ECNT_ENA5 | | | ECNT_RST5 | | | 0x3F | R/W |
| 0x486 | ECNT_CTRL6 | RESERVED | | ECNT_ENA6 | | | ECNT_RST6 | | | 0x3F | R/W |
| 0x487 | ECNT_CTRL7 | RESERVED | | ECNT_ENA7 | | | ECNT_RST7 | | | 0x3F | R/W |
| 0x488 | ECNT_TCH0 | RESERVED | | | | | ECNT_TCH0 | | | 0x07 | R/W |
| 0x489 | ECNT_TCH1 | RESERVED | | | | | ECNT_TCH1 | | | 0x07 | R/W |
| 0x48A | ECNT_TCH2 | RESERVED | | | | | ECNT_TCH2 | | | 0x07 | R/W |
| 0x48B | ECNT_TCH3 | RESERVED | | | | | ECNT_TCH3 | | | 0x07 | R/W |
| 0x48C | ECNT_TCH4 | RESERVED | | | | | ECNT_TCH4 | | | 0x07 | R/W |
| 0x48D | ECNT_TCH5 | RESERVED | | | | | ECNT_TCH5 | | | 0x07 | R/W |
| 0x48E | ECNT_TCH6 | RESERVED | | | | | ECNT_TCH6 | | | 0x07 | R/W |
| 0x48F | ECNT_TCH7 | RESERVED | | | | | ECNT_TCH7 | | | 0x07 | R/W |
| 0x490 | ECNT_STAT0 | RESERVED | | | | LANE_ENA0 | ECNT_TCR0 | | | 0x00 | R |
| 0x491 | ECNT_STAT1 | RESERVED | | | | LANE_ENA1 | ECNT_TCR1 | | | 0x00 | R |
| 0x492 | ECNT_STAT2 | RESERVED | | | | LANE_ENA2 | ECNT_TCR2 | | | 0x00 | R |
| 0x493 | ECNT_STAT3 | RESERVED | | | | LANE_ENA3 | ECNT_TCR3 | | | 0x00 | R |
| 0x494 | ECNT_STAT4 | RESERVED | | | | LANE_ENA4 | ECNT_TCR4 | | | 0x00 | R |

## REGISTER SUMMARY

*Table 54. Register Summary*

| Reg | Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Reset | RW |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x495 | ECNT_STAT5 | RESERVED | | | | LANE_ENA5 | ECNT_TCR5 | | | 0x00 | R |
| 0x496 | ECNT_STAT6 | RESERVED | | | | LANE_ENA6 | ECNT_TCR6 | | | 0x00 | R |
| 0x497 | ECNT_STAT7 | RESERVED | | | | LANE_ENA7 | ECNT_TCR7 | | | 0x00 | R |
| 0x4B0 | LINK_STATUS0 | BDE0 | NIT0 | UEK0 | ILD0 | ILS0 | CKS0 | FS0 | CGS0 | 0x00 | R |
| 0x4B1 | LINK_STATUS1 | BDE1 | NIT1 | UEK1 | ILD1 | ILS1 | CKS1 | FS1 | CGS1 | 0x00 | R |
| 0x4B2 | LINK_STATUS2 | BDE2 | NIT2 | UEK2 | ILD2 | ILS2 | CKS2 | FS2 | CGS2 | 0x00 | R |
| 0x4B3 | LINK_STATUS3 | BDE3 | NIT3 | UEK3 | ILD3 | ILS3 | CKS3 | FS3 | CGS3 | 0x00 | R |
| 0x4B4 | LINK_STATUS4 | BDE4 | NIT4 | UEK4 | ILD4 | ILS4 | CKS4 | FS4 | CGS4 | 0x00 | R |
| 0x4B5 | LINK_STATUS5 | BDE5 | NIT5 | UEK5 | ILD5 | ILS5 | CKS5 | FS5 | CGS5 | 0x00 | R |
| 0x4B6 | LINK_STATUS6 | BDE6 | NIT6 | UEK6 | ILD6 | ILS6 | CKS6 | FS6 | CGS6 | 0x00 | R |
| 0x4B7 | LINK_STATUS7 | BDE7 | NIT7 | UEK7 | ILD7 | ILS7 | CKS7 | FS7 | CGS7 | 0x00 | R |
| 0x4B8 | JESD_IRQ_ENABLEA | EN_BDE | EN_NIT | EN_UEK | EN_ILD | EN_ILS | EN_CKS | EN_FS | EN_CGS | 0x00 | R/W |
| 0x4B9 | JESD_IRQ_ENABLEB | RESERVED | | | | | | | EN_ILAS | 0x00 | R/W |
| 0x4BA | JESD_IRQ_STATUSA | IRQ_BDE | IRQ_NIT | IRQ_UEK | IRQ_ILD | IRQ_ILS | IRQ_CKS | IRQ_FS | IRQ_CGS | 0x00 | R/W |
| 0x4BB | JESD_IRQ_STATUSB | RESERVED | | | | | | | IRQ_ILAS | 0x00 | R/W |
| 0x4BC | IRQ_OUTPUT_MUX_JESD | RESERVED | | | | | | | MUX_JESD_IRQ | 0x00 | R/W |
| 0x580 | BE_SOFT_OFF_GAIN_CTRL | BE_SOFT_OFF_GAIN_EN | RESERVED | | | | BE_GAIN_RAMP_RATE | | | 0x00 | R/W |
| 0x581 | BE_SOFT_OFF_ENABLE | ENA_SHORT_PAERR_SOFTOFF | ENA_LONG_PAERR_SOFTOFF | RESERVED | | ENA_JESD_ERR_SOFTOFF | ROTATE_SOFT_OFF_EN | TXEN_SOFT_OFF_EN | SPI_SOFT_OFF_EN | 0xC6 | R/W |
| 0x582 | BE_SOFT_ON_ENABLE | SPI_SOFT_ON_EN | LONG_LEVEL_SOFTON_EN | RESERVED | | | | | | 0x40 | R/W |
| 0x583 | LONG_PA_THRES_LSB | LONG_PA_THRESHOLD[7:0] | | | | | | | | 0x00 | R/W |
| 0x584 | LONG_PA_THRES_MSB | RESERVED | | | LONG_PA_THRESHOLD[12:8] | | | | | 0x00 | R/W |
| 0x585 | LONG_PA_CONTROL | LONG_PA_ENABLE | RESERVED | | | LONG_PA_AVG_TIME | | | | 0x00 | R/W |
| 0x586 | LONG_PA_POWER_LSB | LONG_PA_POWER[7:0] | | | | | | | | 0x00 | R |
| 0x587 | LONG_PA_POWER_MSB | RESERVED | | | LONG_PA_POWER[12:8] | | | | | 0x00 | R |
| 0x588 | SHORT_PA_THRES_LSB | SHORT_PA_THRESHOLD[7:0] | | | | | | | | 0x00 | R/W |
| 0x589 | SHORT_PA_THRES_MSB | RESERVED | | | SHORT_PA_THRESHOLD[12:8] | | | | | 0x00 | R/W |
| 0x58A | SHORT_PA_CONTROL | SHORT_PA_ENABLE | RESERVED | | | | | SHORT_PA_AVG_TIME | | 0x00 | R/W |
| 0x58B | SHORT_PA_POWER_LSB | SHORT_PA_POWER[7:0] | | | | | | | | 0x00 | R |
| 0x58C | SHORT_PA_POWER_MSB | RESERVED | | | SHORT_PA_POWER[12:8] | | | | | 0x00 | R |

## REGISTER SUMMARY

*Table 54. Register Summary*

| Reg | Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Reset | RW |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x58D | TXEN_SM_0 | RESERVED | | | | | | | ENA_TXENSM | 0x50 | R/W |
| 0x596 | BLANKING_CTRL | RESERVED | | | | SPI_TXEN | ENA_SPI_TXEN | RESERVED | | 0x00 | R/W |
| 0x597 | JESD_PA_INT0 | JESD_PA_INT_CNTRL[7:0] | | | | | | | | 0x00 | R/W |
| 0x598 | JESD_PA_INT1 | RESERVED | | | | | | | JESD_PA_INT_CNTRL[8] | 0x00 | R/W |
| 0x599 | TXEN_FLUSH_CTRL0 | RESERVED | | | | | | | SPI_FLUSH_EN | 0x01 | R/W |
| 0x705 | NVM_LOADER_EN | RESERVED | | | | | | NVM_BLR_DONE | NVM_BLR_EN | 0x00 | R/W |
| 0x790 | DACPLL_PDCTRL0 | PLL_PD5 | PLL_PD4 | | | PLL_PD3 | PLL_PD2 | PLL_PD1 | PLL_PD0 | 0x02 | R/W |
| 0x791 | DACPLL_PDCTRL1 | RESERVED | | | PLL_PD10 | PLL_PD9 | PLL_PD8 | PLL_PD7 | PLL_PD6 | 0x00 | R/W |
| 0x792 | DACPLL_CTRL0 | RESERVED | | | | | | D_CAL_RESET | D_RESET_VCO_DIV | 0x02 | R/W |
| 0x793 | DACPLL_CTRL1 | RESERVED | | | | | | M_DIVIDER-1 | | 0x18 | R/W |
| 0x794 | DACPLL_CTRL2 | RESERVED | | DACPLL_CP | | | | | | 0x04 | R/W |
| 0x795 | DACPLL_CTRL3 | RESERVED | | | | D_CP_CALBITS | | | | 0x08 | R/W |
| 0x796 | DACPLL_CTRL4 | PLL_CTRL0 | | | | RESERVED | | | | 0xD2 | R/W |
| 0x797 | DACPLL_CTRL5 | RESERVED | | PLL_CTRL1 | | | | | | 0x20 | R/W |
| 0x798 | DACPLL_CTRL6 | RESERVED | PLL_CTRL3 | PLL_CTRL2 | | | | | | 0x1C | R/W |
| 0x799 | DACPLL_CTRL7 | ADC_CLK_DIVIDER | | N_DIVIDER | | | | | | 0x08 | R/W |
| 0x7A0 | DACPLL_CTRL9 | RESERVED | | D_EN_VAR_FINE_PRE | RESERVED | | D_EN_VAR_COARSE_PRE | RESERVED | | 0x90 | R/W |
| 0x7A2 | DACPLL_CTRL10 | RESERVED | D_REGULATOR_CAL_WAIT | | D_VCO_CAL_WAIT | | D_VCO_CAL_CYCLES | | RESERVED | 0x35 | R/W |
| 0x7B5 | PLL_STATUS | RESERVED | | | | | | | PLL_LOCK | 0x00 | R |

## REGISTER DETAILS

*Table 55. Register Details*

| Addr. | Name | Bits | Bit Name | Settings | Description | Reset | Access |
|---|---|---|---|---|---|---|---|
| 0x000 | SPI_INTFCONFA | 7 | SOFTRESET_M | | Soft reset (mirror). Set this bit to mirror Bit 0. | 0x0 | R |
| | | 6 | LSBFIRST_M | | LSB first (mirror). Set this bit to mirror Bit 1. | 0x0 | R |
| | | 5 | ADDRINC_M | | Address increment (mirror). Set this bit to mirror Bit 2. | 0x0 | R |
| | | 4 | SDOACTIVE_M | | SDO active (mirror). Set this bit 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, this bit 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, this bit causes SPI 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 autoclears 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. | | |
| | | [5:0] | RESERVED | | Reserved. | 0x0 | R/W |
| 0x003 | SPI_CHIPTYPE | [7:0] | CHIP_TYPE | | Chip type. | 0x4 | R |
| 0x004 | SPI_PRODIDL | [7:0] | PROD_ID[7:0] | | Product ID. | 0x71 | R |
| 0x005 | SPI_PRODIDH | [7:0] | PROD_ID[15:8] | | Product ID. | 0x91 | R |
| 0x006 | SPI_CHIPGRADE | [7:4] | PROD_GRADE | | Product grade. | 0x0 | R |
| | | [3:0] | DEV_REVISION | | Device revision. | 0x02 | R |
| 0x008 | SPI_PAGEINDX | [7:6] | MAINDAC_PAGE | | Sets the main DAC paging. Each high bit in this field pages a DAC starting at the LSB. Both main DACs can be paged and programmed at the same time if desired. | 0x3 | R/W |
| | | [5:0] | CHANNEL_PAGE | | Sets channel paging. Bit 0 and Bit 3 in this field pages a complex channel. Multiple channels can be paged and programmed at a time if desired. Bit 0 pages Channel 0 datapath (for DAC 0). Bit 3 pages Channel 1 datapath (for DAC 1). | 0x3F | R/W |
| 0x00A | SPI_SCRATCHPAD | [7:0] | SCRATCHPAD | | Scratch pad read/write register. | 0x0 | R/W |
| 0x010 | CHIP_ID_L | [7:0] | CHIP_ID[7:0] | | Chip ID serial number. | 0x0 | R |
| 0x011 | CHIP_ID_M1 | [7:0] | CHIP_ID[15:8] | | Chip ID serial number. | 0x0 | R |
| 0x012 | CHIP_ID_M2 | [7:0] | CHIP_ID[23:16] | | Chip ID serial number. | 0x0 | R |
| 0x013 | CHIP_ID_H | [7:0] | CHIP_ID[31:24] | | Chip ID serial number. | 0x0 | R |
| 0x020 | IRQ_ENABLE | [7:5] | RESERVED | | Reserved. | 0x0 | R |
| | | 4 | EN_SYSREF_JITTER | | Enable SYSREF± jitter interrupt. | 0x0 | R/W |

## REGISTER DETAILS

*Table 55. Register Details*

| Addr. | Name | Bits | Bit Name | Settings | Description | Reset | Access |
|---|---|---|---|---|---|---|---|
| | | 3 | EN_DATA_READY | | Enable JESD204B receiver ready (JRX_DATA_READY) low interrupt. | 0x0 | R/W |
| | | 2 | EN_LANE_FIFO | | Enable lane FIFO overflow/underflow interrupt. | 0x0 | R/W |
| | | 1 | EN_PRBSQ | | Enable PRBS imaginary error interrupt. | 0x0 | R/W |
| | | 0 | EN_PRBSI | | Enable PRBS real error interrupt. | 0x0 | R/W |
| 0x021 | IRQ_ENABLE0 | [7:4] | RESERVED | | Reserved. | 0x0 | R |
| | | 3 | EN_DAC0_CAL_DONE | | Enable DAC0 calibration complete interrupt. | 0x0 | R/W |
| | | [2:1] | RESERVED | | Reserved. | 0x0 | R/W |
| | | 0 | EN_PAERR0 | | Enable PA protection error interrupt for DAC0. | 0x0 | R/W |
| 0x022 | IRQ_ENABLE1 | [7:4] | RESERVED | | Reserved. | 0x0 | R |
| | | 3 | EN_DAC1_CAL_DONE | | Enable DAC1 calibration complete interrupt. | 0x0 | R/W |
| | | [2:1] | RESERVED | | Reserved. | 0x0 | R/W |
| | | 0 | EN_PAERR1 | | Enable PA protection error interrupt for DAC1. | 0x0 | R/W |
| 0x023 | IRQ_ENABLE2 | [7:6] | RESERVED | | Reserved. | 0x0 | R |
| | | 5 | EN_DLL_LOST | | Enable DLL lock lost interrupt. | 0x0 | R/W |
| | | 4 | EN_DLL_LOCK | | Enable DLL lock interrupt. | 0x0 | R/W |
| | | [3:2] | RESERVED | | Reserved. | 0x0 | R/W |
| | | 1 | EN_PLL_LOST | | Enable PLL lock lost interrupt. | 0x0 | R/W |
| | | 0 | EN_PLL_LOCK | | Enable PLL lock interrupt. | 0x0 | R/W |
| 0x024 | IRQ_STATUS | [7:5] | RESERVED | | Reserved. | 0x0 | R |
| | | 4 | IRQ_SYSREF_JITTER | | SYSREF± jitter too large. If EN_SYSREF_JITTER is low, IRQ_SYSREF_JITTER shows the current status. If EN_SYSREF_JITTER is high, IRQ_SYSREF_JITTER latches and pulls the $\overline{IRQx}$ pin low (x = the MUX_SYSREF_JITTER setting). Writing a 1 to IRQ_SYSREF_JITTER when latched clears the bit. | 0x0 | R/W |
| | | 3 | IRQ_DATA_READY | | JESD204x receiver data ready is low. If EN_DATA_READY is low, IRQ_DATA_READY shows the current status. If EN_DATA_READY is high, IRQ_DATA_READY latches and pulls the $\overline{IRQx}$ pin low (x = MUX_DATA_READY setting). Writing a 1 to IRQ_DATA_READY when latched clears the bit. | 0x0 | R/W |
| | | 2 | IRQ_LANE_FIFO | | Lane FIFO overflow/underflow. If EN_LANE_FIFO is low, IRQ_LANE_FIFO shows the current status. If EN_LANE_FIFO is high, IRQ_LANE_FIFO latches and pulls the $\overline{IRQx}$ pin low (x = MUX_LANE_FIFO setting). Writing a 1 to IRQ_LANE_FIFO when latched clears the bit. | 0x0 | R/W |
| | | 1 | IRQ_PRBSQ | | DAC1 PRBS error. If EN_PRBSQ is low, IRQ_PRBSQ shows the current status. If EN_PRBSQ is high, IRQ_PRBSQ latches and pulls the $\overline{IRQx}$ pin low (x = MUX_PRBSQ setting). Writing a 1 to IRQ_PRBSQ when latched clears the bit. | 0x0 | R/W |
| | | 0 | IRQ_PRBSI | | DAC0 PRBS error. If EN_PRBSI is low, IRQ_PRBSI shows the current status. If EN_PRBSI is high, IRQ_PRBSI latches and pulls the $\overline{IRQx}$ pin low (x = MUX_PRBSI setting). Writing a 1 to IRQ_PRBSI when latched clears the bit. | 0x0 | R/W |
| 0x025 | IRQ_STATUS0 | [7:4] | RESERVED | | Reserved. | 0x0 | R |

## REGISTER DETAILS

*Table 55. Register Details*

| Addr. | Name | Bits | Bit Name | Settings | Description | Reset | Access |
|---|---|---|---|---|---|---|---|
| | | 3 | IRQ_DAC0_CAL_DONE | | DAC0 calibration done. If EN_DAC0_CAL_DONE is low, IRQ_DAC0_CAL_DONE shows the current status. If EN_DAC0_CAL_DONE is high, IRQ_DAC0_CAL_DONE latches and pulls the $\overline{\text{IRQx}}$ pin low (x = MUX_DAC0_CAL_DONE setting). Writing a 1 to IRQ_DAC0_CAL_DONE when latched clears the bit. | 0x0 | R/W |
| | | [2:1] | RESERVED | | Reserved. | 0x0 | R/W |
| | | 0 | IRQ_PAERR0 | | DAC0 PA error. If EN_PAERR0 is low, IRQ_PAERR0 shows the current status. If EN_PAERR0 is high, IRQ_PAERR0 latches and pulls the $\overline{\text{IRQx}}$ pin low (x = MUX_PAERR0 setting). Writing a 1 to IRQ_PAERR0 when latched clears the bit. | 0x0 | R/W |
| 0x026 | IRQ_STATUS1 | [7:4] | RESERVED | | Reserved. | 0x0 | R |
| | | 3 | IRQ_DAC1_CAL_DONE | | DAC1 calibration done. If EN_DAC0_CAL_ DONE is low, IRQ_DAC1_CAL_DONE shows the current status. If EN_DAC1_CAL_DONE is high, IRQ_DAC1_CAL_DONE latches and pulls the $\overline{\text{IRQx}}$ pin low (x = MUX_DAC1_ CAL_DONE setting). Writing a 1 to IRQ_ DAC1_CAL_DONE when latched clears the bit. | 0x0 | R/W |
| | | [2:1] | RESERVED | | Reserved. | 0x0 | R/W |
| | | 0 | IRQ_PAERR1 | | DAC1 PA error. If EN_PAERR1 is low, IRQ_ PAERR1 shows the current status. If EN_ PAERR1 is high, IRQ_PAERR1 latches and pulls the $\overline{\text{IRQx}}$ pin low (x = MUX_PAERR1 setting). Writing a 1 to IRQ_PAERR1 when latched clears the bit. | 0x0 | R/W |
| 0x027 | IRQ_STATUS2 | [7:6] | RESERVED | | Reserved. | 0x0 | R |
| | | 5 | IRQ_DLL_LOST | | DLL lost. If EN_DLL_LOST is low, IRQ_DLL_ LOST shows the current status. If EN_DLL_ LOST is high, IRQ_DLL_LOST latches and pulls the $\overline{\text{IRQx}}$ pin low (x = MUX_DLL_LOST setting). Writing a 1 to IRQ_DLL_LOST when latched clears the bit. | 0x0 | R/W |
| | | 4 | IRQ_DLL_LOCK | | DLL locked. If EN_DLL_LOCK is low, IRQ_ DLL_LOCK shows current status. If EN_DLL_ LOCK is high, IRQ_DLL_LOCK latches and pulls the $\overline{\text{IRQx}}$ pin low (x = MUX_DLL_LOCK setting). Writing a 1 to IRQ_DLL_LOCK when latched clears the bit. | 0x0 | R/W |
| | | [3:2] | RESERVED | | Reserved. | 0x0 | R/W |
| | | 1 | IRQ_PLL_LOST | | DAC PLL lock lost. If EN_PLL_LOST is low, IRQ_PLL_LOST shows the current status. If EN_PLL_LOST is high, IRQ_PLL_LOST latches and pulls the $\overline{\text{IRQx}}$ pin low (x = MUX_PLL_ LOST setting). Writing a 1 to IRQ_PLL_ LOST when latched clears the bit. | 0x0 | R/W |
| | | 0 | IRQ_PLL_LOCK | | DAC PLL locked. If EN_PLL_LOCK is low, IRQ_PLL_LOCK shows the current status. If EN_PLL_LOCK is high, IRQ_PLL_LOCK latches and pulls the $\overline{\text{IRQx}}$ pin low (x = MUX_PLL_LOCK setting). Writing a 1 to IRQ_PLL_LOCK when latched clears the bit. | 0x0 | R/W |
| 0x028 | IRQ_OUTPUT_MUX | [7:5] | RESERVED | | Reserved. | 0x0 | R |

## REGISTER DETAILS

*Table 55. Register Details*

| Addr. | Name | Bits | Bit Name | Settings | Description | Reset | Access |
|---|---|---|---|---|---|---|---|
| | | 4 | MUX_SYSREF_JITTER | | If EN_SYSREF_JITTER is set, this control chooses the IRQx output pin on which the event is triggered. | 0x0 | R/W |
| | | | | 0 | Route IRQ trigger signal to the IRQ0 pin. | | |
| | | | | 1 | Route IRQ trigger signal to the IRQ1 pin. | | |
| | | 3 | MUX_DATA_READY | | If EN_DATA_READY is set, this control chooses the output IRQx pin on which the event is triggered. | 0x0 | R/W |
| | | | | 0 | Route IRQ trigger signal to the IRQ0 pin. | | |
| | | | | 1 | Route IRQ trigger signal to the IRQ1 pin. | | |
| | | 2 | MUX_LANE_FIFO | | If EN_LANE_FIFO is set, this control chooses the output IRQx pin on which the event is triggered. | 0x0 | R/W |
| | | | | 0 | Route IRQ trigger signal to the IRQ0 pin. | | |
| | | | | 1 | Route IRQ trigger signal to the IRQ1 pin. | | |
| | | 1 | MUX_PRBSQ | | If EN_PRBSQ is set, this control chooses the output IRQx pin on which the event is triggered. | 0x0 | R/W |
| | | | | 0 | Route IRQ trigger signal to the IRQ0 pin. | | |
| | | | | 1 | Route IRQ trigger signal to the IRQ1 pin. | | |
| | | 0 | MUX_PRBSI | | If EN_PRBSI is set, this control chooses the output IRQx pin on which the event is triggered. | 0x0 | R/W |
| | | | | 0 | Route IRQ trigger signal to the IRQ0 pin. | | |
| | | | | 1 | Route IRQ trigger signal to the IRQ1pin. | | |
| 0x029 | IRQ_OUTPUT_MUX0 | [7:4] | RESERVED | | Reserved. | 0x0 | R |
| | | 3 | MUX_DAC0_CAL_DONE | | If EN_DAC0_CAL_DONE is set, this control chooses the output IRQx pin on which the event is triggered. | 0x0 | R/W |
| | | | | 0 | Route IRQ trigger signal to the IRQ0 pin. | | |
| | | | | 1 | Route IRQ trigger signal to the IRQ1 pin. | | |
| | | [2:1] | RESERVED | | Reserved. | 0x0 | R/W |
| | | 0 | MUX_PAERR0 | | If EN_PAERR0 is set, this control chooses the output IRQx pin on which the event is triggered. | 0x0 | R/W |
| | | | | 0 | Route IRQ trigger signal to the IRQ0 pin. | | |
| | | | | 1 | Route IRQ trigger signal to the IRQ1 pin. | | |
| 0x02A | IRQ_OUTPUT_MUX1 | [7:4] | RESERVED | | Reserved. | 0x0 | R |
| | | 3 | MUX_DAC1_CAL_DONE | | If EN_DAC1_CAL_DONE is set, this control chooses the output IRQx pin on which the event is triggered. | 0x0 | R/W |
| | | | | 0 | Route IRQ trigger signal to the IRQ0 pin. | | |
| | | | | 1 | Route IRQ trigger signal to the IRQ1 pin. | | |
| | | [2:1] | RESERVED | | Reserved. | 0x0 | R/W |
| | | 0 | MUX_PAERR1 | | If EN_PAERR1 is set, this control chooses the output IRQx pin on which the event is triggered. | 0x0 | R/W |
| | | | | 0 | Route IRQ trigger signal to the IRQ0 pin. | | |
| | | | | 1 | Route IRQ trigger signal to the IRQ1 pin. | | |
| 0x02B | IRQ_OUTPUT_MUX2 | [7:6] | RESERVED | | Reserved. | 0x0 | R |
| | | 5 | MUX_DLL_LOST | | If EN_DLL_LOST is set, this control chooses the output IRQx pin on which the event is triggered. | 0x0 | R/W |
| | | | | 0 | Route IRQ trigger signal to the IRQ0 pin. | | |
| | | | | 1 | Route IRQ trigger signal to the IRQ1 pin. | | |
| | | 4 | MUX_DLL_LOCK | | If EN_DLL_LOCK is set, this control chooses the output IRQx pin on which the event is triggered. | 0x0 | R/W |
| | | | | 0 | Route IRQ trigger signal to the IRQ0 pin. | | |
| | | | | 1 | Route IRQ trigger signal to the IRQ1 pin. | | |
| | | [3:2] | RESERVED | | Reserved. | 0x0 | R/W |

## REGISTER DETAILS

*Table 55. Register Details*

| Addr. | Name | Bits | Bit Name | Settings | Description | Reset | Access |
|---|---|---|---|---|---|---|---|
| | | 1 | MUX_PLL_LOST | | If EN_PLL_LOST is set, this control chooses the $\overline{IRQx}$ pin on which the event is triggered. | 0x0 | R/W |
| | | | | 0 | Route IRQ trigger signal to the $\overline{IRQ0}$ pin. | | |
| | | | | 1 | Route IRQ trigger signal to the $\overline{IRQ1}$ pin. | | |
| | | 0 | MUX_PLL_LOCK | | If EN_PLL_LOCK is set, this control chooses the $\overline{IRQx}$ pin on which the event is triggered. | 0x0 | R/W |
| | | | | 0 | Route IRQ trigger signal to the $\overline{IRQ0}$ pin. | | |
| | | | | 1 | Route IRQ trigger signal to the $\overline{IRQ1}$ pin. | | |
| 0x02C | IRQ_STATUS_ALL | [7:1] | RESERVED | | Reserved. | 0x0 | R |
| | | 0 | IRQ_STATUS_ALL | | This bit is an OR of all the bits in Register 0x24 to Register 0x27. Writing a one to this bit clears any latched $\overline{IRQx}$ signals in Register 0x24 to Register 0x27. | 0x0 | R/W |
| 0x036 | SYSREF_COUNT | [7:0] | SYSREF_COUNT | | Number of rising SYSREF± edges to ignore before synchronization (pulse counting mode). | 0x0 | R/W |
| 0x039 | SYSREF_ERR_ WINDOW | 7 | RESERVED | | Reserved. | 0x0 | R |
| | | [6:0] | SYSREF_ERR_ WINDOW | | Amount of jitter allowed on the SYSREF± input. SYSREF± jitter variations larger than this trigger an interrupt. Units are in DAC clocks. | 0x0 | R/W |
| 0x03A | SYSREF_MODE | [7:5] | RESERVED | | Reserved. | 0x0 | R |
| | | 4 | SYNC_ROTATION_ DONE | | Synchronization logic rotation complete flag. | 0x1 | R |
| | | [3:2] | RESERVED | | Reserved. | 0x0 | R |
| | | 1 | SYSREF_MODE_ ONESHOT | | Enable one-shot synchronization rotation mode. | 0x0 | R/W |
| | | | | 00 | Monitor mode. Status/error flag for IRQ_SYSREF_JITTER is 1 if the SYSREF± edge is outside the error window (Register 0x039, Bits[6:0]). | | |
| | | | | 01 | Perform a single synchronization on the next SYSREF±, then switch to monitor mode. | | |
| | | 0 | RESERVED | | Reserved. | 0x0 | R/W |
| 0x03B | ROTATION_MODE | 7 | SYNCLOGIC_EN | | This bit must always be set to 1 (default) for both Subclass 0 and Subclass 1 operations. | 0x1 | R/W |
| | | 6 | RESERVED | | Reserved. For proper operation, write this bit to a 1. | 0x0 | R/W |
| | | 5 | PERIODIC_RST_EN | | Synchronization required setting. Always set this bit to 1 for both Subclass 0 and Subclass 1 operation. | 0x1 | R/W |
| | | 4 | NCORST_AFTER_ROT_ EN | | Set this bit to 1 to reset all NCOs after digital reset or synchronization rotation. Either this control or the START_NCO_SYNC bit (Register 0x1E7, Bit 0) can be used to reset all the NCOs (main and channel datapaths). | 0x1 | R/W |
| | | [3:2] | RESERVED | | Reserved. | 0x0 | R |
| | | [1:0] | ROTATION_MODE | | Selects the circuitry to be reset when a synchronization rotation occurs. Certain bits being set to 1 determine the actions taken when a synchronization rotation is performed. Bit 0 corresponds to a SERDES clock reset and realignment. Bit 1 corresponds to a datapath soft off/on gain, which must only be used if PA protection is in use. If PA protection is not used, set Bit 1 to 0. | 0x0 | R/W |
| | | | | 0 | No action, with either the SERDES clocks or the datapath, occurs when a synchronization rotation occurs. | | |

## REGISTER DETAILS

*Table 55. Register Details*

| Addr. | Name | Bits | Bit Name | Settings | Description | Reset | Access |
|---|---|---|---|---|---|---|---|
| | | | | 1 | The links drop and the SERDES clocks are reset. It is recommended to set this bit high so that when a synchronization rotation is performed, the SERDES clocks realign properly. | | |
| | | | | 10 | The datapath automatically uses the soft on/off functionality to turn on and off the datapath stream during a synchronization rotation to avoid corrupted data from being transmitted. Only use this feature if the PA protection block is in use. | | |
| | | | | 11 | Both the SERDES clock reset and datapath soft on/off feature are enabled. | | |
| 0x03F | TX_ENABLE | [7:6] | RESERVED | | Reserved. | 0x0 | R/W |
| | | 5 | TXEN_DATAPATH_DAC1 | | Selects whether the datapath of DAC1 is muted when the TXEN1 pin is brought low. | 0x0 | R/W |
| | | | | 0 | Datapath output is normal. | | |
| | | | | 1 | If TXEN1 = 0, the datapath output is immediately zeroed. If TXEN1 = 1, the datapath outputs normal operation. | | |
| | | 4 | TXEN_DATAPATH_DAC0 | | Selects whether the datapath of DAC0 is muted when the TXEN0 pin is brought low. | 0x0 | R/W |
| | | | | 0 | Datapath output is normal. | | |
| | | | | 1 | If TXEN0 = 0, the datapath output is immediately zeroed. If TXEN0 = 1, the datapath outputs normal operation. | | |
| | | [3:0] | RESERVED | | Reserved. | 0x0 | R/W |
| 0x050 | CAL_CLK_DIV | [7:4] | RESERVED | | Reserved | 0x2 | R/W |
| | | [3:0] | CAL_CLK_DIV | | Calibration register control. Set these bits to 0xA for optimized calibration setting. | 0x8 | R/W |
| 0x051 | CAL_CTRL | 7 | CAL_CTRL0 | | Calibration setting. Set this bit to 1. | 0x1 | R/W |
| | | | | 0 | Reset the calibration engine. | | |
| | | | | 1 | Enable the calibration routine. | | |
| | | [6:3] | RESERVED | | Reserved. | 0x0 | R/W |
| | | [2:1] | CAL_CTRL1 | | Calibration mode selection. Set this bit field to 1 for optimized calibration mode. Paged by the MAINDAC_PAGE bits in Register 0x008. | 0x1 | R/W |
| | | | | 1 | Set calibration control setting. | | |
| | | 0 | CAL_START | | Start calibration. After starting calibration, do not write to any register from Register 0x051 to Register 0x061 until Register 0x052, Bit 2 reads low (indicating that the calibration is no longer active). Paged by the MAINDAC_PAGE bits in Register 0x008. | 0x0 | R/W |
| 0x052 | CAL_STAT | [7:3] | RESERVED | | Reserved. | 0x0 | R/W |
| | | 2 | CAL_ACTIVE | | Calibration active status flag. A readback of 1 indicates the calibration routine is still in progress. This control is paged by the MAINDAC_PAGE bits in Register 0x008. | 0x0 | R |
| | | 1 | CAL_FAIL_SEARCH | | Calibration failure flag. A readback of 1 indicates the calibration routine failed and is possibly not valid. This control is paged by the MAINDAC_PAGE bits in Register 0x008. | 0x0 | R |
| | | 0 | CAL_FINISH | | Calibration complete flag. A readback of 1 indicates the calibration completed. This control is paged by the MAINDAC_PAGE bits in Register 0x008. | 0x0 | R |

## REGISTER DETAILS

*Table 55. Register Details*

| Addr. | Name | Bits | Bit Name | Settings | Description | Reset | Access |
|---|---|---|---|---|---|---|---|
| 0x059 | FSC0 | [7:2] | RESERVED | | Reserved. | 0xA | R/W |
| | | [1:0] | FSC_CTRL[1:0] | | Sets the full-scale current control. This control is paged by the MAINDAC_PAGE bits in Register 0x008. Full-scale current = 16 mA + FSC_CTRL × (25/1024) (mA). | 0x0 | R/W |
| 0x05A | FSC1 | [7:0] | FSC_CTRL[9:2] | | Sets the full-scale current control. This control is paged by the MAINDAC_PAGE bits in Register 0x008. Full-scale current = 16 mA + FSC_CTRL × (25/1024) (mA). | 0x28 | R/W |
| 0x061 | CAL_DEBUG0 | 7 | RESERVED | | Reserved. | 0x0 | R/W |
| | | 6 | CAL_CTRL2 | | Calibration control. Set this bit to 1 for optimized calibration setting. | 0x1 | R/W |
| | | 5 | CAL_CTRL3 | | Calibration control. Set this bit to 1 for optimized calibration setting. | 0x1 | R/W |
| | | 4 | RESERVED | | Reserved. | 0x0 | R/W |
| | | 3 | CAL_CTRL4 | | Calibration control. Set this bit to 1 for optimized calibration setting. | 0x0 | R/W |
| | | [2:0] | RESERVED | | Reserved. | 0x0 | R/W |
| 0x081 | CLK_CTRL | [7:2] | RESERVED | | Reserved. | 0x0 | R/W |
| | | 1 | CAL_CLK_PD1 | | After the calibration is complete for DAC1 (Register 0x052, Bit 0 = 1), set this bit high to power down the calibration clock. | 0x0 | R/W |
| | | 0 | CAL_CLK_PD0 | | After the calibration is complete for DAC0 (Register 0x052, Bit 0 = 1), set this bit high to power down the calibration clock. | 0x0 | R/W |
| 0x083 | NVM_CTRL0 | 7 | NVM_CTRL0A | | NVM register control for the ring oscillator. | 0x0 | R/W |
| | | [6:2] | RESERVED | | Reserved. | 0x0 | R |
| | | [1:0] | NVM_CTRL0B | | NVM register control for the ring oscillator. | 0x2 | R/W |
| | | | | 00 | Divide by 8. | | |
| | | | | 01 | Divide by 16. | | |
| | | | | 10 | Divide by 32. | | |
| | | | | 11 | Divide by 64. | | |
| 0x084 | SYSREF_CTRL | 7 | RESERVED | | Reserved. | 0x0 | R/W |
| | | 6 | SYSREF_INPUTMODE | | Sets the input mode type for the SYSREF± pins. | 0x0 | R/W |
| | | | | 0 | AC couple SYSREF±. | | |
| | | | | 1 | DC couple SYSREF±. | | |
| | | [5:1] | RESERVED | | Reserved. | 0x0 | R/W |
| | | 0 | SYSREF_PD | | Power down the SYSREF± receiver and synchronization circuitry. If using Subclass 0, set this bit to 1 because the SYSREF± pins are not used. | 0x0 | R/W |
| | | | | 0 | SYSREF± receiver is powered on. | | |
| | | | | 1 | SYSREF± receiver is powered down. | | |
| 0x085 | NVM_CTRL1 | 7 | RESERVED | | Reserved. | 0x0 | R |
| | | [6:4] | NVM_CTRL1A | | NVM control. | 0x1 | R/W |
| | | [3:2] | RESERVED | | Reserved. | 0x0 | R |
| | | 1 | NVM_CTRL1B | | NVM control. | 0x1 | R/W |
| | | 0 | NVM_CTRL1C | | NVM control. | 0x1 | R/W |
| 0x08D | ADC_CLK_CTRL0 | [7:5] | RESERVED | | Reserved. | 0x0 | R |
| | | [4:0] | CLKOUT_SWING | | Controls the swing level of the ADC clock driver. Note that swing can be negative (inverts clock). | 0x0 | R/W |

## REGISTER DETAILS

*Table 55. Register Details*

| Addr. | Name | Bits | Bit Name | Settings | Description | Reset | Access |
|---|---|---|---|---|---|---|---|
| | | | | | The calculation for Code 0 to Code 9 is as follows:<br>ADC driver swing = 993 mV − CLKOUT_SWING × 77 mV.<br>The calculation for Code 10 to Code 19 is as follows:<br>ADC driver swing = (20 − CLKOUT_SWING × 77 mV) − 1 V. | | |
| 0x08F | ADC_CLK_CTRL2 | [7:1] | RESERVED | | Reserved. | 0x0 | R |
| | | 0 | PD_CLKOUT_DRIVER | | Powers down the CLKOUT± output driver. | 0x0 | R/W |
| 0x090 | DAC_POWERDOWN | [7:2] | RESERVED | | Reserved. | 0x0 | R |
| | | 1 | DAC_PD1 | | Powers down DAC1. | 0x1 | R/W |
| | | | | 0 | Power up DAC1. | | |
| | | | | 1 | Power down DAC1. | | |
| | | 0 | DAC_PD0 | | Powers down DAC0. | 0x1 | R/W |
| | | | | 0 | Power up DAC0. | | |
| | | | | 1 | Power down DAC0. | | |
| 0x091 | ACLK_CTRL | [7:1] | RESERVED | | Reserved. | 0x0 | R/W |
| | | 0 | ACLK_POWERDOWN | | Analog clock receiver power-down. | 0x1 | R/W |
| 0x094 | PLL_CLK_DIV | [7:2] | RESERVED | | Reserved. | 0x0 | R |
| | | 1 | PLL_VCO_DIV3_EN | | Enable PLL output clock divide by 3. | 0x0 | R/W |
| | | 0 | PLL_VCO_DIV2_EN | | Enable PLL output clock divide by 2. | 0x0 | R/W |
| | | | | 0 | DAC clock = PLL VCO clock frequency. | | |
| | | | | 1 | DAC clock = PLL VCO clock frequency ÷ 2. | | |
| 0x095 | PLL_BYPASS | [7:1] | RESERVED | | Reserved. | 0x0 | R |
| | | 0 | PLL_BYPASS | | Enable direct clocking (bypassing the PLL clock). | 0x0 | R/W |
| | | | | 0 | Use the internal PLL to generate the DAC clock. | | |
| | | | | 1 | Bypass the PLL and directly clock with the DAC clock frequency. | | |
| 0x09A | NVM_CTRL | 7 | PD_BGR | | Bias power-down. Set this bit to 1 to power down the internal bias. | 0x0 | R/W |
| | | [6:0] | RESERVED | | Reserved. | 0x0 | R/W |
| 0x0C0 | DELAY_LINE_PD | [7:6] | RESERVED | | Reserved. | 0x0 | R |
| | | 5 | DLL_CTRL0B | | DLL control. Set this bit to 0 to power up the delay line during the device configuration sequence. | 0x1 | R/W |
| | | 4 | DLL_CTRL0A | | DLL control. Set this bit to 0 to power up the delay line during the device configuration sequence. | 0x1 | R/W |
| | | [3:1] | RESERVED | | Reserved. | 0x0 | R |
| | | 0 | DLL_PD | | Power down delay line. Set this bit to 0 to power up the delay line during the device configuration sequence. | 0x1 | R/W |
| | | | | 0 | Power up and enable the delay line. | | |
| | | | | 1 | Power down and bypass the delay line. | | |
| 0x0C1 | DLL_CTRL0 | [7:6] | DLL_CTRL1C | | DAC control setting. Set this control to 1 for optimal performance. | 0x1 | R/W |
| | | 5 | DLL_CTRL1B | | DLL control search mode. If the DAC frequency is <4.5 GHz, set this bit to 0; otherwise, set this bit to 1. | 0x1 | R/W |
| | | [4:3] | DLL_CTRL1A | | DLL control search direction. Set this control to 1 for optimal performance. | 0x2 | R/W |
| | | [2:1] | RESERVED | | Reserved. | 0x0 | R |
| | | 0 | DLL_ENABLE | | DLL controller enable. | 0x0 | R/W |

## REGISTER DETAILS

*Table 55. Register Details*

| Addr. | Name | Bits | Bit Name | Settings | Description | Reset | Access |
|---|---|---|---|---|---|---|---|
| | | | | 0 | Disable DLL. | | |
| | | | | 1 | Enable DLL. | | |
| 0x0C3 | DLL_STATUS | [7:1] | RESERVED | | Reserved. | 0x0 | R |
| | | 0 | DLL_LOCK | | DLL lock indicator. This control reads back 1 if the DLL locks. | 0x0 | R |
| 0x0C7 | DLL_READ | [7:1] | RESERVED | | Reserved. | 0x0 | R |
| | | 0 | DLL_READ_EN | | Enable DLL readback status. A transition of 0 to 1 updates the lock status bit readback in Register 0x0C3. | 0x0 | R/W |
| 0x0CC | DLL_FINE_DELAY0 | [7:6] | RESERVED | | Reserved. | 0x0 | R |
| | | [5:0] | DLL_FINE_DELAY0 | | DLL delay control. | 0x0 | R/W |
| 0x0CD | DLL_FINE_DELAY1 | [7:6] | RESERVED | | Reserved. | 0x0 | R |
| | | [5:0] | DLL_FINE_DELAY1 | | DLL delay control. | 0x0 | R/W |
| 0x0DB | DLL_UPDATE | [7:1] | RESERVED | | Reserved. | 0x0 | R |
| | | 0 | DLL_DELAY_UPDATE | | DLL update control. A transition from 0 to 1 updates the DLL circuitry with the current register control settings. | 0x0 | R/W |
| 0x100 | DIG_RESET | [7:1] | RESERVED | | Reserved. | 0x0 | R |
| | | 0 | DIG_DATAPATH_PD | | Holds all digital logic (SERDES digital, digital clock generation, and digital datapath) in reset until clock tree is stable. | 0x1 | R/W |
| | | | | 0 | Normal operating mode. | | |
| | | | | 1 | Holds the digital logic in reset. Must be released (set to 0) after clocks to the chip are stable (PLL and DLL blocks are locked) to use the digital datapath. | | |
| 0x110 | JESD_MODE | 7 | MODE_NOT_IN_TABLE | | Programmed JESD204B mode and interpolation mode combination is not valid. Select a different combination. | 0x0 | R |
| | | 6 | COM_SYNC | | Combine the $\overline{\text{SYNCOUTx}}\pm$ signals in dual link case. | 0x0 | R/W |
| | | [5:0] | JESD_MODE | | Sets the JESD204B mode configuration. See Table 12 for the JESD204B supported operating modes and compatible interpolation rates. Bit 5 of this control determines single link (set to 0) or dual link (set to 1). Bits[4:0] set the desired JESD204B mode according to Table 12. | 0x20 | R/W |
| 0x111 | INTRP_MODE | [7:4] | DP_INTERP_MODE | | Sets main datapath interpolation rate. See Table 12 for the JESD204B supported operating modes and compatible JESD204B modes and channel interpolation rates. | 0x8 | R/W |
| | | | | 0x1 | 1×. | | |
| | | | | 0x2 | 2×. | | |
| | | | | 0x4 | 4×. | | |
| | | | | 0x6 | 6×. | | |
| | | | | 0x8 | 8×. | | |
| | | | | 0xC | 12×. | | |
| | | [3:0] | CH_INTERP_MODE | | Sets channel interpolation rate. See Table 12 for the JESD204B supported operating modes and compatible JESD204B modes and main datapath interpolation rates. | 0x4 | R/W |
| | | | | 0x1 | 1×. | | |
| | | | | 0x2 | 2×. | | |

## REGISTER DETAILS

*Table 55. Register Details*

| Addr. | Name | Bits | Bit Name | Settings | Description | Reset | Access |
|---|---|---|---|---|---|---|---|
| | | | | 0x3 | 3×. | | |
| | | | | 0x4 | 4×. | | |
| | | | | 0x6 | 6×. | | |
| | | | | 0x8 | 8×. | | |
| 0x112 | DDSM_DATAPATH_CFG | [7:6] | RESERVED | | Reserved. | 0x0 | R |
| | | [5:4] | DDSM_MODE | | Modulator switch mode selection. This control chooses the mode of operation for the main datapath NCO being configured. This control is paged by the MAINDAC_PAGE bits in Register 0x008. | 0x0 | R/W |
| | | | | 00 | DAC0 = I0; DAC1 = I1. | | |
| | | | | 01 | DAC0 = I0 + I1; DAC1 = Q0 + Q1. | | |
| | | | | 10 | DAC0 = I0; DAC1 = Q0. | | |
| | | | | 11 | DAC0 = I0 + I1; DAC1 = 0. | | |
| | | 3 | DDSM_NCO_EN | | Main datapath modulation enable. If the JESD204B mode chosen is a complex mode (main datapath interpolation >1×), this bit must be set to 1 for each main datapath being used. If no modulation is desired, set the FTW to be 0. This control is paged by the MAINDAC_PAGE bits in Register 0x008. | 0x0 | R/W |
| | | | | 0 | Disable main datapath NCO. | | |
| | | | | 1 | Enable main datapath NCO. | | |
| | | 2 | DDSM_MODULUS_EN | | Enable main datapath modulus DDS. This control is paged by the MAINDAC_PAGE bits in Register 0x008. | 0x0 | R/W |
| | | | | 0 | Disable modulus DDS. | | |
| | | | | 1 | Enable modulus DDS. | | |
| | | 1 | DDSM_SEL_SIDEBAND | | Selects upper or lower sideband from modulation result. This control is paged by the MAINDAC_PAGE bits in Register 0x008. | 0x0 | R/W |
| | | | | 0 | Use upper sideband. | | |
| | | | | 1 | Use lower sideband = spectral flip. | | |
| | | 0 | EN_SYNC_ALL_CHNL_NCO_RESETS | | Selects the signal channel NCOS used for resets and FTW updates. This control is paged by the MAINDAC_PAGE bits in Register 0x008. | 0x1 | R/W |
| | | | | 0 | Channel NCOs reset or update their FTW based on channel NCO update requests. | | |
| | | | | 1 | Channel NCOs reset or update their FTW based on main datapath NCO update requests. | | |
| 0x113 | DDSM_FTW_UPDATE | 7 | RESERVED | | Reserved. | 0x0 | R |
| | | [6:4] | DDSM_FTW_REQ_MODE | | Frequency tuning word automatic update mode. This control is paged by the MAINDAC_PAGE bits in Register 0x008. | 0x0 | R/W |
| | | | | 000 | No automatic requests are generated when the FTW registers are written. | | |
| | | | | 001 | Automatically generates a DDSM_FTW_LOAD_REQ after DDSM_FTW Bits[7:0] are written. | | |
| | | | | 010 | Automatically generates a DDSM_FTW_LOAD_REQ after DDSM_FTW Bits[15:8] are written. | | |
| | | | | 011 | Automatically generates a DDSM_FTW_LOAD_REQ after DDSM_FTW Bits[23:16] are written. | | |
| | | | | 100 | Automatically generates a DDSM_FTW_LOAD_REQ after DDSM_FTW Bits[31:24] are written. | | |

## REGISTER DETAILS

*Table 55. Register Details*

| Addr. | Name | Bits | Bit Name | Settings | Description | Reset | Access |
|---|---|---|---|---|---|---|---|
| | | | | 101 | Automatically generate a DDSM_FTW_LOAD_REQ after DDSM_FTW Bits[39:32] is written. | | |
| | | | | 110 | Automatically generates a DDSM_FTW_LOAD_REQ after DDSM_FTW Bits[47:40] are written. | | |
| | | 3 | RESERVED | | Reserved. | 0x0 | R |
| | | 2 | DDSM_FTW_LOAD_SYSREF | | Uses the next rising edge of SYSREF± to trigger FTW load and reset. This bit also loads the calibration tone FTW, as well as the main NCO FTW on a rising edge detection. This control is paged by the MAINDAC_PAGE bits in Register 0x008. | 0x0 | R/W |
| | | 1 | DDSM_FTW_LOAD_ACK | | Frequency tuning word update acknow-ledge. This bit reads back 1 if the FTW and phase offset word is loaded properly. This control is paged by the MAINDAC_PAGE bits in Register 0x008. | 0x0 | R |
| | | | | 0 | FTW is not loaded. | | |
| | | | | 1 | FTW is loaded. | | |
| | | 0 | DDSM_FTW_LOAD_REQ | | Frequency tuning word update request from the SPI. This bit also loads the calibration tone FTW, as well as the main NCO FTW on a rising edge detection. This control is paged by the MAINDAC_PAGE bits in Register 0x008. | 0x0 | R/W |
| | | | | 0 | Clear DDSM_FTW_LOAD_ACK. | | |
| | | | | 1 | 0 to 1 transition loads the FTW. | | |
| 0x114 | DDSM_FTW0 | [7:0] | DDSM_FTW[7:0] | | Sets the main datapath NCO FTW. If DDSM_MODULUS_EN is low, the main datapath NCO frequency = $f_{DAC}$ × (DDSM_FTW/$2^{48}$). If DDSM_MODULUS_EN is high, the main datapath NCO frequency = $f_{DAC}$ × (DDSM_FTW + DDSM_ACC_DELTA/DDSM_ACC_MODULUS)/$2^{48}$. DDSM_ACC_DELTA must be >0. DDSM_ACC_DELTA must be >DDSM_ACC_MODULUS. This control is paged by the MAINDAC_PAGE bits in Register 0x008. | 0x0 | R/W |
| 0x115 | DDSM_FTW1 | [7:0] | DDSM_FTW[15:8] | | Sets the main datapath NCO FTW. If DDSM_MODULUS_EN is low, the main datapath NCO frequency = $f_{DAC}$ × (DDSM_FTW/$2^{48}$). If DDSM_MODULUS_EN is high, the main datapath NCO frequency = $f_{DAC}$ × (DDSM_FTW + DDSM_ACC_DELTA/DDSM_ACC_MODULUS)/$2^{48}$. DDSM_ACC_DELTA must be > 0. DDSM_ACC_DELTA must be > DDSM_ACC_MODULUS. This control is paged by the MAINDAC_PAGE bits in Register 0x008. | 0x0 | R/W |
| 0x116 | DDSM_FTW2 | [7:0] | DDSM_FTW[23:16] | | Sets the main datapath NCO FTW. If DDSM_MODULUS_EN is low, the main datapath NCO frequency = $f_{DAC}$ × (DDSM_FTW/$2^{48}$). If DDSM_MODULUS_EN is high, the main datapath NCO frequency = $f_{DAC}$ × (DDSM_FTW + DDSM_ACC_DELTA/DDSM_ACC_MODULUS)/$2^{48}$. DDSM_ACC_DELTA must be > 0. DDSM_ACC_DELTA must be > DDSM_ACC_MODULUS. This control is paged by the MAINDAC_PAGE bits in Register 0x008. | 0x0 | R/W |

## REGISTER DETAILS

*Table 55. Register Details*

| Addr. | Name | Bits | Bit Name | Settings | Description | Reset | Access |
|---|---|---|---|---|---|---|---|
| 0x117 | DDSM_FTW3 | [7:0] | DDSM_FTW[31:24] | | Sets the main datapath NCO FTW. If DDSM_MODULUS_EN is low, the main datapath NCO frequency = $f_{DAC}$ × (DDSM_FTW/$2^{48}$). If DDSM_MODULUS_EN is high, the main datapath NCO frequency = $f_{DAC}$ × (DDSM_FTW + DDSM_ACC_DELTA/DDSM_ACC_MODULUS)/$2^{48}$. DDSM_ACC_DELTA must be > 0. DDSM_ACC_DELTA must be > DDSM_ACC_MODULUS. This control is paged by the MAINDAC_PAGE bits in Register 0x008. | 0x0 | R/W |
| 0x118 | DDSM_FTW4 | [7:0] | DDSM_FTW[39:32] | | Sets the main datapath NCO FTW. If DDSM_MODULUS_EN is low, the main datapath NCO frequency = $f_{DAC}$ × (DDSM_FTW/$2^{48}$). If DDSM_MODULUS_EN is high, the main datapath NCO frequency = $f_{DAC}$ × (DDSM_FTW + DDSM_ACC_DELTA/DDSM_ACC_MODULUS)/$2^{48}$. DDSM_ACC_DELTA must be > 0. DDSM_ACC_DELTA must be > DDSM_ACC_MODULUS. This control is paged by the MAINDAC_PAGE bits in Register 0x008. | 0x0 | R/W |
| 0x119 | DDSM_FTW5 | [7:0] | DDSM_FTW[47:40] | | Sets the main datapath NCO FTW. If DDSM_MODULUS_EN is low, the main datapath NCO frequency = $f_{DAC}$ × (DDSM_FTW/$2^{48}$). If DDSM_MODULUS_EN is high, the main datapath NCO frequency = $f_{DAC}$ × (DDSM_FTW + DDSM_ACC_DELTA/DDSM_ACC_MODULUS)/$2^{48}$. DDSM_ACC_DELTA must be > 0. DDSM_ACC_DELTA must be > DDSM_ACC_MODULUS. This control is paged by the MAINDAC_PAGE bits in Register 0x008. | 0x0 | R/W |
| 0x11C | DDSM_PHASE_OFFSET0 | [7:0] | DDSM_NCO_PHASE_OFFSET[7:0] | | Sets main datapath NCO phase offset. Code is in 16-bit, twos complement format. Degrees offset = 180 × (code/$2^{15}$). This control is paged by the MAINDAC_PAGE bits in Register 0x008. | 0x0 | R/W |
| 0x11D | DDSM_PHASE_OFFSET1 | [7:0] | DDSM_NCO_PHASE_OFFSET[15:8] | | Sets main datapath NCO phase offset. Code is in 16-bit, twos complement format. Degrees offset = 180 × (code/$2^{15}$). This control is paged by the MAINDAC_PAGE bits in Register 0x008. | 0x0 | R/W |
| 0x124 | DDSM_ACC_MODULUS0 | [7:0] | DDSM_ACC_MODULUS[7:0] | | Sets DDSM_ACC_MODULUS. If DDSM_MODULUS_EN is high, the main datapath NCO frequency = $f_{DAC}$ × (DDSM_FTW + DDSM_ACC_DELTA/DDSM_ACC_MODULUS)/$2^{48}$. DDSM_ACC_DELTA must be > 0. DDSM_ACC_DELTA must be > DDSM_ACC_MODULUS. This control is paged by the MAINDAC_PAGE bits in Register 0x008. | 0x0 | R/W |
| 0x125 | DDSM_ACC_MODULUS1 | [7:0] | DDSM_ACC_MODULUS[15:8] | | Sets DDSM_ACC_MODULUS. If DDSM_MODULUS_EN is high, the main datapath NCO frequency = $f_{DAC}$ × (DDSM_FTW + DDSM_ACC_DELTA/DDSM_ACC_MODULUS)/$2^{48}$. DDSM_ACC_DELTA must be > 0. DDSM_ACC_DELTA must be > DDSM_ACC_MODULUS. This control is paged by the MAINDAC_PAGE bits in Register 0x008. | 0x0 | R/W |

## REGISTER DETAILS

*Table 55. Register Details*

| Addr. | Name | Bits | Bit Name | Settings | Description | Reset | Access |
|---|---|---|---|---|---|---|---|
| 0x126 | DDSM_ACC_ MODULUS2 | [7:0] | DDSM_ACC_ MODULUS[23:16] | | Sets DDSM_ACC_MODULUS. If DDSM_ MODULUS_EN is high, the main datapath NCO frequency = $f_{DAC}$ × (DDSM_FTW + DDSM_ACC_DELTA/DDSM_ACC_MODULUS)/$2^{48}$. DDSM_ACC_DELTA must be > 0. DDSM_ ACC_DELTA must be > DDSM_ACC_ MODULUS. This control is paged by the MAINDAC_PAGE bits in Register 0x008. | 0x0 | R/W |
| 0x127 | DDSM_ACC_ MODULUS3 | [7:0] | DDSM_ACC_ MODULUS[31:24] | | Sets DDSM_ACC_MODULUS. If DDSM_ MODULUS_EN is high, the main datapath NCO frequency = $f_{DAC}$ × (DDSM_FTW + DDSM_ACC_DELTA/DDSM_ACC_MODULUS)/$2^{48}$. DDSM_ACC_DELTA must be > 0. DDSM_ ACC_DELTA must be > DDSM_ACC_ MODULUS. This control is paged by the MAINDAC_PAGE bits in Register 0x008. | 0x0 | R/W |
| 0x128 | DDSM_ACC_ MODULUS4 | [7:0] | DDSM_ACC_ MODULUS[39:32] | | Sets DDSM_ACC_MODULUS. If DDSM_ MODULUS_EN is high, the main datapath NCO frequency = $f_{DAC}$ × (DDSM_FTW + DDSM_ACC_DELTA/DDSM_ACC_MODULUS)/$2^{48}$. DDSM_ACC_DELTA must be > 0. DDSM_ ACC_DELTA must be > DDSM_ACC_ MODULUS. This control is paged by the MAINDAC_PAGE bits in Register 0x008. | 0x0 | R/W |
| 0x129 | DDSM_ACC_ MODULUS5 | [7:0] | DDSM_ACC_ MODULUS[47:40] | | Sets DDSM_ACC_MODULUS. If DDSM_ MODULUS_EN is high, the main datapath NCO frequency = $f_{DAC}$ × (DDSM_FTW + DDSM_ACC_DELTA/DDSM_ACC_MODULUS)/$2^{48}$. DDSM_ACC_DELTA must be > 0. DDSM_ ACC_DELTA must be > DDSM_ACC_ MODULUS. This control is paged by the MAINDAC_PAGE bits in Register 0x008. | 0x0 | R/W |
| 0x12A | DDSM_ACC_DELTA0 | [7:0] | DDSM_ACC_ DELTA[7:0] | | Sets DDSM_ACC_DELTA. If DDSM_ MODULUS_EN is high, main datapath NCO frequency = $f_{DAC}$ × (DDSM_FTW + DDSM_ ACC_DELTA/DDSM_ACC_MODULUS)/$2^{48}$. DDSM_ACC_DELTA must be > 0. DDSM_ ACC_DELTA must be > DDSM_ACC_ MODULUS. This control is paged by the MAINDAC_PAGE bits in Register 0x008. | 0x0 | R/W |
| 0x12B | DDSM_ACC_DELTA1 | [7:0] | DDSM_ACC_ DELTA[15:8] | | Sets DDSM_ACC_DELTA. If DDSM_ MODULUS_EN is high, main datapath NCO frequency = $f_{DAC}$ × (DDSM_FTW + DDSM_ ACC_DELTA/DDSM_ACC_MODULUS)/$2^{48}$. DDSM_ACC_DELTA must be > 0. DDSM_ ACC_DELTA must be > DDSM_ACC_ MODULUS. This control is paged by the MAINDAC_PAGE bits in Register 0x008. | 0x0 | R/W |
| 0x12C | DDSM_ACC_DELTA2 | [7:0] | DDSM_ACC_ DELTA[23:16] | | Sets DDSM_ACC_DELTA. If DDSM_ MODULUS_EN is high, main datapath NCO frequency = $f_{DAC}$ × (DDSM_FTW + DDSM_ | 0x0 | R/W |

## REGISTER DETAILS

*Table 55. Register Details*

| Addr. | Name | Bits | Bit Name | Settings | Description | Reset | Access |
|---|---|---|---|---|---|---|---|
| | | | | | ACC_DELTA/DDSM_ACC_MODULUS)/$2^{48}$. DDSM_ACC_DELTA must be > 0. DDSM_ ACC_DELTA must be > DDSM_ACC_ MODULUS. This control is paged by the MAINDAC_PAGE bits in Register 0x008. | | |
| 0x12D | DDSM_ACC_DELTA3 | [7:0] | DDSM_ACC_ DELTA[31:24] | | Sets DDSM_ACC_DELTA. If DDSM_ MODULUS_EN is high, main datapath NCO frequency = $f_{DAC}$ × (DDSM_FTW + DDSM_ ACC_DELTA/DDSM_ACC_MODULUS)/$2^{48}$. DDSM_ACC_DELTA must be > 0. DDSM_ ACC_DELTA must be > DDSM_ACC_ MODULUS. This control is paged by the MAINDAC_PAGE bits in Register 0x008. | 0x0 | R/W |
| 0x12E | DDSM_ACC_DELTA4 | [7:0] | DDSM_ACC_ DELTA[39:32] | | Sets DDSM_ACC_DELTA. If DDSM_ MODULUS_EN is high, main datapath NCO frequency = $f_{DAC}$ × (DDSM_FTW + DDSM_ ACC_DELTA/DDSM_ACC_MODULUS)/$2^{48}$. DDSM_ACC_DELTA must be > 0. DDSM_ ACC_DELTA must be > DDSM_ACC_ MODULUS. This control is paged by the MAINDAC_PAGE bits in Register 0x008. | 0x0 | R/W |
| 0x12F | DDSM_ACC_DELTA5 | [7:0] | DDSM_ACC_ DELTA[47:40] | | Sets DDSM_ACC_DELTA. If DDSM_ MODULUS_EN is high, main datapath NCO frequency = $f_{DAC}$ × (DDSM_FTW + DDSM_ ACC_DELTA/DDSM_ACC_MODULUS)/$2^{48}$. DDSM_ACC_DELTA must be > 0. DDSM_ ACC_DELTA must be > DDSM_ACC_ MODULUS. This control is paged by the MAINDAC_PAGE bits in Register 0x008. | 0x0 | R/W |
| 0x130 | DDSC_DATAPATH_ CFG | 7 | RESERVED | | Reserved. | 0x0 | R |
| | | 6 | DDSC_NCO_EN | | Channel datapath modulation enable. If the JESD204B mode chosen is a complex mode (channel interpolation >1×), this bit must be set to 1 for each channel datapath being used. If no modulation is desired, set the FTW to 0. This control is paged by the CHANNEL_PAGE bits in Register 0x008. | 0x0 | R/W |
| | | | | 0 | Disable channel NCO. | | |
| | | | | 1 | Enable channel NCO. | | |
| | | [5:3] | RESERVED | | Reserved. | 0x0 | R/W |
| | | 2 | DDSC_MODULUS_EN | | Enable channel modulus DDS. This control is paged by the CHANNEL_PAGE bits in Register 0x008. | 0x0 | R/W |
| | | | | 0 | Disable modulus DDS. | | |
| | | | | 1 | Enable modulus DDS. | | |
| | | 1 | DDSC_SEL_SIDEBAND | | Selects upper or lower sideband from modulation result. This control is paged by the CHANNEL_PAGE bits in Register 0x008. | 0x0 | R/W |
| | | | | 0 | Use upper sideband. | | |
| | | | | 1 | Use lower sideband = spectral flip. | | |
| | | 0 | DDSC_EN_DC_INPUT | | Enable test tone generation by sending dc to input level to channel DDS. Set the amplitude in the DC_TEST_INPUT_ | 0x0 | R/W |

## REGISTER DETAILS

*Table 55. Register Details*

| Addr. | Name | Bits | Bit Name | Settings | Description | Reset | Access |
|---|---|---|---|---|---|---|---|
| | | | | | AMPLITUDE control (Register 0x148 and Register 0x149). This control is paged by the CHANNEL_PAGE bits in Register 0x008. | | |
| | | | | 0 | Disable test tone generation. | | |
| | | | | 1 | Enable test tone generation. | | |
| 0x131 | DDSC_FTW_UPDATE | [7:3] | RESERVED | | Reserved. | 0x0 | R |
| | | 2 | DDSC_FTW_LOAD_SYSREF | | Use next rising edge of SYSREF± to trigger FTW load and reset. This control is paged by the CHANNEL_PAGE bits in Register 0x008. | 0x0 | R/W |
| | | 1 | DDSC_FTW_LOAD_ACK | | Frequency tuning word update acknowledge bit. This bit reads back 1 if the FTW and phase offset word is loaded properly. This control is paged by the CHANNEL_PAGE bits in Register 0x008. | 0x0 | R |
| | | | | 0 | FTW is not loaded. | | |
| | | | | 1 | FTW is loaded. | | |
| | | 0 | DDSC_FTW_LOAD_REQ | | Frequency tuning word update request from the SPI. This control is paged by the CHANNEL_PAGE bits in Register 0x008. | 0x0 | R/W |
| | | | | 0 | No FTW update. | | |
| | | | | 1 | 0 to 1 transition loads the FTW. | | |
| 0x132 | DDSC_FTW0 | [7:0] | DDSC_FTW[7:0] | | Sets the channel datapath NCO FTW. If DDSC_MODULUS_EN is low, the main datapath NCO frequency = $f_{DAC}$ × (DDSC_FTW/$2^{48}$). If DDSC_MODULUS_EN is high, main datapath NCO frequency = $f_{DAC}$ × (DDSC_FTW + DDSC_ACC_DELTA/DDSC_ACC_MODULUS)/$2^{48}$. DDSC_ACC_DELTA must be > 0. DDSC_ACC_DELTA must be > DDSC_ACC_MODULUS. This control is paged by the CHANNEL_PAGE bits in Register 0x008. | 0x0 | R/W |
| 0x133 | DDSC_FTW1 | [7:0] | DDSC_FTW[15:8] | | Sets the channel datapath NCO FTW. If DDSC_MODULUS_EN is low, the main datapath NCO frequency = $f_{DAC}$ × (DDSC_FTW/$2^{48}$). If DDSC_MODULUS_EN is high, main datapath NCO frequency = $f_{DAC}$ × (DDSC_FTW + DDSC_ACC_DELTA/DDSC_ACC_MODULUS)/$2^{48}$. DDSC_ACC_DELTA must be > 0. DDSC_ACC_DELTA must be > DDSC_ACC_MODULUS. This control is paged by the CHANNEL_PAGE bits in Register 0x008. | 0x0 | R/W |
| 0x134 | DDSC_FTW2 | [7:0] | DDSC_FTW[23:16] | | Sets the channel datapath NCO FTW. If DDSC_MODULUS_EN is low, the main datapath NCO frequency = $f_{DAC}$ × (DDSC_FTW/$2^{48}$). If DDSC_MODULUS_EN is high, main datapath NCO frequency = $f_{DAC}$ × (DDSC_FTW + DDSC_ACC_DELTA/DDSC_ACC_MODULUS)/$2^{48}$. DDSC_ACC_DELTA must be > 0. DDSC_ACC_DELTA must be > DDSC_ACC_MODULUS. This control is paged by the CHANNEL_PAGE bits in Register 0x008. | 0x0 | R/W |
| 0x135 | DDSC_FTW3 | [7:0] | DDSC_FTW[31:24] | | Sets the channel datapath NCO FTW. If DDSC_MODULUS_EN is low, the main datapath NCO frequency = $f_{DAC}$ × (DDSC_ | 0x0 | R/W |

## REGISTER DETAILS

*Table 55. Register Details*

| Addr. | Name | Bits | Bit Name | Settings | Description | Reset | Access |
|---|---|---|---|---|---|---|---|
| | | | | | FTW/$2^{48}$). If DDSC_MODULUS_EN is high, main datapath NCO frequency = $f_{DAC}$ × (DDSC_FTW + DDSC_ACC_DELTA/DDSC_ ACC_MODULUS)/$2^{48}$. DDSC_ACC_DELTA must be > 0. DDSC_ACC_DELTA must be > DDSC_ACC_MODULUS. This control is paged by the CHANNEL_PAGE bits in Register 0x008. | | |
| 0x136 | DDSC_FTW4 | [7:0] | DDSC_FTW[39:32] | | Sets the channel datapath NCO FTW. If DDSC_MODULUS_EN is low, the main datapath NCO frequency = $f_{DAC}$ × (DDSC_ FTW/$2^{48}$). If DDSC_MODULUS_EN is high, main datapath NCO frequency = $f_{DAC}$ × (DDSC_FTW + DDSC_ACC_DELTA/DDSC_ ACC_MODULUS)/$2^{48}$. DDSC_ACC_DELTA must be > 0. DDSC_ACC_DELTA must be > DDSC_ACC_MODULUS. This control is paged by the CHANNEL_PAGE bits in Register 0x008. | 0x0 | R/W |
| 0x137 | DDSC_FTW5 | [7:0] | DDSC_FTW[47:40] | | Sets the channel datapath NCO FTW. If DDSC_MODULUS_EN is low, the main datapath NCO frequency = $f_{DAC}$ × (DDSC_ FTW/$2^{48}$). If DDSC_MODULUS_EN is high, main datapath NCO frequency = $f_{DAC}$ × (DDSC_FTW + DDSC_ACC_DELTA/DDSC_ ACC_MODULUS)/$2^{48}$. DDSC_ACC_DELTA must be > 0. DDSC_ACC_DELTA must be > DDSC_ACC_MODULUS. This control is paged by the CHANNEL_PAGE bits in Register 0x008. | 0x0 | R/W |
| 0x138 | DDSC_PHASE_ OFFSET0 | [7:0] | DDSC_NCO_ PHASE_OFFSET[7:0] | | Sets the channel NCO phase offset. Code is in 16-bit, twos complement format. Degrees offset = 180 × (code/$2^{15}$). This control is paged by the CHANNEL_PAGE bits in Register 0x008. | 0x0 | R/W |
| 0x139 | DDSC_PHASE_ OFFSET1 | [7:0] | DDSC_NCO_ PHASE_ OFFSET[15:8] | | Sets the channel NCO phase offset. Code is in 16-bit, twos complement format. Degrees offset = 180 × (code/$2^{15}$). This control is paged by the CHANNEL_PAGE bits in Register 0x008. | 0x0 | R/W |
| 0x13A | DDSC_ACC_ MODULUS0 | [7:0] | DDSC_ACC_ MODULUS[7:0] | | Sets DDSC_ACC_MODULUS. If DDSC_ MODULUS_EN is high, the main datapath NCO frequency = $f_{DAC}$ × (DDSC_FTW + DDSC_ ACC_DELTA/DDSC_ACC_MODULUS)/$2^{48}$. DDSC_ACC_DELTA must be > 0. DDSC_ACC_ DELTA must be > DDSC_ACC_MODULUS. This control is paged by the CHANNEL_ PAGE bits in Register 0x008. | 0x0 | R/W |
| 0x13B | DDSC_ACC_ MODULUS1 | [7:0] | DDSC_ACC_ MODULUS[15:8] | | Sets DDSC_ACC_MODULUS. If DDSC_ MODULUS_EN is high, the main datapath NCO frequency = $f_{DAC}$ × (DDSC_FTW + DDSC_ ACC_DELTA/DDSC_ACC_MODULUS)/$2^{48}$. DDSC_ACC_DELTA must be > 0. DDSC_ACC_ DELTA must be > DDSC_ACC_MODULUS. This control is paged by the CHANNEL_ PAGE bits in Register 0x008. | 0x0 | R/W |
| 0x13C | DDSC_ACC_ | [7:0] | DDSC_ACC_ | | Sets DDSC_ACC_MODULUS. If DDSC_ | 0x0 | R/W |

## REGISTER DETAILS

*Table 55. Register Details*

| Addr. | Name | Bits | Bit Name | Settings | Description | Reset | Access |
|---|---|---|---|---|---|---|---|
| | MODULUS2 | | MODULUS[23:16] | | MODULUS_EN is high, the main datapath NCO frequency = $f_{DAC}$ × (DDSC_FTW + DDSC_ ACC_DELTA/DDSC_ACC_MODULUS)/$2^{48}$. DDSC_ACC_DELTA must be > 0. DDSC_ACC_ DELTA must be > DDSC_ACC_MODULUS. This control is paged by the CHANNEL_ PAGE bits in Register 0x008. | | |
| 0x13D | DDSC_ACC_ MODULUS3 | [7:0] | DDSC_ACC_ MODULUS[31:24] | | Sets DDSC_ACC_MODULUS. If DDSC_ MODULUS_EN is high, the main datapath NCO frequency = $f_{DAC}$ × (DDSC_FTW + DDSC_ ACC_DELTA/DDSC_ACC_MODULUS)/$2^{48}$. DDSC_ACC_DELTA must be > 0. DDSC_ACC_ DELTA must be > DDSC_ACC_MODULUS. This control is paged by the CHANNEL_ PAGE bits in Register 0x008. | 0x0 | R/W |
| 0x13E | DDSC_ACC_ MODULUS4 | [7:0] | DDSC_ACC_ MODULUS[39:32] | | Sets DDSC_ACC_MODULUS. If DDSC_ MODULUS_EN is high, the main datapath NCO frequency = $f_{DAC}$ × (DDSC_FTW + DDSC_ ACC_DELTA/DDSC_ACC_MODULUS)/$2^{48}$. DDSC_ACC_DELTA must be > 0. DDSC_ACC_ DELTA must be > DDSC_ACC_MODULUS. This control is paged by the CHANNEL_ PAGE bits in Register 0x008. | 0x0 | R/W |
| 0x13F | DDSC_ACC_ MODULUS5 | [7:0] | DDSC_ACC_ MODULUS[47:40] | | Sets DDSC_ACC_MODULUS. If DDSC_ MODULUS_EN is high, the main datapath NCO frequency = $f_{DAC}$ × (DDSC_FTW + DDSC_ ACC_DELTA/DDSC_ACC_MODULUS)/$2^{48}$. DDSC_ACC_DELTA must be > 0. DDSC_ACC_ DELTA must be > DDSC_ACC_MODULUS. This control is paged by the CHANNEL_ PAGE bits in Register 0x008. | 0x0 | R/W |
| 0x140 | DDSC_ACC_DELTA0 | [7:0] | DDSC_ACC_ DELTA[7:0] | | Sets DDSC_ACC_DELTA. If DDSC_ MODULUS_EN is high, the main datapath NCO frequency = $f_{DAC}$ × (DDSC_FTW + DDSC_ ACC_DELTA/DDSC_ACC_MODULUS)/$2^{48}$. DDSC_ACC_DELTA must be > 0. DDSC_ ACC_DELTA must be > DDSC_ACC_ MODULUS. This control is paged by the CHANNEL_PAGE bits in Register 0x008. | 0x0 | R/W |
| 0x141 | DDSC_ACC_ DELTA1 | [7:0] | DDSC_ACC_ DELTA[15:8] | | Sets DDSC_ACC_DELTA. If DDSC_ MODULUS_EN is high, the main datapath NCO frequency = $f_{DAC}$ × (DDSC_FTW + DDSC_ ACC_DELTA/DDSC_ACC_MODULUS)/$2^{48}$. DDSC_ACC_DELTA must be > 0. DDSC_ ACC_DELTA must be > DDSC_ACC_ MODULUS. This control is paged by the CHANNEL_PAGE bits in Register 0x008. | 0x0 | R/W |
| 0x142 | DDSC_ACC_DELTA2 | [7:0] | DDSC_ACC_ DELTA[23:16] | | Sets DDSC_ACC_DELTA. If DDSC_ MODULUS_EN is high, the main datapath NCO frequency = $f_{DAC}$ × (DDSC_FTW + DDSC_ | 0x0 | R/W |

## REGISTER DETAILS

*Table 55. Register Details*

| Addr. | Name | Bits | Bit Name | Settings | Description | Reset | Access |
|---|---|---|---|---|---|---|---|
| | | | | | ACC_DELTA/DDSC_ACC_MODULUS)/$2^{48}$. DDSC_ACC_DELTA must be > 0. DDSC_ ACC_DELTA must be > DDSC_ACC_ MODULUS. This control is paged by the CHANNEL_PAGE bits in Register 0x008. | | |
| 0x143 | DDSC_ACC_DELTA3 | [7:0] | DDSC_ACC_ DELTA[31:24] | | Sets DDSC_ACC_DELTA. If DDSC_ MODULUS_EN is high, the main datapath NCO frequency = $f_{DAC}$ × (DDSC_FTW + DDSC_ ACC_DELTA/DDSC_ACC_MODULUS)/$2^{48}$. DDSC_ACC_DELTA must be > 0. DDSC_ ACC_DELTA must be > DDSC_ACC_ MODULUS. This control is paged by the CHANNEL_PAGE bits in Register 0x008. | 0x0 | R/W |
| 0x144 | DDSC_ACC_DELTA4 | [7:0] | DDSC_ACC_ DELTA[39:32] | | Sets DDSC_ACC_DELTA. If DDSC_ MODULUS_EN is high, the main datapath NCO frequency = $f_{DAC}$ × (DDSC_FTW + DDSC_ ACC_DELTA/DDSC_ACC_MODULUS)/$2^{48}$. DDSC_ACC_DELTA must be > 0. DDSC_ ACC_DELTA must be > DDSC_ACC_ MODULUS. This control is paged by the CHANNEL_PAGE bits in Register 0x008. | 0x0 | R/W |
| 0x145 | DDSC_ACC_DELTA5 | [7:0] | DDSC_ACC_ DELTA[47:40] | | Sets DDSC_ACC_DELTA. If DDSC_ MODULUS_EN is high, the main datapath NCO frequency = $f_{DAC}$ × (DDSC_FTW + DDSC_ ACC_DELTA/DDSC_ACC_MODULUS)/$2^{48}$. DDSC_ACC_DELTA must be > 0. DDSC_ ACC_DELTA must be > DDSC_ACC_ MODULUS. This control is paged by the CHANNEL_PAGE bits in Register 0x008. | 0x0 | R/W |
| 0x146 | CHNL_GAIN0 | [7:0] | CHNL_GAIN[7:0] | | Sets the scalar channel gain value. This control is paged by the CHANNEL_PAGE bits in Register 0x008. Channel gain = CHNL_GAIN/$2^{11}$. | 0x0 | R/W |
| 0x147 | CHNL_GAIN1 | [7:4] | RESERVED | | Reserved. | 0x0 | R |
| | | [3:0] | CHNL_GAIN[11:8] | | Sets the scalar channel gain value. This control is paged by the CHANNEL_PAGE bits in Register 0x008. Channel gain = CHNL_GAIN/$2^{11}$. | 0x8 | R/W |
| 0x148 | DC_CAL_TONE0 | [7:0] | DC_TEST_INPUT_ AMPLITUDE[7:0] | | DC test tone amplitude. This amplitude goes to both I and Q paths. Set these bits to 0x50FF for a full-scale tone and ensure DDSC_EN_DC_INPUT in Register 0x130 Bit 0 is set to 1. This control is paged by the CHANNEL_PAGE control in Register 0x008. | 0x0 | R/W |
| 0x149 | DC_CAL_TONE1 | [7:0] | DC_TEST_INPUT_ AMPLITUDE[15:8] | | DC test tone amplitude. This amplitude goes to both I and Q paths. Set to 0x50FF for a full-scale tone and ensure that DDSC_EN_DC_INPUT (Register 0x130, Bit 0) is set to 1. This control is paged by the CHANNEL_PAGE bits in Register 0x008. | 0x0 | R/W |
| 0x14B | PRBS | 7 | PRBS_GOOD_Q | | DAC1 good data indicator. | 0x0 | R |
| | | | | 1 | Correct PRBS sequence detected. | | |
| | | | | 0 | Incorrect sequence detected. Sticky; reset to 1 by PRBS_RESET. | | |
| | | 6 | PRBS_GOOD_I | | DAC0 good data indicator. | 0x0 | R |

## REGISTER DETAILS

*Table 55. Register Details*

| Addr. | Name | Bits | Bit Name | Settings | Description | Reset | Access |
|---|---|---|---|---|---|---|---|
| | | | | 0 | Incorrect sequence detected. Sticky; reset to 1 by PRBS_RESET. | | |
| | | | | 1 | Correct PRBS sequence detected. | | |
| | | 5 | RESERVED | | Reserved. | 0x0 | R |
| | | 4 | PRBS_INV_Q | | DAC1 data inversion. | 0x1 | R/W |
| | | | | 0 | Expect normal data. | | |
| | | | | 1 | Expect inverted data. | | |
| | | 3 | PRBS_INV_I | | DAC0 data inversion. | 0x0 | R/W |
| | | | | 0 | Expect normal data. | | |
| | | | | 1 | Expect inverted data. | | |
| | | 2 | PRBS_MODE | | Select which PRBS polynomial is used for the datapath PRBS test. | 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 | | DAC0 PRBS error count. | 0x0 | R |
| 0x14D | PRBS_ERROR_Q | [7:0] | PRBS_COUNT_Q | | DAC1 PRBS error count. | 0x0 | R |
| 0x14E | PRBS_CHANSEL | [7:3] | RESERVED | | Reserved. | 0x0 | R |
| | | [2:0] | PRBS_CHANSEL | | Selects the channel to which the PRBS_ GOOD_x and PRBS_COUNT_x bit field readbacks correspond. | 0x7 | R/W |
| | | | | 0 | Select Channel 0 for PRBS_COUNT_x and PRBS_GOOD_x (Channel 0, DAC0). | | |
| | | | | 1 | Select Channel 1 for PRBS_COUNT_x and PRBS_GOOD_x (Channel 1, DAC0). | | |
| | | | | 2 | Select Channel 2 for PRBS_COUNT_x and PRBS_GOOD_x (Channel 2, DAC0). | | |
| | | | | 3 | Select Channel 3 for PRBS_COUNT_x and PRBS_GOOD_x (Channel 0, DAC1). | | |
| | | | | 4 | Select Channel 4 for PRBS_COUNT_x and PRBS_GOOD_x (Channel 1, DAC1). | | |
| | | | | 5 | Select Channel 5 for PRBS_COUNT_x and PRBS_GOOD_x (Channel 2, DAC1). | | |
| | | | | 6 | OR all channels for PRBS_GOOD_x, sum all channels for PRBS_COUNT_x. | | |
| 0x151 | DECODE_MODE | [7:6] | RESERVED | | Reserved. | 0x0 | R/W |
| | | [5:4] | MSB_MODE | | MSB shuffling control. | 0x0 | R/W |
| | | [3:0] | RESERVED | | Reserved. | 0x0 | R |
| 0x1DE | SPI_ENABLE | [7:2] | RESERVED | | Reserved. | 0x0 | R |
| | | 1 | SPI_EN1 | | Enable SPI control. | 0x1 | R/W |
| | | 0 | SPI_EN0 | | Enable SPI control. | 0x1 | R/W |
| 0x1E2 | DDSM_CAL_FTW0 | [7:0] | DDSM_CAL_ FTW[7:0] | | FTW of the calibration accumulator. This control is paged by the MAINDAC_PAGE bits in Register 0x008. | 0x0 | R/W |

## REGISTER DETAILS

*Table 55. Register Details*

| Addr. | Name | Bits | Bit Name | Settings | Description | Reset | Access |
|---|---|---|---|---|---|---|---|
| 0x1E3 | DDSM_CAL_FTW1 | [7:0] | DDSM_CAL_FTW[15:8] | | FTW of the calibration accumulator. This control is paged by the MAINDAC_PAGE bits in Register 0x008. | 0x0 | R/W |
| 0x1E4 | DDSM_CAL_FTW2 | [7:0] | DDSM_CAL_FTW[23:16] | | FTW of the calibration accumulator. This control is paged by the MAINDAC_PAGE bits in Register 0x008. | 0x0 | R/W |
| 0x1E5 | DDSM_CAL_FTW3 | [7:0] | DDSM_CAL_FTW[31:24] | | FTW of the calibration accumulator. This control is paged by the MAINDAC_PAGE bits in Register 0x008. | 0x0 | R/W |
| 0x1E6 | DDSM_CAL_MODE_DEF | [7:3] | RESERVED | | Reserved. | 0x0 | R |
| | | 2 | DDSM_EN_CAL_ACC | | Enable clock calibration accumulator. This bit must be first set high, and then must load the calibration FTW into Register 0x1E2 to Register 0x1E5 to take effect. This control is paged by the MAINDAC_PAGE bits in Register 0x008. | 0x0 | R/W |
| | | | | 0 | Disabled (does not clock the calibration frequency accumulator). | | |
| | | | | 1 | Enables (turns on the clock to the calibration frequency accumulator). | | |
| | | 1 | DDSM_EN_CAL_DC_INPUT | | Enable dc input to calibration DDS. This control is paged by the MAINDAC_PAGE bits in Register 0x008. | 0x0 | R/W |
| | | | | 0 | Mux in datapath signal to the input of the final DDS. | | |
| | | | | 1 | Mux in dc to the input of the final DDS. | | |
| | | 0 | DDSM_EN_CAL_FREQ_TUNE | | Enable tuning of the signal to calibration frequency for DAC0 only. This control is paged by the MAINDAC_PAGE bits in Register 0x008. | 0x0 | R/W |
| | | | | 0 | Disable calibration frequency tuning. | | |
| | | | | 1 | Enable calibration frequency tuning. | | |
| 0x1E7 | DATAPATH_NCO_SYNC_CFG | [7:2] | RESERVED | | Reserved. | 0x0 | R |
| | | 1 | ALL_NCO_SYNC_ACK | | Acknowledge signal that all the active NCOs are loaded. This bit is the acknowledge indicator for both the START_NCO_SYNC bit (Bit 0 of this register) and the NCORST_AFTER_ROT_EN bit (Register 0x03B, Bit 4) method of resetting the NCOs. This control is paged by the MAINDAC_PAGE bits in Register 0x008. | 0x0 | R |
| | | 0 | START_NCO_SYNC | | Used to start the sync of the NCOs on a rising edge of the SPI bit or SYSREF± signal, depending on which is chosen as the update trigger. Upon receiving a trigger, the FTWs are loaded first, and then a synchronization occurs. This control is paged by the MAINDAC_PAGE bits in Register 0x008. | 0x0 | R/W |
| 0x200 | MASTER_PD | [7:1] | RESERVED | | Reserved. | 0x0 | R |
| | | 0 | SERDES_MASTER_PD | | Powers down the entire JESD204B Rx analog (all eight channels and bias). | 0x1 | R/W |
| 0x201 | PHY_PD | [7:0] | PHY_PD | | SPI override to power down the individual PHYs. Bit 0 controls the SERDIN0± PHY.<br>Bit 1 controls the SERDIN1± PHY.<br>Bit 2 controls the SERDIN2± PHY.<br>Bit 3 controls the SERDIN3± PHY. | 0xEE | R/W |

## REGISTER DETAILS

*Table 55. Register Details*

| Addr. | Name | Bits | Bit Name | Settings | Description | Reset | Access |
|---|---|---|---|---|---|---|---|
| | | | | | Bit 4 controls the SERDIN4± PHY.<br>Bit 5 controls the SERDIN5± PHY.<br>Bit 6 controls the SERDIN6± PHY.<br>Bit 7 controls the SERDIN7± PHY. | | |
| 0x203 | GENERIC_PD | [7:2] | RESERVED | | Reserved. | 0x0 | R |
| | | 1 | PD_$\overline{\text{SYNCOUT0}}$ | | Powers down the $\overline{\text{SYNCOUT0}}$ driver. | 0x0 | R/W |
| | | | | 0 | Enables the $\overline{\text{SYNCOUT0}}$ output pins. | | |
| | | | | 1 | Powers down the $\overline{\text{SYNCOUT0}}$ output pins. | | |
| | | 0 | PD_$\overline{\text{SYNCOUT1}}$ | | Powers down the $\overline{\text{SYNCOUT1}}$ driver. | 0x1 | R/W |
| | | | | 0 | Enables the $\overline{\text{SYNCOUT1}}$ output pins. | | |
| | | | | 1 | Powers down the $\overline{\text{SYNCOUT1}}$ output pins. | | |
| 0x206 | CDR_RESET | [7:1] | RESERVED | | Reserved. | 0x0 | R |
| | | 0 | CDR_PHY_RESET | | PHY reset control bit. Set this bit to 1 to take the PHYs out of reset during device operation. | 0x0 | R/W |
| 0x210 | CBUS_ADDR | [7:0] | SERDES_CBUS_ADDR | | SERDES configuration control register to set SERDES configuration address controls. | 0x0 | R/W |
| 0x212 | CBUS_WRSTROBE_PHY | [7:0] | SERDES_CBUS_WR0 | | SERDES configuration control register to commit the SERDES configuration controls written. | 0x0 | R/W |
| 0x213 | CBUS_WRSTROBE_OTHER | [7:1] | RESERVED | | Reserved. | 0x0 | R |
| | | 0 | SERDES_CBUS_WR1 | | SERDES configuration control register to commit the SERDES configuration controls written. | 0x0 | R/W |
| 0x216 | CBUS_WDATA | [7:0] | SERDES_CBUS_DATA | | SERDES configuration control register to set the SERDES configuration control data. | 0x0 | R/W |
| 0x240 | EQ_BOOST_PHY_3_0 | [7:6] | EQ_BOOST_PHY3 | | Equalizer setting for PHY 3 based on insertion loss of the system. | 0x3 | R/W |
| | | | | 10 | Insertion loss ≤ 11 dB. | | |
| | | | | 11 | Insertion loss > 11 dB. | | |
| | | [5:4] | EQ_BOOST_PHY2 | | Equalizer setting for PHY 2 based on insertion loss of the system. | 0x3 | R/W |
| | | | | 10 | Insertion loss ≤ 11 dB. | | |
| | | | | 11 | Insertion loss > 11 dB. | | |
| | | [3:2] | EQ_BOOST_PHY1 | | Equalizer setting for PHY 1 based on insertion loss of the system. | 0x3 | R/W |
| | | | | 10 | Insertion loss ≤ 11 dB. | | |
| | | | | 11 | Insertion loss > 11 dB. | | |
| | | [1:0] | EQ_BOOST_PHY0 | | Equalizer setting for PHY 0 based on insertion loss of the system. | 0x3 | R/W |
| | | | | 10 | Insertion loss ≤ 11 dB. | | |
| | | | | 11 | Insertion loss > 11 dB. | | |
| 0x234 | CDR_BITINVERSE | | SEL_IF_PARDATAINV_DES_RC_CH | | Output data inversion bit controls. Set Bit x corresponding to PHY x to invert the bit polarity. | 0x66 | R/W |
| | | | | 0 | Not inverted. | | |
| | | | | 1 | Inverted. | | |
| 0x241 | EQ_BOOST_PHY_7_4 | [7:6] | EQ_BOOST_PHY7 | | Equalizer setting for PHY 7 based on insertion loss of the system. | 0x3 | R/W |
| | | | | 10 | Insertion loss ≤ 11 dB. | | |
| | | | | 11 | Insertion loss > 11 dB. | | |

## REGISTER DETAILS

*Table 55. Register Details*

| Addr. | Name | Bits | Bit Name | Settings | Description | Reset | Access |
|---|---|---|---|---|---|---|---|
| | | [5:4] | EQ_BOOST_PHY6 | | Equalizer setting for PHY 6 based on insertion loss of the system. | 0x3 | R/W |
| | | | | 10 | Insertion loss ≤ 11 dB. | | |
| | | | | 11 | Insertion loss > 11 dB. | | |
| | | [3:2] | EQ_BOOST_PHY5 | | Equalizer setting for PHY 5 based on insertion loss of the system. | 0x3 | R/W |
| | | | | 10 | Insertion loss ≤ 11 dB. | | |
| | | | | 11 | Insertion loss > 11 dB. | | |
| | | [1:0] | EQ_BOOST_PHY4 | | Equalizer setting for PHY 4 based on insertion loss of the system. | 0x3 | R/W |
| | | | | 10 | Insertion loss ≤ 11 dB. | | |
| | | | | 11 | Insertion loss > 11 dB. | | |
| 0x242 | EQ_GAIN_PHY_3_0 | [7:6] | EQ_GAIN_PHY3 | | Equalizer gain for PHY 3 based on insertion loss of the system. | 0x3 | R/W |
| | | | | 01 | Insertion loss ≤ 11 dB. | | |
| | | | | 11 | Insertion loss > 11 dB. | | |
| | | [5:4] | EQ_GAIN_PHY2 | | Equalizer gain for PHY 2 based on insertion loss of the system. | 0x3 | R/W |
| | | | | 01 | Insertion loss ≤ 11 dB. | | |
| | | | | 11 | Insertion loss > 11 dB. | | |
| | | [3:2] | EQ_GAIN_PHY1 | | Equalizer gain for PHY 1 based on insertion loss of the system. | 0x3 | R/W |
| | | | | 01 | Insertion loss ≤ 11 dB. | | |
| | | | | 11 | Insertion loss > 11 dB. | | |
| | | [1:0] | EQ_GAIN_PHY0 | | Equalizer gain for PHY 0 based on insertion loss of the system. | 0x3 | R/W |
| | | | | 01 | Insertion loss ≤ 11 dB. | | |
| | | | | 11 | Insertion loss > 11 dB. | | |
| 0x243 | EQ_GAIN_PHY_7_4 | [7:6] | EQ_GAIN_PHY7 | | Equalizer gain for PHY 7 based on insertion loss of the system. | 0x3 | R/W |
| | | | | 01 | Insertion loss ≤ 11 dB. | | |
| | | | | 11 | Insertion loss > 11 dB. | | |
| | | [5:4] | EQ_GAIN_PHY6 | | Equalizer gain for PHY 6 based on insertion loss of the system. | 0x3 | R/W |
| | | | | 01 | Insertion loss ≤ 11 dB. | | |
| | | | | 11 | Insertion loss > 11 dB. | | |
| | | [3:2] | EQ_GAIN_PHY5 | | Equalizer gain for PHY 5 based on insertion loss of the system. | 0x3 | R/W |
| | | | | 01 | Insertion loss ≤ 11 dB. | | |
| | | | | 11 | Insertion loss > 11 dB. | | |
| | | [1:0] | EQ_GAIN_PHY4 | | Equalizer gain for PHY 4 based on insertion loss of the system. | 0x3 | R/W |
| | | | | 01 | Insertion loss ≤ 11 dB. | | |
| | | | | 11 | Insertion loss > 11 dB. | | |
| 0x244 | EQ_FB_PHY_0 | [7:5] | RESERVED | | Reserved. | 0x0 | R |
| | | [4:0] | EQ_PHY_0 | | SERDES equalizer setting for PHY0. Set this control to 0x1F for optimal performance. | 0x19 | R/W |
| 0x245 | EQ_FB_PHY_1 | [7:5] | RESERVED | | Reserved. | 0x0 | R |
| | | [4:0] | EQ_PHY1 | | SERDES equalizer setting for PHY1. Set this control to 0x1F for optimal performance. | 0x19 | R/W |

## REGISTER DETAILS

*Table 55. Register Details*

| Addr. | Name | Bits | Bit Name | Settings | Description | Reset | Access |
|---|---|---|---|---|---|---|---|
| 0x246 | EQ_FB_PHY_2 | [7:5] | RESERVED | | Reserved. | 0x0 | R |
| | | [4:0] | EQ_PHY2 | | SERDES equalizer setting for PHY2. Set this control to 0x1F for optimal performance. | 0x19 | R/W |
| 0x247 | EQ_FB_PHY_3 | [7:5] | RESERVED | | Reserved. | 0x0 | R |
| | | [4:0] | EQ_PHY3 | | SERDES equalizer setting for PHY3. Set this control to 0x1F for optimal performance. | 0x19 | R/W |
| 0x248 | EQ_FB_PHY_4 | [7:5] | RESERVED | | Reserved. | 0x0 | R |
| | | [4:0] | EQ_PHY4 | | SERDES equalizer setting for PHY4. Set this control to 0x1F for optimal performance. | 0x19 | R/W |
| 0x249 | EQ_FB_PHY_5 | [7:5] | RESERVED | | Reserved. | 0x0 | R |
| | | [4:0] | EQ_PHY5 | | SERDES equalizer setting for PHY5. Set this control to 0x1F for optimal performance. | 0x19 | R/W |
| 0x24A | EQ_FB_PHY_6 | [7:5] | RESERVED | | Reserved. | 0x0 | R |
| | | [4:0] | EQ_PHY6 | | SERDES equalizer setting for PHY6. Set this control to 0x1F for optimal performance. | 0x19 | R/W |
| 0x24B | EQ_FB_PHY_7 | [7:5] | RESERVED | | Reserved. | 0x0 | R |
| | | [4:0] | EQ_PHY7 | | SERDES equalizer setting for PHY7. Set this control to 0x1F for optimal performance. | 0x19 | R/W |
| 0x250 | LBT_REG_CNTRL_0 | [7:0] | EN_LBT_DES_RC_CH | | Enable loopback test for desired physical lanes per PHY, with Bit x corresponding to PHY x. | 0x0 | R/W |
| 0x251 | LBT_REG_CNTRL_1 | [7:2] | RESERVED | | Reserved. | 0x0 | R |
| | | 1 | EN_LBT_ HALFRATE_DES_RC | | Enables half rate mode for the loopback test. If this bit is set to 1, the output data rate = 2× the input clock frequency. If this bit is set to 0, the output data rate = the input clock frequency. | 0x1 | R/W |
| | | 0 | INIT_LBT_SYNC_ DES_RC | | Initiate the loopback test by toggling this bit from 0 to 1, then back to 0. | 0x0 | R/W |
| 0x253 | $\overline{\text{SYNCOUT0}}$_CTRL | [7:1] | RESERVED | | Reserved. | 0x0 | R/W |
| | | 0 | SEL_$\overline{\text{SYNCOUT0}}$_ MODE | | This control determines the output driver mode for the $\overline{\text{SYNCOUT0±}}$ pin operation. Both $\overline{\text{SYNCOUT0±}}$ and $\overline{\text{SYNCOUT1±}}$ must be set to the same mode of operation. | 0x0 | R/W |
| | | | | 0 | $\overline{\text{SYNCOUT0±}}$ is set to CMOS output. | | |
| | | | | 1 | $\overline{\text{SYNCOUT0±}}$ is set to LVDS output. | | |
| 0x254 | $\overline{\text{SYNCOUT1}}$_CTRL | [7:1] | RESERVED | | Reserved. | 0x0 | R/W |
| | | 0 | SEL_$\overline{\text{SYNCOUT1}}$_ MODE | | This control determines the output driver mode for the $\overline{\text{SYNCOUT1±}}$ pin operation. Both $\overline{\text{SYNCOUT0±}}$ and $\overline{\text{SYNCOUT1±}}$ must be set to the same mode of operation. | 0x0 | R/W |
| | | | | 0 | $\overline{\text{SYNCOUT1±}}$ is set to CMOS output. | | |
| | | | | 1 | $\overline{\text{SYNCOUT1±}}$ is set to LVDS output. | | |
| 0x280 | PLL_ENABLE_CTRL | [7:3] | RESERVED | | Reserved. | 0x0 | R |
| | | 2 | LOLSTICKYCLEAR_ LCPLL_RC | | Clears out loss of lock bit. | 0x0 | R/W |
| | | 1 | LDSYNTH_LCPLL_RC | | Pulse high to start VCO calibration (without restarting the regulator or remeasuring the temperature). | 0x0 | R/W |
| | | 0 | SERDES_PLL_ STARTUP | | SERDES circuitry blocks are powered off when this bit is set to 0. Set this bit to 1 at the end of the SERDES configuration writes. When this bit is set to 1, it powers up the SERDES PLL blocks and starts the LDO and calibration routine to lock the | 0x1 | R/W |

## REGISTER DETAILS

*Table 55. Register Details*

| Addr. | Name | Bits | Bit Name | Settings | Description | Reset | Access |
|---|---|---|---|---|---|---|---|
| | | | | | PLL automatically to the appropriate lane rate based on the JESD204B mode and interpolation options programmed in the device. The SERDES_PLL_LOCK bit (Register 0x281, Bit 0) reads 1 when the PLL achieves lock. | | |
| 0x281 | PLL_STATUS | [7:1] | RESERVED | | Reserved. | 0x0 | R |
| | | 0 | SERDES_PLL_LOCK | | PLL is locked when this bit is high. | 0x0 | R |
| 0x300 | GENERAL_JRX_ CTRL_0 | [7:4] | RESERVED | | Reserved. | 0x0 | R |
| | | 3 | LINK_MODE | | Reads back 0 when in single-link mode and 1 when in dual-link mode. | 0x0 | R/W |
| | | 2 | LINK_PAGE | | Link paging. This bit selects which link register map is used. This paging affects Register 0x400 to Register 0x4BB. | 0x0 | R/W |
| | | | | 0 | Page QBD0 for Link 0. | | |
| | | | | 1 | Page QBD1 for Link 1. | | |
| | | [1:0] | LINK_EN | | These bits bring up the JESD204B digital Rx when all link parameters are programmed and all clocks are ready. Bit 0 applies to Link 0, whereas Bit 1 applies to Link 1. Link 1 is only available in dual-link mode. | 0x0 | R/W |
| 0x302 | DYN_LINK_ LATENCY_0 | [7:6] | RESERVED | | Reserved. | 0x0 | R |
| | | [5:0] | DYN_LINK_ LATENCY_0 | | Dynamic link latency, Link 0. Latency between the LMFC Rx for Link 0 and the last arriving LMFC boundary in units of PCLK cycles. | 0x0 | R |
| 0x303 | DYN_LINK_ LATENCY_1 | [7:6] | RESERVED | | Reserved. | 0x0 | R |
| | | [5:0] | DYN_LINK_ LATENCY_1 | | Dynamic link latency, Link 1. Latency between the LMFC Rx for Link 1 and the last arriving LMFC boundary in units of PCLK cycles. | 0x0 | R |
| 0x304 | LMFC_DELAY_0 | [7:6] | RESERVED | | Reserved. | 0x0 | R |
| | | [5:0] | LMFC_DELAY_0 | | LMFC delay, Link 0. Delay from the LMFC to the LMFC Rx for Link 0 in units of PCLK cycles. | 0x0 | R/W |
| 0x305 | LMFC_DELAY_1 | [7:6] | RESERVED | | Reserved. | 0x0 | R |
| | | [5:0] | LMFC_DELAY_1 | | LMFC delay, Link 1. Delay from the LMFC to the LMFC Rx for Link 1 in units of PCLK cycles. | 0x0 | R/W |
| 0x306 | LMFC_VAR_0 | [7:6] | RESERVED | | Reserved. | 0x0 | R |
| | | [5:0] | LMFC_VAR_0 | | Variable delay buffer, Link 0. These bits set when data is read from a buffer to be consistent across links and power cycles (in units of PCLK cycles). The maximum value is 0xC. | 0x3F | R/W |
| 0x307 | LMFC_VAR_1 | [7:6] | RESERVED | | Reserved. | 0x0 | R |
| | | [5:0] | LMFC_VAR_1 | | Variable delay buffer, Link 1. These bits set when data is read from a buffer to be con-sistent across links and power cycles (in units of PCLK cycles). The maximum value is 0xC. | 0x3F | R/W |
| 0x308 | XBAR_LN_0_1 | [7:6] | RESERVED | | Reserved. | 0x0 | R |
| | | [5:3] | LOGICAL_LANE1_ SRC | | Logical Lane 1 source. These bits select a physical lane to be mapped onto Logical 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±. | | |

## REGISTER DETAILS

*Table 55. Register Details*

| Addr. | Name | Bits | Bit Name | Settings | Description | Reset | Access |
|---|---|---|---|---|---|---|---|
| | | | | 101 | Data is from SERDIN5±. | | |
| | | | | 110 | Data is from SERDIN6±. | | |
| | | | | 111 | Data is from SERDIN7±. | | |
| | | [2:0] | LOGICAL_LANE0_SRC | | Logical Lane 0 source. These bits select a physical lane to be mapped onto Logical 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] | LOGICAL_LANE3_SRC | | Logical Lane 3 source. These bits select a physical lane to be mapped onto Logical 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±. | | |
| | | | | 100 | Data is from SERDIN4±. | | |
| | | | | 101 | Data is from SERDIN5±. | | |
| | | | | 110 | Data is from SERDIN6±. | | |
| | | | | 111 | Data is from SERDIN7±. | | |
| | | [2:0] | LOGICAL_LANE2_SRC | | Logical Lane 2 source. These bits select a physical lane to be mapped onto Logical 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] | LOGICAL_LANE5_SRC | | Logical Lane 5 source. These bits select a physical lane to be mapped onto Logical 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] | LOGICAL_LANE4_SRC | | Logical Lane 4 source. These bits select a physical lane to be mapped onto Logical Lane 4. | 0x4 | R/W |
| | | | | 000 | Data is from SERDIN0±. | | |
| | | | | 001 | Data is from SERDIN1±. | | |

## REGISTER DETAILS

*Table 55. Register Details*

| 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±. | | |
| 0x30B | XBAR_LN_6_7 | [7:6] | RESERVED | | Reserved. | 0x0 | R |
| | | [5:3] | LOGICAL_LANE7_SRC | | Logical Lane 7 source. These bits select a physical lane to be mapped onto Logical 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] | LOGICAL_LANE6_SRC | | Logical Lane 6 source. These bits select a physical lane to be mapped onto Logical Lane 6. | 0x6 | 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±. | | |
| 0x30C | FIFO_STATUS_REG_0 | [7:0] | LANE_FIFO_FULL | | Bit x corresponds to FIFO full flag for data from SERDINx±. | 0x0 | R |
| 0x30D | FIFO_STATUS_REG_1 | [7:0] | LANE_FIFO_EMPTY | | Bit x corresponds to FIFO empty flag for data from SERDINx±. | 0x0 | R |
| 0x311 | $\overline{\text{SYNCOUT}}$_GEN_0 | [7:4] | RESERVED | | Reserved. | 0x0 | R |
| | | 3 | EOMF_MASK_1 | | Mask end of multiframe (EOMF) from QBD1. Assert $\overline{\text{SYNCOUT1}}$± based on the loss of the multiframe synchronization. | 0x0 | R/W |
| | | | | 0 | Do not assert $\overline{\text{SYNCOUT1}}$± on loss of multiframe. | | |
| | | | | 1 | Assert $\overline{\text{SYNCOUT1}}$± on loss of multiframe. | | |
| | | 2 | EOMF_MASK_0 | | Mask EOMF from QBD0. Assert $\overline{\text{SYNCOUT0}}$± based on the loss of the multiframe synchronization. | 0x0 | R/W |
| | | | | 0 | Do not assert $\overline{\text{SYNCOUT0}}$± on loss of multiframe. | | |
| | | | | 1 | Assert $\overline{\text{SYNCOUT0}}$± on loss of multiframe. | | |
| | | 1 | EOF_MASK_1 | | Mask EOF from QBD1. Assert $\overline{\text{SYNCOUT1}}$± based on loss of frame synchronization. | 0x0 | R/W |
| | | | | 0 | Do not assert $\overline{\text{SYNCOUT1}}$± on loss of frame. | | |
| | | | | 1 | Assert $\overline{\text{SYNCOUT1}}$± on loss of frame. | | |
| | | 0 | EOF_MASK_0 | | Mask EOF from QBD0. Assert $\overline{\text{SYNCOUT0}}$± based on loss of frame synchronization. | 0x0 | R/W |
| | | | | 0 | Do not assert $\overline{\text{SYNCOUT0}}$± on loss of frame. | | |
| | | | | 1 | Assert $\overline{\text{SYNCOUT0}}$± on loss of frame. | | |

## REGISTER DETAILS

*Table 55. Register Details*

| Addr. | Name | Bits | Bit Name | Settings | Description | Reset | Access |
|---|---|---|---|---|---|---|---|
| 0x312 | SYNCOUT_GEN_1 | [7:4] | SYNC_ERR_DUR | | Duration of SYNCOUTx± low for the purposes of the synchronization error report. Duration = (0.5 + code) PCLK cycles. To most closely match the specified value, set these bits as close as possible to f/2 PCLK cycles. These bits are shared between SYNCOUT0± and SYNCOUT1±. | 0x0 | R/W |
| | | [3:0] | RESERVED | | Reserved. | 0x0 | R/W |
| 0x315 | PHY_PRBS_TEST_EN | [7:0] | PHY_TEST_EN | | Enable PHY BER by ungating the clocks. | 0x0 | R/W |
| | | | | 0 | PHY test disable. | | |
| | | | | 1 | PHY test enable. | | |
| 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 | | Starts and stops the PHY PRBS test. | 0x0 | R/W |
| | | | | 0 | Test not started. | | |
| | | | | 1 | Test started. | | |
| | | 0 | PHY_TEST_RESET | | Resets 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 to set the error flag for the 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 to set the error flag for the 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 to set the error flag for the 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 error count from the 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 error count from the 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 error count from the selected lane. | 0x0 | R |
| 0x31D | PHY_PRBS_TEST_ STATUS | [7:0] | PHY_PRBS_PASS | | Reports PHY BER pass/fail for each lane. Bit x is high when Lane x passes. | 0xFF | R |
| 0x31E | PHY_DATA_ | [7:2] | RESERVED | | Reserved. | 0x0 | R |

## REGISTER DETAILS

*Table 55. Register Details*

| Addr. | Name | Bits | Bit Name | Settings | Description | Reset | Access |
|---|---|---|---|---|---|---|---|
| | SNAPSHOT_CTRL | 1 | PHY_GRAB_MODE | | This bit determines whether to use the 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 | | Transitioning this bit from 0 to 1 causes the 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 select. Selects 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 channel select. Select which channel of the DAC to check. | 0x0 | R/W |
| | | | | 00 | Channel 0. | | |
| | | | | 01 | Channel 1. | | |
| | | | | 10 | Channel 2. | | |
| | | 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 bit field is the lower eight bits of the expected DAC sample during the short transport layer test and is | 0x0 | R/W |

## REGISTER DETAILS

*Table 55. Register Details*

| Addr. | Name | Bits | Bit Name | Settings | Description | Reset | Access |
|---|---|---|---|---|---|---|---|
| | | | | | used to compare with the received sample at the JESD204B Rx output. | | |
| 0x32E | SHORT_TPL_TEST_2 | [7:0] | SHORT_TPL_REF_SP_MSB | | Short transport layer test reference sample, MSB. This bit field is the upper eight bits of the expected DAC sample during the short transport layer test and is used to compare with the received sample at the JESD204B Rx output. | 0x0 | R/W |
| 0x32F | SHORT_TPL_TEST_3 | 7 | RESERVED | | Reserved. | 0x0 | R/W |
| | | 6 | SHORT_TPL_IQ_SAMPLE_SEL | | Selects I or Q of the DAC to check the testing of the short transport layer. | | |
| | | | | 0 | Select I data to test. | | |
| | | | | 1 | Select Q data to test. | | |
| | | [5:1] | RESERVED | | Reserved. | 0x0 | R/W |
| | | 0 | SHORT_TPL_FAIL | | Short transport layer test fail. This bit shows if the selected DAC sample matches the expected sample for the short transport layer test. 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 | | Logical lane invert. Each bit of this control inverses the JESD204B deserialized data from one specific JESD204B Rx PHY. Set Bit x high to invert the JESD204B deserialized data on Logical Lane x. | 0x0 | R/W |
| 0x400 | DID_REG | [7:0] | DID_RD | | Received ILAS configuration on Lane 0. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R |
| 0x401 | BID_REG | [7:0] | BID_RD | | Received ILAS configuration on Lane 0. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R |
| 0x402 | LID0_REG | 7 | RESERVED | | Reserved. | 0x0 | R |
| | | 6 | ADJDIR_RD | | Received ILAS configuration on Lane 0. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R |
| | | 5 | PHADJ_RD | | Received ILAS configuration on Lane 0. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R |
| | | [4:0] | LL_LID0 | | Received ILAS LID configuration on Lane 0. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R |
| 0x403 | SCR_L_REG | 7 | SCR_RD | | Received ILAS configuration on Lane 0. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R |
| | | | | 0 | Scrambling is disabled. | | |
| | | | | 1 | Scrambling is enabled. | | |
| | | [6:5] | RESERVED | | Reserved. | 0x0 | R |
| | | [4:0] | L_RD-1 | | Received ILAS configuration on Lane 0. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R |
| | | | | 00000 | 1 lane per converter device. | | |
| | | | | 00001 | 2 lanes per converter device. | | |
| | | | | 00010 | 3 lanes per converter device. | | |
| | | | | 00011 | 4 lanes per converter device. | | |

## REGISTER DETAILS

*Table 55. Register Details*

| Addr. | Name | Bits | Bit Name | Settings | Description | Reset | Access |
|---|---|---|---|---|---|---|---|
| 0x404 | F_REG | [7:0] | F_RD-1 | | Received ILAS configuration on Lane 0. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R |
| | | | | 0 | 1 octet per frame. | | |
| | | | | 1 | 2 octets per frame. | | |
| | | | | 10 | 3 octets per frame. | | |
| | | | | 11 | 4 octets per frame. | | |
| 0x405 | K_REG | [7:5] | RESERVED | | Reserved. | 0x0 | R |
| | | [4:0] | K_RD-1 | | Received ILAS configuration on Lane 0. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R |
| | | | | 00000 | Default value. | | |
| | | | | 11111 | 32 frames per multiframe. | | |
| 0x406 | M_REG | [7:0] | M_RD-1 | | Received ILAS configuration on Lane 0. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R |
| 0x407 | CS_N_REG | [7:6] | CS_RD | | Received ILAS configuration on Lane 0. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R |
| | | 5 | RESERVED | | Reserved. | 0x0 | R |
| | | [4:0] | N_RD-1 | | Received ILAS configuration on Lane 0. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R |
| 0x408 | NP_REG | [7:5] | SUBCLASSV_RD | | Received ILAS configuration on Lane 0. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R |
| | | [4:0] | NP_RD-1 | | Received ILAS configuration on Lane 0. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R |
| 0x409 | S_REG | [7:5] | JESDV_RD-1 | | Received ILAS configuration on Lane 0. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R |
| | | | | 000 | JESD204A. | | |
| | | | | 001 | JESD204B. | | |
| | | [4:0] | S_RD-1 | | Received ILAS configuration on Lane 0. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R |
| 0x40A | HD_CF_REG | 7 | HD_RD | | Received ILAS configuration on Lane 0. This control is paged by the LINK_PAGE control in Register 0x300. | 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. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R |
| 0x40B | RES1_REG | [7:0] | RES1_RD | | Received ILAS configuration on Lane 0. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R |
| 0x40C | RES2_REG | [7:0] | RES2_RD | | Received ILAS configuration on Lane 0. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R |

## REGISTER DETAILS

*Table 55. Register Details*

| Addr. | Name | Bits | Bit Name | Settings | Description | Reset | Access |
|---|---|---|---|---|---|---|---|
| 0x40D | CHECKSUM0_REG | [7:0] | LL_FCHK0 | | Received checksum during ILAS on Lane 0. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R |
| 0x40E | COMPSUM0_REG | [7:0] | LL_FCMP0 | | Computed checksum on Lane 0. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R |
| 0x412 | LID1_REG | [7:5] | RESERVED | | Reserved. | 0x0 | R |
| | | [4:0] | LL_LID1 | | Received ILAS LID configuration on Lane 1. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R |
| 0x415 | CHECKSUM1_REG | [7:0] | LL_FCHK1 | | Received checksum during ILAS on Lane 1. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R |
| 0x416 | COMPSUM1_REG | [7:0] | LL_FCMP1 | | Computed checksum on Lane 1. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R |
| 0x41A | LID2_REG | [7:5] | RESERVED | | Reserved. | 0x0 | R |
| | | [4:0] | LL_LID2 | | Received ILAS LID configuration on Lane 2. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R |
| 0x41D | CHECKSUM2_REG | [7:0] | LL_FCHK2 | | Received checksum during ILAS on Lane 2. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R |
| 0x41E | COMPSUM2_REG | [7:0] | LL_FCMP2 | | Computed checksum on Lane 2. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R |
| 0x422 | LID3_REG | [7:5] | RESERVED | | Reserved. | 0x0 | R |
| | | [4:0] | LL_LID3 | | Received ILAS LID configuration on Lane 3. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R |
| 0x425 | CHECKSUM3_REG | [7:0] | LL_FCHK3 | | Received checksum during ILAS on Lane 3. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R |
| 0x426 | COMPSUM3_REG | [7:0] | LL_FCMP3 | | Computed checksum on Lane 3. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R |
| 0x42A | LID4_REG | [7:5] | RESERVED | | Reserved. | 0x0 | R |
| | | [4:0] | LL_LID4 | | Received ILAS LID configuration on Lane 4. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R |
| 0x42D | CHECKSUM4_REG | [7:0] | LL_FCHK4 | | Received checksum during ILAS on Lane 4. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R |
| 0x42E | COMPSUM4_REG | [7:0] | LL_FCMP4 | | Computed checksum on Lane 4. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R |
| 0x432 | LID5_REG | [7:5] | RESERVED | | Reserved. | 0x0 | R |
| | | [4:0] | LL_LID5 | | Received ILAS LID configuration on Lane 5. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R |
| 0x435 | CHECKSUM5_REG | [7:0] | LL_FCHK5 | | Received checksum during ILAS on Lane 5. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R |
| 0x436 | COMPSUM5_REG | [7:0] | LL_FCMP5 | | Computed checksum on Lane 5. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R |
| 0x43A | LID6_REG | [7:5] | RESERVED | | Reserved. | 0x0 | R |

## REGISTER DETAILS

*Table 55. Register Details*

| Addr. | Name | Bits | Bit Name | Settings | Description | Reset | Access |
|---|---|---|---|---|---|---|---|
| | | [4:0] | LL_LID6 | | Received ILAS LID configuration on Lane 6. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R |
| 0x43D | CHECKSUM6_REG | [7:0] | LL_FCHK6 | | Received checksum during ILAS on Lane 6. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R |
| 0x43E | COMPSUM6_REG | [7:0] | LL_FCMP6 | | Computed checksum on Lane 6. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R |
| 0x442 | LID7_REG | [7:5] | RESERVED | | Reserved. | 0x0 | R |
| | | [4:0] | LL_LID7 | | Received ILAS LID configuration on Lane 7. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R |
| 0x445 | CHECKSUM7_REG | [7:0] | LL_FCHK7 | | Received checksum during ILAS on Lane 7. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R |
| 0x446 | COMPSUM7_REG | [7:0] | LL_FCMP7 | | Computed checksum on Lane 7. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R |
| 0x450 | ILS_DID | [7:0] | DID | | Device (link) identification number. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R/W |
| 0x451 | ILS_BID | [7:0] | BID | | Bank ID, extension to DID. This control is paged by the LINK_PAGE control in Register 0x300. 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 the DAC LMFC. Link information is received on Link Lane 0 as specified in Section 8.3 of JESD204B. Only Link 0 is supported. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R/W |
| | | 5 | PHADJ | | Phase adjustment request to the DAC. Only Link 0 is supported. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R/W |
| | | [4:0] | LID0 | | Lane identification number (within link). This control is paged by the LINK_PAGE control in Register 0x300. 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 |
| 0x453 | ILS_SCR_L | 7 | SCR | | Scramble enabled for the link. This control is paged by the LINK_PAGE control in Register 0x300. | 0x1 | R/W |
| | | | | 0 | Descrambling is disabled. | | |
| | | | | 1 | Descrambling is enabled. | | |
| | | [6:5] | RESERVED | | Reserved. | 0x0 | R |
| | | [4:0] | L-1 | | Number of lanes per converter (minus 1). This control is paged by the LINK_PAGE control in Register 0x300. | 0x7 | R/W |
| 0x454 | ILS_F | [7:0] | F-1 | | Number of octets per frame per lane (minus 1). This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R/W |
| 0x455 | ILS_K | [7:5] | RESERVED | | Reserved. | 0x0 | R |
| | | [4:0] | K-1 | | Number of frames per multiframe (minus 1). This control is paged by the LINK_PAGE control in Register 0x300. | 0x1F | R/W |
| | | | | 11111 | 32 frames per multiframe. | | |

## REGISTER DETAILS

*Table 55. Register Details*

| Addr. | Name | Bits | Bit Name | Settings | Description | Reset | Access |
|---|---|---|---|---|---|---|---|
| 0x456 | ILS_M | [7:0] | M-1 | | Number of subchannels per link (minus 1). This control is paged by the LINK_PAGE control in Register 0x300. | 0x1 | R/W |
| 0x457 | ILS_CS_N | [7:6] | CS | | Number of control bits per sample. Only Link 0 is supported. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R |
| | | 5 | RESERVED | | Reserved. | 0x0 | R |
| | | [4:0] | N-1 | | Converter resolution (minus 1). This control is paged by the LINK_PAGE control in Register 0x300. | 0xF | R/W |
| 0x458 | ILS_NP | [7:5] | SUBCLASSV | | Device subclass version. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R/W |
| | | | | 000 | Subclass 0. | | |
| | | | | 001 | Subclass 1. | | |
| | | [4:0] | NP-1 | | Total number of bits per sample (minus 1). This control is paged by the LINK_PAGE control in Register 0x300. | 0xF | R/W |
| 0x459 | ILS_S | [7:5] | JESDV | | JESD204 version. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R/W |
| | | | | 000 | JESD204A. | | |
| | | | | 001 | JESD204B. | | |
| | | [4:0] | S-1 | | Number of samples per converter per frame cycle (minus 1). This control is paged by the LINK_PAGE control in Register 0x300. | 0x1 | R/W |
| 0x45A | ILS_HD_CF | 7 | HD | | High density format, always set to 1. This control is paged by the LINK_PAGE control in Register 0x300. | 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. Only Link 0 is supported. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R |
| 0x45B | ILS_RES1 | [7:0] | RES1 | | Reserved field 1. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R/W |
| 0x45C | ILS_RES2 | [7:0] | RES2 | | Reserved field 2. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R/W |
| 0x45D | ILS_CHECKSUM | [7:0] | FCHK0 | | Calculated link configuration checksum. This control is paged by the LINK_PAGE control in Register 0x300. 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. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R |
| | | | | 0 | Deskew failed. | | |
| | | | | 1 | Deskew achieved. | | |
| | | 6 | ILD6 | | Interlane deskew status for Lane 6. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R |
| | | | | 0 | Deskew failed. | | |
| | | | | 1 | Deskew achieved. | | |
| | | 5 | ILD5 | | Interlane deskew status for Lane 5. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R |
| | | | | 0 | Deskew failed. | | |
| | | | | 1 | Deskew achieved. | | |

## REGISTER DETAILS

*Table 55. Register Details*

| Addr. | Name | Bits | Bit Name | Settings | Description | Reset | Access |
|---|---|---|---|---|---|---|---|
| | | 4 | ILD4 | | Interlane deskew status for Lane 4. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R |
| | | | | 0 | Deskew failed. | | |
| | | | | 1 | Deskew achieved. | | |
| | | 3 | ILD3 | | Interlane deskew status for Lane 3. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R |
| | | | | 0 | Deskew failed. | | |
| | | | | 1 | Deskew achieved. | | |
| | | 2 | ILD2 | | Interlane deskew status for Lane 2. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R |
| | | | | 0 | Deskew failed. | | |
| | | | | 1 | Deskew achieved. | | |
| | | 1 | ILD1 | | Interlane deskew status for Lane 1. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R |
| | | | | 0 | Deskew failed. | | |
| | | | | 1 | Deskew achieved. | | |
| | | 0 | ILD0 | | Interlane deskew status for Lane 0. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R |
| | | | | 0 | Deskew failed. | | |
| | | | | 1 | Deskew achieved. | | |
| 0x46D | BAD_DISPARITY | 7 | BDE7 | | Bad disparity errors status for Lane 7. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R |
| | | | | 0 | Error count < ETH[7:0] value. | | |
| | | | | 1 | Error count ≥ ETH[7:0] value. | | |
| | | 6 | BDE6 | | Bad disparity errors status for Lane 6. This control is paged by the LINK_PAGE control in Register 0x300. | 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. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R |
| | | | | 0 | Error count < ETH[7:0] value. | | |
| | | | | 1 | Error count ≥ ETH[7:0] value. | | |
| | | 4 | BDE4 | | Bad disparity errors status for Lane 4. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R |
| | | | | 0 | Error count < ETH[7:0] value. | | |
| | | | | 1 | Error count ≥ ETH[7:0] value. | | |
| | | 3 | BDE3 | | Bad disparity errors status for Lane 3. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R |
| | | | | 0 | Error count < ETH[7:0] value. | | |
| | | | | 1 | Error count ≥ ETH[7:0] value. | | |
| | | 2 | BDE2 | | Bad disparity errors status for Lane 2. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R |
| | | | | 0 | Error count < ETH[7:0] value. | | |
| | | | | 1 | Error count ≥ ETH[7:0] value. | | |
| | | 1 | BDE1 | | Bad disparity errors status for Lane 1. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R |
| | | | | 0 | Error count < ETH[7:0] value. | | |
| | | | | 1 | Error count ≥ ETH[7:0] value. | | |
| | | 0 | BDE0 | | Bad disparity errors status for Lane 0. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R |

## REGISTER DETAILS

*Table 55. Register Details*

| Addr. | Name | Bits | Bit Name | Settings | Description | Reset | Access |
|---|---|---|---|---|---|---|---|
| | | | | 0 | Error count < ETH[7:0] value. | | |
| | | | | 1 | Error count ≥ ETH[7:0] value. | | |
| 0x46E | NOT_IN_TABLE | 7 | NIT7 | | Not in table errors status for Lane 7. This control is paged by the LINK_PAGE control in Register 0x300. | 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. This control is paged by the LINK_PAGE control in Register 0x300. | 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. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R |
| | | | | 0 | Error count < ETH[7:0] value. | | |
| | | | | 1 | Error count ≥ ETH[7:0] value. | | |
| | | 4 | NIT4 | | Not in table errors status for Lane 4. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R |
| | | | | 0 | Error count < ETH[7:0] value. | | |
| | | | | 1 | Error count ≥ ETH[7:0] value. | | |
| | | 3 | NIT3 | | Not in table errors status for Lane 3. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R |
| | | | | 0 | Error count < ETH[7:0] value. | | |
| | | | | 1 | Error count ≥ ETH[7:0] value. | | |
| | | 2 | NIT2 | | Not in table errors status for Lane 2. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R |
| | | | | 0 | Error count < ETH[7:0] value. | | |
| | | | | 1 | Error count ≥ ETH[7:0] value. | | |
| | | 1 | NIT1 | | Not in table errors status for Lane 1. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R |
| | | | | 0 | Error count < ETH[7:0] value. | | |
| | | | | 1 | Error count ≥ ETH[7:0] value. | | |
| | | 0 | NIT0 | | Not in table errors status for Lane 0. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R |
| | | | | 0 | Error count < ETH[7:0] value. | | |
| | | | | 1 | Error count ≥ ETH[7:0] value. | | |
| 0x46F | UNEXPECTED_KCHAR | 7 | UEK7 | | Unexpected K character errors status, Lane 7. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R |
| | | | | 0 | Error count < ETH[7:0] value. | | |
| | | | | 1 | Error count ≥ ETH[7:0] value. | | |
| | | 6 | UEK6 | | Unexpected K character errors status, Lane 6. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R |
| | | | | 0 | Error count < ETH[7:0] value. | | |
| | | | | 1 | Error count ≥ ETH[7:0] value. | | |
| | | 5 | UEK5 | | Unexpected K character errors status, Lane 5. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R |
| | | | | 0 | Error count < ETH[7:0] value. | | |
| | | | | 1 | Error count ≥ ETH[7:0] value. | | |

## REGISTER DETAILS

*Table 55. Register Details*

| Addr. | Name | Bits | Bit Name | Settings | Description | Reset | Access |
|---|---|---|---|---|---|---|---|
| | | 4 | UEK4 | | Unexpected K character errors status, Lane 4. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R |
| | | | | 0 | Error count < ETH[7:0] value. | | |
| | | | | 1 | Error count ≥ ETH[7:0] value. | | |
| | | 3 | UEK3 | | Unexpected K character errors status, Lane 3. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R |
| | | | | 0 | Error count < ETH[7:0] value. | | |
| | | | | 1 | Error count ≥ ETH[7:0] value. | | |
| | | 2 | UEK2 | | Unexpected K character errors status, Lane 2. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R |
| | | | | 0 | Error count < ETH[7:0] value. | | |
| | | | | 1 | Error count ≥ ETH[7:0] value. | | |
| | | 1 | UEK1 | | Unexpected K character errors status, Lane 1. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R |
| | | | | 0 | Error count < ETH[7:0] value. | | |
| | | | | 1 | Error count ≥ ETH[7:0] value. | | |
| | | 0 | UEK0 | | Unexpected K character errors status, Lane 0. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R |
| | | | | 0 | Error count < ETH[7:0] value. | | |
| | | | | 1 | Error count ≥ ETH[7:0] value. | | |
| 0x470 | CODE_GRP_SYNC | 7 | CGS7 | | Code group synchronization status for Lane 7. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R |
| | | | | 0 | Synchronization lost. | | |
| | | | | 1 | Synchronization achieved. | | |
| | | 6 | CGS6 | | Code group synchronization status for Lane 6. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R |
| | | | | 0 | Synchronization lost. | | |
| | | | | 1 | Synchronization achieved. | | |
| | | 5 | CGS5 | | Code group synchronization status for Lane 5. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R |
| | | | | 0 | Synchronization lost. | | |
| | | | | 1 | Synchronization achieved. | | |
| | | 4 | CGS4 | | Code group synchronization status for Lane 4. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R |
| | | | | 0 | Synchronization lost. | | |
| | | | | 1 | Synchronization achieved. | | |
| | | 3 | CGS3 | | Code group synchronization status for Lane 3. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R |
| | | | | 0 | Synchronization lost. | | |
| | | | | 1 | Synchronization achieved. | | |

## REGISTER DETAILS

*Table 55. Register Details*

| Addr. | Name | Bits | Bit Name | Settings | Description | Reset | Access |
|---|---|---|---|---|---|---|---|
| | | 2 | CGS2 | | Code group synchronization status for Lane 2. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R |
| | | | | 0 | Synchronization lost. | | |
| | | | | 1 | Synchronization achieved. | | |
| | | 1 | CGS1 | | Code group synchronization status for Lane 1. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R |
| | | | | 0 | Synchronization lost. | | |
| | | | | 1 | Synchronization achieved. | | |
| | | 0 | CGS0 | | Code group synchronization status for Lane 0. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R |
| | | | | 0 | Synchronization lost. | | |
| | | | | 1 | Synchronization achieved. | | |
| 0x471 | FRAME_SYNC | 7 | FS7 | | Frame synchronization status for Lane 7. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R |
| | | | | 0 | Synchronization lost. | | |
| | | | | 1 | Synchronization achieved. | | |
| | | 6 | FS6 | | Frame synchronization status for Lane 6. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R |
| | | | | 0 | Synchronization lost. | | |
| | | | | 1 | Synchronization achieved. | | |
| | | 5 | FS5 | | Frame synchronization status for Lane 5. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R |
| | | | | 0 | Synchronization lost. | | |
| | | | | 1 | Synchronization achieved. | | |
| | | 4 | FS4 | | Frame synchronization status for Lane 4. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R |
| | | | | 0 | Synchronization lost. | | |
| | | | | 1 | Synchronization achieved. | | |
| | | 3 | FS3 | | Frame synchronization status for Lane 3. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R |
| | | | | 0 | Synchronization lost. | | |
| | | | | 1 | Synchronization achieved. | | |
| | | 2 | FS2 | | Frame synchronization status for Lane 2. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R |
| | | | | 0 | Synchronization lost. | | |
| | | | | 1 | Synchronization achieved. | | |
| | | 1 | FS1 | | Frame synchronization status for Lane 1. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R |
| | | | | 0 | Synchronization lost. | | |
| | | | | 1 | Synchronization achieved. | | |

## REGISTER DETAILS

*Table 55. Register Details*

| Addr. | Name | Bits | Bit Name | Settings | Description | Reset | Access |
|---|---|---|---|---|---|---|---|
| | | 0 | FS0 | | Frame synchronization status for Lane 0. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R |
| | | | | 0 | Synchronization lost. | | |
| | | | | 1 | Synchronization achieved. | | |
| 0x472 | GOOD_CHECKSUM | 7 | CKS7 | | Computed checksum status for Lane 7. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R |
| | | | | 0 | Checksum is incorrect. | | |
| | | | | 1 | Checksum is correct. | | |
| | | 6 | CKS6 | | Computed checksum status for Lane 6. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R |
| | | | | 0 | Checksum is incorrect. | | |
| | | | | 1 | Checksum is correct. | | |
| | | 5 | CKS5 | | Computed checksum status for Lane 5. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R |
| | | | | 0 | Checksum is incorrect. | | |
| | | | | 1 | Checksum is correct. | | |
| | | 4 | CKS4 | | Computed checksum status for Lane 4. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R |
| | | | | 0 | Checksum is incorrect. | | |
| | | | | 1 | Checksum is correct. | | |
| | | 3 | CKS3 | | Computed checksum status for Lane 3. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R |
| | | | | 0 | Checksum is incorrect. | | |
| | | | | 1 | Checksum is correct. | | |
| | | 2 | CKS2 | | Computed checksum status for Lane 2. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R |
| | | | | 0 | Checksum is incorrect. | | |
| | | | | 1 | Checksum is correct. | | |
| | | 1 | CKS1 | | Computed checksum status for Lane 1. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R |
| | | | | 0 | Checksum is incorrect. | | |
| | | | | 1 | Checksum is correct. | | |
| | | 0 | CKS0 | | Computed checksum status for Lane 0. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R |
| | | | | 0 | Checksum is incorrect. | | |
| | | | | 1 | Checksum is correct. | | |
| 0x473 | INIT_LANE_SYNC | 7 | ILS7 | | Initial lane synchronization status for Lane 7. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R |
| | | | | 0 | Synchronization lost. | | |
| | | | | 1 | Synchronization achieved. | | |
| | | 6 | ILS6 | | Initial lane synchronization status for Lane 6. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R |
| | | | | 0 | Synchronization lost. | | |
| | | | | 1 | Synchronization achieved. | | |
| | | 5 | ILS5 | | Initial lane synchronization status for Lane 5. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R |
| | | | | 0 | Synchronization lost. | | |

## REGISTER DETAILS

*Table 55. Register Details*

| Addr. | Name | Bits | Bit Name | Settings | Description | Reset | Access |
|---|---|---|---|---|---|---|---|
| | | | | 1 | Synchronization achieved. | | |
| | | 4 | ILS4 | | Initial lane synchronization status for Lane 4. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R |
| | | | | 0 | Synchronization lost. | | |
| | | | | 1 | Synchronization achieved. | | |
| | | 3 | ILS3 | | Initial lane synchronization status for Lane 3. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R |
| | | | | 0 | Synchronization lost. | | |
| | | | | 1 | Synchronization achieved. | | |
| | | 2 | ILS2 | | Initial lane synchronization status for Lane 2. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R |
| | | | | 0 | Synchronization lost. | | |
| | | | | 1 | Synchronization achieved. | | |
| | | 1 | ILS1 | | Initial lane synchronization status for Lane 1. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R |
| | | | | 0 | Synchronization lost. | | |
| | | | | 1 | Synchronization achieved. | | |
| | | 0 | ILS0 | | Initial lane synchronization status for Lane 0. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R |
| | | | | 0 | Synchronization lost. | | |
| | | | | 1 | Synchronization achieved. | | |
| 0x475 | CTRLREG0 | [7:4] | RESERVED | | Reserved. | 0x0 | R/W |
| | | 3 | SOFTRST | | QBD soft reset. Active high synchronous reset. Resets all hardware to power-on state. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R/W |
| | | 2 | FORCESYNCREQ | | Command from application to assert a synchronization request. Active high. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R/W |
| | | 1 | RESERVED | | Reserved. | 0x0 | R |
| | | 0 | REPL_FRM_ENA | | When this level input is set, it enables the replacement of frames received in error. This control is paged by the LINK_PAGE control in Register 0x300. 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 not in table (NIT) and running disparity (RD) errors. This control is paged by the LINK_PAGE control in Register 0x300. Set this bit to 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. | 0x1 | R/W |
| | | | | 0 | NIT has no effect on RD error. | | |
| | | | | 1 | NIT error masks concurrent with RD error. | | |
| | | [3:1] | RESERVED | | Reserved. | 0x0 | R/W |

## REGISTER DETAILS

*Table 55. Register Details*

| Addr. | Name | Bits | Bit Name | Settings | Description | Reset | Access |
|---|---|---|---|---|---|---|---|
| | | 0 | FCHK_N | | Checksum calculation method. This control is paged by the LINK_PAGE control in Register 0x300. 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 | Checksum is calculated by summing the individual fields in the link configuration table as defined in Section 8.3, Table 20 of the JESD204B standard. | | |
| | | | | 1 | Checksum is calculated by summing the registers containing the packed link configuration fields (sum of Register 0x450 to Register 0x45A, modulo 256). | | |
| 0x477 | CTRLREG2 | 7 | ILS_MODE | | Data link layer test mode is enabled when this bit is set to 1. CGS pattern is followed by a perpetual ILAS sequence. This control is paged by the LINK_PAGE control in Register 0x300. 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 | CGS pattern is followed by a perpetual ILAS sequence. | | |
| | | 6 | RESERVED | | Reserved. | 0x0 | R/W |
| | | 5 | REPDATATEST | | Repetitive data test enable using the JTSPAT pattern. To enable the test, Bit 7 of this register must = 0. This control is paged by the LINK_PAGE control in Register 0x300. 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 control is paged by the LINK_PAGE control in Register 0x300. 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 | When this bit = 0, simultaneous errors on multiple lanes are reported as one error. | | |
| | | | | 1 | Selected when this bit = 1 and when REPDATATEST = 1. Detected errors from all lanes are trapped in a counter and sequentially signaled on $\overline{\text{SYNCOUTx±}}$. | | |
| | | 3 | AR_ECNTR | | Automatic reset of error counter. The error counter that causes assertion of $\overline{\text{SYNCOUTx±}}$ is automatically reset to 0 when AR_ECNTR = 1. All other counters are unaffected. This control is paged by the LINK_PAGE control in Register 0x300. 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 ILAS. This control is paged by the LINK_PAGE control in Register 0x300. 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 |

## REGISTER DETAILS

*Table 55. Register Details*

| Addr. | Name | Bits | Bit Name | Settings | Description | Reset | Access |
|---|---|---|---|---|---|---|---|
| 0x47C | ERRORTHRES | [7:0] | ETH | | Error counter threshold value. These bits set when a $\overline{\text{SYNCOUTx±}}$ error or interrupt is sent due to BD, NIT, or UEK errors. This control is paged by the LINK_PAGE control in Register 0x300. 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 | | $\overline{\text{SYNCOUTx±}}$ assertion enable mask for BD, NIT, and UEK error conditions. This control is paged by the LINK_PAGE control in Register 0x300. Active high, $\overline{\text{SYNCOUTx±}}$ 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, $\overline{\text{SYNCOUTx±}}$ 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 character error (UEK). | 0x7 | R/W |
| 0x480 | ECNT_CTRL0 | [7:6] | RESERVED | | Reserved. | 0x0 | R |
| | | [5:3] | ECNT_ENA0 | | Error counter enables for Lane 0. This control is paged by the LINK_PAGE control in Register 0x300. Counters of each lane are addressed as follows:<br>Bit 5 = unexpected K character error (UEK).<br>Bit 4 = not in table error (NIT).<br>Bit 3 = bad disparity error (BDE). | 0x7 | R/W |
| | | [2:0] | ECNT_RST0 | | Reset error counters for Lane 0. This control is paged by the LINK_PAGE control in Register 0x300. Counters of each lane are addressed as follows:<br>Bit 2 = bad disparity error (BDE).<br>Bit 1 = not in table error (NIT).<br>Bit 0 = unexpected K character error (UEK). | 0x7 | R/W |
| 0x481 | ECNT_CTRL1 | [7:6] | RESERVED | | Reserved. | 0x0 | R |
| | | [5:3] | ECNT_ENA1 | | Error counter enables for Lane 1. This control is paged by the LINK_PAGE control in Register 0x300. Counters of each lane are addressed as follows:<br>Bit 5 = unexpected K character error (UEK).<br>Bit 4 = not in table error (NIT).<br>Bit 3 = bad disparity error (BDE). | 0x7 | R/W |
| | | [2:0] | ECNT_RST1 | | Reset error counters for Lane 1. This control is paged by the LINK_PAGE control in Register 0x300. Counters of each lane are addressed as follows:<br>Bit 2 = unexpected K character error (UEK).<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 |

## REGISTER DETAILS

*Table 55. Register Details*

| Addr. | Name | Bits | Bit Name | Settings | Description | Reset | Access |
|---|---|---|---|---|---|---|---|
| | | [5:3] | ECNT_ENA2 | | Error counter enables for Lane 2. This control is paged by the LINK_PAGE control in Register 0x300. Counters of each lane are addressed as follows:<br>Bit 5 = unexpected K character error (UEK).<br>Bit 4 = not in table error (NIT).<br>Bit 3 = bad disparity error (BDE). | 0x7 | R/W |
| | | [2:0] | ECNT_RST2 | | Reset error counters for Lane 2. This control is paged by the LINK_PAGE control in Register 0x300. Counters of each lane are addressed as follows:<br>Bit 2 = unexpected K character error (UEK).<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 counter enables for Lane 3. This control is paged by the LINK_PAGE control in Register 0x300. Counters of each lane are addressed as follows:<br>Bit 5 = unexpected K character error (UEK).<br>Bit 4 = not in table error (NIT).<br>Bit 3 = bad disparity error (BDE). | 0x7 | R/W |
| | | [2:0] | ECNT_RST3 | | Reset error counters for Lane 3. This control is paged by the LINK_PAGE control in Register 0x300. Counters of each lane are addressed as follows:<br>Bit 2 = unexpected K character error (UEK).<br>Bit 1 = not in table error (NIT).<br>Bit 0 = bad disparity error (BDE). | 0x7 | R/W |
| 0x484 | ECNT_CTRL4 | [7:6] | RESERVED | | Reserved. | 0x0 | R |
| | | [5:3] | ECNT_ENA4 | | Error counter enables for Lane 4. This control is paged by the LINK_PAGE control in Register 0x300. Counters of each lane are addressed as follows:<br>Bit 5 = unexpected K character error (UEK).<br>Bit 4 = not in table error (NIT).<br>Bit 3 = bad disparity error (BDE). | 0x7 | R/W |
| | | [2:0] | ECNT_RST4 | | Reset error counters for Lane 4. This control is paged by the LINK_PAGE control in Register 0x300. Counters of each lane are addressed as follows:<br>Bit 2 = unexpected K character error (UEK).<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 counter enables for Lane 5. This control is paged by the LINK_PAGE control in Register 0x300. Counters of each lane are addressed as follows:<br>Bit 5 = unexpected K character error (UEK).<br>Bit 4 = not in table error (NIT).<br>Bit 3 = bad disparity error (BDE). | 0x7 | R/W |
| | | [2:0] | ECNT_RST5 | | Reset error counters for Lane 5. This control is paged by the LINK_PAGE control in Register 0x300. Counters of each lane are addressed as follows:<br>Bit 2 = unexpected K character error (UEK).<br>Bit 1 = not in table error (NIT).<br>Bit 0 = bad disparity error (BDE). | 0x7 | R/W |

## REGISTER DETAILS

*Table 55. Register Details*

| Addr. | Name | Bits | Bit Name | Settings | Description | Reset | Access |
|---|---|---|---|---|---|---|---|
| 0x486 | ECNT_CTRL6 | [7:6] | RESERVED | | Reserved. | 0x0 | R |
| | | [5:3] | ECNT_ENA6 | | Error counter enables for Lane 6. This control is paged by the LINK_PAGE control in Register 0x300. Counters of each lane are addressed as follows:<br>Bit 5 = unexpected K character error (UEK).<br>Bit 4 = not in table error (NIT).<br>Bit 3 = bad disparity error (BDE). | 0x7 | R/W |
| | | [2:0] | ECNT_RST6 | | Reset error counters for Lane 6. This control is paged by the LINK_PAGE control in Register 0x300. Counters of each lane are addressed as follows:<br>Bit 2 = unexpected K character error (UEK).<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 counter enables for Lane 7. This control is paged by the LINK_PAGE control in Register 0x300. Counters of each lane are addressed as follows:<br>Bit 5 = unexpected K character error (UEK).<br>Bit 4 = not in table error (NIT).<br>Bit 3 = bad disparity error (BDE). | 0x7 | R/W |
| | | [2:0] | ECNT_RST7 | | Reset error counters for Lane 7. This control is paged by the LINK_PAGE control in Register 0x300. Counters of each lane are addressed as follows:<br>Bit 2 = unexpected K character error (UEK).<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. This control is paged by the LINK_PAGE control in Register 0x300. 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 character error (UEK).<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. This control is paged by the LINK_PAGE control in Register 0x300. 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 character error (UEK).<br>Bit 1 = not in table error (NIT).<br>Bit 0 = bad disparity error (BDE). | 0x7 | R/W |

## REGISTER DETAILS

*Table 55. Register Details*

| Addr. | Name | Bits | Bit Name | Settings | Description | Reset | Access |
|---|---|---|---|---|---|---|---|
| | | | | | 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. | | |
| 0x48A | ECNT_TCH2 | [7:3] | RESERVED | | Reserved. | 0x0 | R |
| | | [2:0] | ECNT_TCH2 | | Terminal count hold enable of error counters for Lane 2. This control is paged by the LINK_PAGE control in Register 0x300. 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 character error (UEK).<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 |
| 0x48B | ECNT_TCH3 | [7:3] | RESERVED | | Reserved. | 0x0 | R |
| | | [2:0] | ECNT_TCH3 | | Terminal count hold enable of error counters for Lane 3. This control is paged by the LINK_PAGE control in Register 0x300. 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 character error (UEK).<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. This control is paged by the LINK_PAGE control in Register 0x300. 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 character error (UEK).<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. This control is paged by the LINK_PAGE control in Register 0x300. 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: | 0x7 | R/W |

## REGISTER DETAILS

*Table 55. Register Details*

| Addr. | Name | Bits | Bit Name | Settings | Description | Reset | Access |
|---|---|---|---|---|---|---|---|
| | | | | | Bit 2 = unexpected K character error (UEK).<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. | | |
| 0x48E | ECNT_TCH6 | [7:3] | RESERVED | | Reserved. | 0x0 | R |
| | | [2:0] | ECNT_TCH6 | | Terminal count hold enable of error counters for Lane 6. This control is paged by the LINK_PAGE control in Register 0x300. 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 character error (UEK).<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. This control is paged by the LINK_PAGE control in Register 0x300. 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 character error (UEK).<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. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R |
| | | [2:0] | ECNT_TCR0 | | Terminal count reached indicator of error counters for Lane 0. This control is paged by the LINK_PAGE control in Register 0x300. Set these bits to 1 when the corresponding counter terminal count value of 0xFF is reached. If ECNT_TCHx is set, the terminal count value for the corresponding counter is held until the counter is reset by the user; otherwise, the counter rolls over and continues counting. Counters of each lane are addressed as follows:<br>Bit 2 = unexpected K character error (UEK).<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 |

## REGISTER DETAILS

*Table 55. Register Details*

| Addr. | Name | Bits | Bit Name | Settings | Description | Reset | Access |
|---|---|---|---|---|---|---|---|
| | | 3 | LANE_ENA1 | | This output indicates if Lane 1 is enabled. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R |
| | | [2:0] | ECNT_TCR1 | | Terminal count reached indicator of error counters for Lane 1. This control is paged by the LINK_PAGE control in Register 0x300. Set these bits to 1 when the corresponding counter terminal count value of 0xFF is reached. If ECNT_TCHx is set, the terminal count value for the corresponding counter is held until the counter is reset by the user; otherwise, the counter rolls over and continues counting. Counters of each lane are addressed as follows:<br>Bit 2 = unexpected K character error (UEK).<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. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R |
| | | [2:0] | ECNT_TCR2 | | Terminal count reached indicator of error counters for Lane 2. This control is paged by the LINK_PAGE control in Register 0x300. Set these bits to 1 when the corresponding counter terminal count value of 0xFF is reached. If ECNT_TCHx is set, the terminal count value for the corresponding counter is held until the counter is reset by the user; otherwise, the counter rolls over and continues counting. Counters of each lane are addressed as follows:<br>Bit 2 = unexpected K character error (UEK).<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. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R |
| | | [2:0] | ECNT_TCR3 | | Terminal count reached indicator of error counters for Lane 3. This control is paged by the LINK_PAGE control in Register 0x300. Set these bits to 1 when the corresponding counter terminal count value of 0xFF is reached. If ECNT_TCHx is set, the terminal count value for the corresponding counter is held until the counter is reset by the user; otherwise, the counter rolls over and continues counting. Counters of each lane are addressed as follows:<br>Bit 2 = unexpected K character error (UEK).<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. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R |
| | | [2:0] | ECNT_TCR4 | | Terminal count reached indicator of error counters for Lane 4. This control is paged by the LINK_PAGE control in Register 0x300. Set these bits to 1 when | 0x0 | R |

## REGISTER DETAILS

*Table 55. Register Details*

| Addr. | Name | Bits | Bit Name | Settings | Description | Reset | Access |
|---|---|---|---|---|---|---|---|
| | | | | | the corresponding counter terminal count value of 0xFF is reached. If ECNT_TCHx is set, the terminal count value for the corresponding counter is held until the counter is reset by the user; otherwise, the counter rolls over and continues counting. Counters of each lane are addressed as follows:<br>Bit 2 = unexpected K character error (UEK).<br>Bit 1 = not in table error (NIT).<br>Bit 0 = bad disparity error (BDE). | | |
| 0x495 | ECNT_STAT5 | [7:4] | RESERVED | | Reserved. | 0x0 | R |
| | | 3 | LANE_ENA5 | | This output indicates if Lane 5 is enabled. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R |
| | | [2:0] | ECNT_TCR5 | | Terminal count reached indicator of error counters for Lane 5. This control is paged by the LINK_PAGE control in Register 0x300. Set these bits to 1 when the corresponding counter terminal count value of 0xFF is reached. If ECNT_TCHx is set, the terminal count value for the corresponding counter is held until the counter is reset by the user; otherwise, the counter rolls over and continues counting. Counters of each lane are addressed as follows:<br>Bit 2 = unexpected K character error (UEK).<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. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R |
| | | [2:0] | ECNT_TCR6 | | Terminal count reached indicator of error counters for Lane 6. This control is paged by the LINK_PAGE control in Register 0x300. Set these bits to 1 when the corresponding counter terminal count value of 0xFF is reached. If ECNT_TCHx is set, the terminal count value for the corresponding counter is held until the counter is reset by the user; otherwise, the counter rolls over and continues counting. Counters of each lane are addressed as follows:<br>Bit 2 = unexpected K character error (UEK).<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. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R |
| | | [2:0] | ECNT_TCR7 | | Terminal count reached indicator of error counters for Lane 7. This control is paged by the LINK_PAGE control in Register 0x300. Set these bits to 1 when the corresponding counter terminal count value of 0xFF is reached. If ECNT_TCHx is set, the terminal count value for the corresponding counter is held until the counter is reset by the user; otherwise, the counter rolls over and continues counting. Counters of each lane are addressed as follows: | 0x0 | R |

## REGISTER DETAILS

*Table 55. Register Details*

| Addr. | Name | Bits | Bit Name | Settings | Description | Reset | Access |
|---|---|---|---|---|---|---|---|
| | | | | | Bit 2 = unexpected K character error (UEK).<br>Bit 1 = not in table error (NIT).<br>Bit 0 = bad disparity error (BDE). | | |
| 0x4B0 | LINK_STATUS0 | 7 | BDE0 | | Bad disparity errors status for Lane 0. This control is paged by the LINK_PAGE control in Register 0x300. | 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. This control is paged by the LINK_PAGE control in Register 0x300. | 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. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R |
| | | | | 0 | Error count < ETH[7:0] value. | | |
| | | | | 1 | Error count ≥ ETH[7:0] value. | | |
| | | 4 | ILD0 | | Interlane deskew status for Lane 0. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R |
| | | | | 0 | Deskew failed. | | |
| | | | | 1 | Deskew achieved. | | |
| | | 3 | ILS0 | | Initial lane synchronization status for Lane 0. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R |
| | | | | 0 | Synchronization lost. | | |
| | | | | 1 | Synchronization achieved. | | |
| | | 2 | CKS0 | | Computed checksum status for Lane 0. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R |
| | | | | 0 | Checksum is incorrect. | | |
| | | | | 1 | Checksum is correct. | | |
| | | 1 | FS0 | | Frame synchronization status for Lane 0. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R |
| | | | | 0 | Synchronization lost. | | |
| | | | | 1 | Synchronization achieved. | | |
| | | 0 | CGS0 | | Code group synchronization status for Lane 0. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R |
| | | | | 0 | Synchronization lost. | | |
| | | | | 1 | Synchronization achieved. | | |
| 0x4B1 | LINK_STATUS1 | 7 | BDE1 | | Bad disparity errors status for Lane 1. This control is paged by the LINK_PAGE control in Register 0x300. | 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. This control is paged by the LINK_PAGE control in Register 0x300. | 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. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R |
| | | | | 0 | Error count < ETH[7:0] value. | | |

## REGISTER DETAILS

*Table 55. Register Details*

| Addr. | Name | Bits | Bit Name | Settings | Description | Reset | Access |
|---|---|---|---|---|---|---|---|
| | | | | 1 | Error count ≥ ETH[7:0] value. | | |
| | | 4 | ILD1 | | Interlane deskew status for Lane 1. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R |
| | | | | 0 | Deskew failed. | | |
| | | | | 1 | Deskew achieved. | | |
| | | 3 | ILS1 | | Initial lane synchronization status for Lane 1. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R |
| | | | | 0 | Synchronization lost. | | |
| | | | | 1 | Synchronization achieved. | | |
| | | 2 | CKS1 | | Computed checksum status for Lane 1. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R |
| | | | | 0 | Checksum is incorrect. | | |
| | | | | 1 | Checksum is correct. | | |
| | | 1 | FS1 | | Frame synchronization status for Lane 1. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R |
| | | | | 0 | Synchronization lost. | | |
| | | | | 1 | Synchronization achieved. | | |
| | | 0 | CGS1 | | Code group synchronization status for Lane 1. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R |
| | | | | 0 | Synchronization lost. | | |
| | | | | 1 | Synchronization achieved. | | |
| 0x4B2 | LINK_STATUS2 | 7 | BDE2 | | Bad disparity errors status for Lane 2. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R |
| | | | | 0 | Error count < ETH[7:0] value. | | |
| | | | | 1 | Error count ≥ ETH[7:0] value. | | |
| | | 6 | NIT2 | | Not in table errors status for Lane 2. This control is paged by the LINK_PAGE control in Register 0x300. | 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. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R |
| | | | | 0 | Error count < ETH[7:0] value. | | |
| | | | | 1 | Error count ≥ ETH[7:0] value. | | |
| | | 4 | ILD2 | | Interlane deskew status for Lane 2. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R |
| | | | | 0 | Deskew failed. | | |
| | | | | 1 | Deskew achieved. | | |
| | | 3 | ILS2 | | Initial lane synchronization status for Lane 2. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R |
| | | | | 0 | Synchronization lost. | | |
| | | | | 1 | Synchronization achieved. | | |
| | | 2 | CKS2 | | Computed checksum status for Lane 2. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R |
| | | | | 0 | Checksum is incorrect. | | |
| | | | | 1 | Checksum is correct. | | |

## REGISTER DETAILS

*Table 55. Register Details*

| Addr. | Name | Bits | Bit Name | Settings | Description | Reset | Access |
|---|---|---|---|---|---|---|---|
| | | 1 | FS2 | | Frame synchronization status for Lane 2. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R |
| | | | | 0 | Synchronization lost. | | |
| | | | | 1 | Synchronization achieved. | | |
| | | 0 | CGS2 | | Code group synchronization status for Lane 2. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R |
| | | | | 0 | Synchronization lost. | | |
| | | | | 1 | Synchronization achieved. | | |
| 0x4B3 | LINK_STATUS3 | 7 | BDE3 | | Bad disparity errors status for Lane 3. This control is paged by the LINK_PAGE control in Register 0x300. | 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. This control is paged by the LINK_PAGE control in Register 0x300. | 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. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R |
| | | | | 0 | Error count < ETH[7:0] value. | | |
| | | | | 1 | Error count ≥ ETH[7:0] value. | | |
| | | 4 | ILD3 | | Interlane deskew status for Lane 3. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R |
| | | | | 0 | Deskew failed. | | |
| | | | | 1 | Deskew achieved. | | |
| | | 3 | ILS3 | | Initial lane synchronization status for Lane 3. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R |
| | | | | 0 | Synchronization lost. | | |
| | | | | 1 | Synchronization achieved. | | |
| | | 2 | CKS3 | | Computed checksum status for Lane 3. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R |
| | | | | 0 | Checksum is incorrect. | | |
| | | | | 1 | Checksum is correct. | | |
| | | 1 | FS3 | | Frame synchronization status for Lane 3. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R |
| | | | | 0 | Synchronization lost. | | |
| | | | | 1 | Synchronization achieved. | | |
| | | 0 | CGS3 | | Code group synchronization status for Lane 3. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R |
| | | | | 0 | Synchronization lost. | | |
| | | | | 1 | Synchronization achieved. | | |
| 0x4B4 | LINK_STATUS4 | 7 | BDE4 | | Bad disparity errors status for Lane 4. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R |
| | | | | 0 | Error count < ETH[7:0] value. | | |
| | | | | 1 | Error count ≥ ETH[7:0] value. | | |

## REGISTER DETAILS

*Table 55. Register Details*

| Addr. | Name | Bits | Bit Name | Settings | Description | Reset | Access |
|---|---|---|---|---|---|---|---|
| | | 6 | NIT4 | | Not in table errors status for Lane 4. This control is paged by the LINK_PAGE control in Register 0x300. | 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. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R |
| | | | | 0 | Error count < ETH[7:0] value. | | |
| | | | | 1 | Error count ≥ ETH[7:0] value. | | |
| | | 4 | ILD4 | | Interlane deskew status for Lane 4. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R |
| | | | | 0 | Deskew failed. | | |
| | | | | 1 | Deskew achieved. | | |
| | | 3 | ILS4 | | Initial lane synchronization status for Lane 4. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R |
| | | | | 0 | Synchronization lost. | | |
| | | | | 1 | Synchronization achieved. | | |
| | | 2 | CKS4 | | Computed checksum status for Lane 4. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R |
| | | | | 0 | Checksum is incorrect. | | |
| | | | | 1 | Checksum is correct. | | |
| | | 1 | FS4 | | Frame synchronization status for Lane 4. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R |
| | | | | 0 | Synchronization lost. | | |
| | | | | 1 | Synchronization achieved. | | |
| | | 0 | CGS4 | | Code group synchronization status for Lane 4. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R |
| | | | | 0 | Synchronization lost. | | |
| | | | | 1 | Synchronization achieved. | | |
| 0x4B5 | LINK_STATUS5 | 7 | BDE5 | | Bad disparity errors status for Lane 5. This control is paged by the LINK_PAGE control in Register 0x300. | 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. This control is paged by the LINK_PAGE control in Register 0x300. | 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. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R |
| | | | | 0 | Error count < ETH[7:0] value. | | |
| | | | | 1 | Error count ≥ ETH[7:0] value. | | |
| | | 4 | ILD5 | | Interlane deskew status for Lane 5. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R |
| | | | | 0 | Deskew failed. | | |
| | | | | 1 | Deskew achieved. | | |

## REGISTER DETAILS

*Table 55. Register Details*

| Addr. | Name | Bits | Bit Name | Settings | Description | Reset | Access |
|---|---|---|---|---|---|---|---|
| | | 3 | ILS5 | | Initial lane synchronization status for Lane 5. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R |
| | | | | 0 | Synchronization lost. | | |
| | | | | 1 | Synchronization achieved. | | |
| | | 2 | CKS5 | | Computed checksum status for Lane 5. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R |
| | | | | 0 | Checksum is incorrect. | | |
| | | | | 1 | Checksum is correct. | | |
| | | 1 | FS5 | | Frame synchronization status for Lane 5. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R |
| | | | | 0 | Synchronization lost. | | |
| | | | | 1 | Synchronization achieved. | | |
| | | 0 | CGS5 | | Code group synchronization status for Lane 5. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R |
| | | | | 0 | Synchronization lost. | | |
| | | | | 1 | Synchronization achieved. | | |
| 0x4B6 | LINK_STATUS6 | 7 | BDE6 | | Bad disparity errors status for Lane 6. This control is paged by the LINK_PAGE control in Register 0x300. | 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. This control is paged by the LINK_PAGE control in Register 0x300. | 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. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R |
| | | | | 0 | Error count < ETH[7:0] value. | | |
| | | | | 1 | Error count ≥ ETH[7:0] value. | | |
| | | 4 | ILD6 | | Interlane deskew status for Lane 6. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R |
| | | | | 0 | Deskew failed. | | |
| | | | | 1 | Deskew achieved. | | |
| | | 3 | ILS6 | | Initial lane synchronization status for Lane 6. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R |
| | | | | 0 | Synchronization lost. | | |
| | | | | 1 | Synchronization achieved. | | |
| | | 2 | CKS6 | | Computed checksum status for Lane 6. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R |
| | | | | 0 | Checksum is incorrect. | | |
| | | | | 1 | Checksum is correct. | | |
| | | 1 | FS6 | | Frame synchronization status for Lane 6. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R |
| | | | | 0 | Synchronization lost. | | |
| | | | | 1 | Synchronization achieved. | | |

## REGISTER DETAILS

*Table 55. Register Details*

| Addr. | Name | Bits | Bit Name | Settings | Description | Reset | Access |
|---|---|---|---|---|---|---|---|
| | | 0 | CGS6 | | Code group synchronization status for Lane 6. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R |
| | | | | 0 | Synchronization lost. | | |
| | | | | 1 | Synchronization achieved. | | |
| 0x4B7 | LINK_STATUS7 | 7 | BDE7 | | Bad disparity errors status for Lane 7. This control is paged by the LINK_PAGE control in Register 0x300. | 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. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R |
| | | | | 0 | Error count < ETH[7:0] value. | | |
| | | | | 1 | Error count ≥ ETH[7:0] value. | | |
| | | 5 | UEK7 | | Unexpected K character errors status Lane 7. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R |
| | | | | 0 | Error count < ETH[7:0] value. | | |
| | | | | 1 | Error count ≥ ETH[7:0] value. | | |
| | | 4 | ILD7 | | Interlane deskew status for Lane 7. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R |
| | | | | 0 | Deskew failed. | | |
| | | | | 1 | Deskew achieved. | | |
| | | 3 | ILS7 | | Initial lane synchronization status for Lane 7. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R |
| | | | | 0 | Synchronization lost. | | |
| | | | | 1 | Synchronization achieved. | | |
| | | 2 | CKS7 | | Computed checksum status for Lane 7. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R |
| | | | | 0 | Checksum is incorrect. | | |
| | | | | 1 | Checksum is correct. | | |
| | | 1 | FS7 | | Frame synchronization status for Lane 7. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R |
| | | | | 0 | Synchronization lost. | | |
| | | | | 1 | Synchronization achieved. | | |
| | | 0 | CGS7 | | Code group synchronization status for Lane 7. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R |
| | | | | 0 | Synchronization lost. | | |
| | | | | 1 | Synchronization achieved. | | |
| 0x4B8 | JESD_IRQ_ENABLEA | 7 | EN_BDE | | Bad disparity error counter. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R/W |
| | | 6 | EN_NIT | | Not in table error counter. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R/W |
| | | 5 | EN_UEK | | Unexpected K error counter. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R/W |
| | | 4 | EN_ILD | | Interlane deskew. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R/W |
| | | 3 | EN_ILS | | Initial lane synchronization. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R/W |

## REGISTER DETAILS

*Table 55. Register Details*

| Addr. | Name | Bits | Bit Name | Settings | Description | Reset | Access |
|---|---|---|---|---|---|---|---|
| | | 2 | EN_CKS | | Good checksum. This bit 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 only looks at data sent by the transmitter and not the checksum programmed into Register 0x45D. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R/W |
| | | 1 | EN_FS | | Frame synchronization. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R/W |
| | | 0 | EN_CGS | | Code group synchronization. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R/W |
| 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 obtained by the receiver. The first set of data is the ILAS data sent over the JESD204B link by the transmitter. The second set of data is the ILAS data programmed into the receiver via the SPI (Register 0x450 to Register 0x45D). If any of the data differs, the IRQ is triggered. Note that all of the ILAS data, including the checksum, is compared. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R/W |
| 0x4BA | JESD_IRQ_STATUSA | 7 | IRQ_BDE | | Bad disparity error counter. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R/W |
| | | 6 | IRQ_NIT | | Not in table error counter. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R/W |
| | | 5 | IRQ_UEK | | Unexpected K error counter. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R/W |
| | | 4 | IRQ_ILD | | Interlane deskew. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R/W |
| | | 3 | IRQ_ILS | | Initial lane synchronization. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R/W |
| | | 2 | IRQ_CKS | | Good checksum. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R/W |
| | | 1 | IRQ_FS | | Frame synchronization. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R/W |
| | | 0 | IRQ_CGS | | Code group synchronization. This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R/W |
| 0x4BB | JESD_IRQ_STATUSB | [7:1] | RESERVED | | Reserved. | 0x0 | R |
| | | 0 | IRQ_ILAS | | Configuration mismatch (checked for Lane 0 only). This control is paged by the LINK_PAGE control in Register 0x300. | 0x0 | R/W |
| 0x4BC | IRQ_OUTPUT_MUX_JESD | [7:1] | RESERVED | | Reserved. | 0x0 | R |
| | | 0 | MUX_JESD_IRQ | | Selects which IRQ pin is connected to the JESD204B $\overline{IRQx}$ sources. | 0x0 | R/W |
| | | | | 0 | Route the IRQ trigger signal to the $\overline{IRQ0}$ pin. | | |
| | | | | 1 | Route the IRQ trigger signal to the $\overline{IRQ1}$ pin. | | |
| 0x580 | BE_SOFT_OFF_GAIN_CTRL | 7 | BE_SOFT_OFF_GAIN_EN | | Must be 1 to use soft off/on. This control is paged by the MAINDAC_PAGE control in Register 0x008. | 0x0 | R/W |
| | | [6:3] | RESERVED | | Reserved. | 0x0 | R |

## REGISTER DETAILS

*Table 55. Register Details*

| Addr. | Name | Bits | Bit Name | Settings | Description | Reset | Access |
|---|---|---|---|---|---|---|---|
| | | [2:0] | BE_GAIN_RAMP_RATE | | Sets ramp rate. The gain ramps from 0 to 1 (or 1 to 0) in 32 steps over $2^{(CODE + 8)}$ DAC clock periods. This control is paged by the MAINDAC_PAGE control in Register 0x008. | 0x0 | R/W |
| 0x581 | BE_SOFT_OFF_ENABLE | 7 | ENA_SHORT_PAERR_SOFTOFF | | Enable short PA error soft off. This control is paged by the MAINDAC_PAGE control in Register 0x008. | 0x1 | R/W |
| | | 6 | ENA_LONG_PAERR_SOFTOFF | | Enable long PA error soft off. This control is paged by the MAINDAC_PAGE control in Register 0x008. | 0x1 | R/W |
| | | [5:4] | RESERVED | | Reserved. | 0x0 | R |
| | | 3 | ENA_JESD_ERR_SOFTOFF | | Enable JESD204B side error soft off. This control is paged by the MAINDAC_PAGE control in Register 0x008. | 0x0 | R/W |
| | | 2 | ROTATE_SOFT_OFF_EN | | When set to 1, the synchronization logic rotation triggers the DAC output soft off. Note that Register 0x03B, Bit 0 must also be high. This control is paged by the MAINDAC_PAGE control in Register 0x008. | 0x1 | R/W |
| | | 1 | TXEN_SOFT_OFF_EN | | When set to 1, a TXENx falling edge triggers the DAC output soft off. This control is paged by the MAINDAC_PAGE control in Register 0x008. | 0x1 | R/W |
| | | 0 | SPI_SOFT_OFF_EN | | Force a soft off when gain is 1. This control is paged by the MAINDAC_PAGE control in Register 0x008. | 0x0 | R/W |
| 0x582 | BE_SOFT_ON_ENABLE | 7 | SPI_SOFT_ON_EN | | Force a soft on when gain is 0. This control is paged by the MAINDAC_PAGE control in Register 0x008. | 0x0 | R/W |
| | | 6 | LONG_LEVEL_SOFTON_EN | | When set to 1, this bit enables the long level soft on. This control is paged by the MAINDAC_PAGE control in Register 0x008. | 0x1 | R/W |
| | | [5:0] | RESERVED | | Reserved. | 0x0 | R/W |
| 0x583 | LONG_PA_THRES_LSB | [7:0] | LONG_PA_THRESHOLD[7:0] | | Long average power threshold for comparison. This control is paged by the MAINDAC_PAGE control in Register 0x008. | 0x0 | R/W |
| 0x584 | LONG_PA_THRES_MSB | [7:5] | RESERVED | | Reserved. | 0x0 | R |
| | | [4:0] | LONG_PA_THRESHOLD[12:8] | | Long average power threshold for comparison. This control is paged by the MAINDAC_PAGE control in Register 0x008. | 0x0 | R/W |
| 0x585 | LONG_PA_CONTROL | 7 | LONG_PA_ENABLE | | Enable long average power calculation and error detection. This control is paged by the MAINDAC_PAGE control in Register 0x008. | 0x0 | R/W |
| | | [6:4] | RESERVED | | Reserved. | 0x0 | R |
| | | [3:0] | LONG_PA_AVG_TIME | | Sets length of long PA averaging time. The maximum value of this control is 10. This control is paged by the MAINDAC_PAGE control in Register 0x008. Averaging time = $2^{(9 + LONG_PA_AVG_TIME)}$ (PA clock periods). A PA clock period is calculated by the following:<br>If the main interpolation is >1× and the channel interpolation is >1×, PA clock period = 4 × main interpolation × DAC clock period.<br>If channel interpolation = 1× and main interpolation is >1×, PA clock period = 8 × main interpolation × DAC clock period.<br>Otherwise, PA clock period = 32 × DAC clock period. | 0x0 | R/W |
| 0x586 | LONG_PA_POWER_LSB | [7:0] | LONG_PA_POWER[7:0] | | Long average power readback. Power detected at data bus = I2 + Q2. The data bus calculation only | 0x0 | R |

## REGISTER DETAILS

*Table 55. Register Details*

| Addr. | Name | Bits | Bit Name | Settings | Description | Reset | Access |
|---|---|---|---|---|---|---|---|
| | | | | | uses the 6 MSBs of the I and Q data bus samples. This control is paged by the MAINDAC_PAGE control in Register 0x008. | | |
| 0x587 | LONG_PA_POWER_MSB | [7:5] | RESERVED | | Reserved. | 0x0 | R |
| | | [4:0] | LONG_PA_POWER[12:8] | | Long average power readback. Power detected at data bus = I2 + Q2. The data bus calculation only uses the 6 MSBs of the I and Q data bus samples. This control is paged by the MAINDAC_PAGE control in Register 0x008. | 0x0 | R |
| 0x588 | SHORT_PA_THRES_LSB | [7:0] | SHORT_PA_THRESHOLD[7:0] | | Short average power threshold for comparison. This control is paged by the MAINDAC_PAGE control in Register 0x008. | 0x0 | R/W |
| 0x589 | SHORT_PA_THRES_MSB | [7:5] | RESERVED | | Reserved. | 0x0 | R |
| | | [4:0] | SHORT_PA_THRESHOLD[12:8] | | Short average power threshold for comparison. This control is paged by the MAINDAC_PAGE control in Register 0x008. | 0x0 | R/W |
| 0x58A | SHORT_PA_CONTROL | 7 | SHORT_PA_ENABLE | | Enable short average power calculation and error detection. This control is paged by the MAINDAC_PAGE control in Register 0x008. | 0x0 | R/W |
| | | [6:2] | RESERVED | | Reserved. | 0x0 | R |
| | | [1:0] | SHORT_PA_AVG_TIME | | Sets length of short PA averaging. This control is paged by the MAINDAC_PAGE control in Register 0x008. Averaging time = $2^{SHORT_PA_AVG_TIME}$ (PA clock periods). A PA clock period is calculated by the following:<br>If the main interpolation is >1× and the channel interpolation is >1×, PA clock period = 4 × main interpolation × DAC clock period.<br>If channel interpolation = 1× and main interpolation is >1×, PA clock period = 8 × main interpolation × DAC clock period.<br>Otherwise, PA clock period = 32 × DAC clock period. | 0x0 | R/W |
| 0x58B | SHORT_PA_POWER_LSB | [7:0] | SHORT_PA_POWER[7:0] | | Short average power readback. Power detected at data bus = I2 + Q2. The data bus calculation only uses the 6 MSBs of the I and Q data bus samples. This control is paged by the MAINDAC_PAGE control in Register 0x008. | 0x0 | R |
| 0x58C | SHORT_PA_POWER_MSB | [7:5] | RESERVED | | Reserved. | 0x0 | R |
| | | [4:0] | SHORT_PA_POWER[12:8] | | Short average power readback. Power detected at data bus = I2 + Q2. The data bus calculation only uses the 6 MSBs of the I and Q data bus samples. This control is paged by the MAINDAC_PAGE control in Register 0x008. | 0x0 | R |
| 0x58D | TXEN_SM_0 | [7:1] | RESERVED | | Reserved. | 0x1 | R/W |
| | | 0 | ENA_TXENSM | | Enable TXEN state machine. This control is paged by the MAINDAC_PAGE control in Register 0x008. | 0x0 | R/W |
| 0x596 | BLANKING_CTRL | [7:4] | RESERVED | | Reserved. | 0x0 | R |
| | | 3 | SPI_TXEN | | If ENA_SPI_TXEN (Bit 2 of this register) = 1, the value of this register is the value of the TXENx status. This control is paged by the MAINDAC_PAGE control in Register 0x008. | 0x0 | R/W |

## REGISTER DETAILS

*Table 55. Register Details*

| Addr. | Name | Bits | Bit Name | Settings | Description | Reset | Access |
|---|---|---|---|---|---|---|---|
| | | 2 | ENA_SPI_TXEN | | Enable TXENx control via the SPI by setting this bit to 1. This control is paged by the MAINDAC_PAGE control in Register 0x008. | 0x0 | R/W |
| | | [1:0] | RESERVED | | Reserved. | 0x0 | R |
| 0x597 | JESD_PA_INT0 | [7:0] | JESD_PA_INT_ CNTRL[7:0] | | Each bit enables a JESD204B PA interrupt.<br>Bit 8 = CGS.<br>Bit 7 = frame sync.<br>Bit 6 = good check sum.<br>Bit 5 = initial lane sync.<br>Bit 4 = interlane deskew.<br>Bit 3 = bad disparity error counter.<br>Bit 2 = NIT error counter.<br>Bit 1= UEK error counter.<br>Bit 0 = lane FIFO overflow or underflow. | 0x0 | R/W |
| 0x598 | JESD_PA_INT1 | [7:1] | RESERVED | | Reserved. | 0x0 | R |
| | | 0 | JESD_PA_INT_ CNTRL[8] | | Each bit enables a JESD204B PA interrupt.<br>Bit 8 = CGS<br>Bit 7 = frame sync.<br>Bit 6 = good check sum.<br>Bit 5 = initial lane sync.<br>Bit 4 = interlane deskew.<br>Bit 3 = bad disparity error counter.<br>Bit 2 = NIT error counter.<br>Bit 1= UEK error counter.<br>Bit 0 = lane FIFO overflow or underflow. | 0x0 | R/W |
| 0x599 | TXEN_FLUSH_CTRL0 | [7:1] | RESERVED | | Reserved. | 0x0 | R |
| | | 0 | SPI_FLUSH_EN | | Enable datapath flush. This control is paged by the MAINDAC_PAGE control in Register 0x008. | 0x1 | R/W |
| 0x705 | NVM_LOADER_EN | [7:2] | RESERVED | | Reserved. | 0x0 | R |
| | | 1 | NVM_BLR_DONE | | NVM boot loader successfully completed when this bit reads back a value of 1. | 0x0 | R |
| | | 0 | NVM_BLR_EN | | Enable bootloader. This bit self clears when the boot loader completes or fails. | 0x0 | R/W |
| 0x790 | DACPLL_PDCTRL0 | 7 | PLL_PD5 | | PLL power-down control. Write this bit to 1 if bypassing the PLL. If using the PLL, keep this value at default (0). | 0x0 | R/W |
| | | [6:4] | PLL_PD4 | | PLL power-down control. Write this bit to 1 if bypassing the PLL. If using the PLL, keep this value at default (0). | 0x0 | R/W |
| | | 3 | PLL_PD3 | | PLL power-down control. Write this bit to 1 if bypassing the PLL. If using the PLL, keep this value default (0). | 0x0 | R/W |
| | | 2 | PLL_PD2 | | PLL power-down control. Write this bit to 1 if bypassing the PLL. If using the PLL, keep this value at default (0). | 0x0 | R/W |
| | | 1 | PLL_PD1 | | PLL power-down control. Write this bit to 1 if bypassing the PLL. If using the PLL, write this bit to 0. | 0x1 | R/W |
| | | 0 | PLL_PD0 | | PLL power-down control. Write this bit to 1 if bypassing the PLL. If using the PLL, keep this value at default (0). | 0x0 | R/W |

## REGISTER DETAILS

*Table 55. Register Details*

| Addr. | Name | Bits | Bit Name | Settings | Description | Reset | Access |
|---|---|---|---|---|---|---|---|
| 0x791 | DACPLL_PDCTRL1 | [7:5] | RESERVED | | Reserved. | 0x0 | R/W |
| | | 4 | PLL_PD10 | | PLL power-down control. Write this bit to 1 if bypassing the PLL. If using the PLL, keep this value at default (0). | 0x0 | R/W |
| | | 3 | PLL_PD9 | | PLL power-down control. Write this bit to 1 if bypassing the PLL. If using the PLL, keep this value at default (0). | 0x0 | R/W |
| | | 2 | PLL_PD8 | | PLL power-down control. Write this bit to 1 if bypassing the PLL. If using the PLL, keep this value at default (0). | 0x0 | R/W |
| | | 1 | PLL_PD7 | | PLL power-down control. Write this bit to 1 if bypassing the PLL. If using the PLL, keep this value at default (0). | 0x0 | R/W |
| | | 0 | PLL_PD6 | | PLL power-down control. Write this bit to 1 if bypassing the PLL. If using the PLL, keep this value at default (0). | 0x0 | R/W |
| 0x792 | DACPLL_CTRL0 | [7:2] | RESERVED | | Reserved. | 0x0 | R |
| | | 1 | D_CAL_RESET | | Resets VCO calibration. | 0x1 | R/W |
| | | 0 | D_RESET_VCO_DIV | | Setting this high holds the VCO output divider in reset. This has the effect of turning off the input (and output) of the ADC clock driver. | 0x0 | R/W |
| 0x793 | DACPLL_CTRL1 | [7:2] | RESERVED | | Reserved. | 0x0 | R |
| | | [1:0] | M_DIVIDER−1 | | Programmable predivider value for PFD (in n − 1 notation). M_DIVIDER = PLL reference clock/PFD frequency. For optimal spectral performance, choose an M divider setting that selects a high PFD frequency within the allowable PFD range. For 9.96 GHz ≤ VCO frequency ≤ 10.87 GHz, 25 MHz ≤ PFD frequency ≤ 225 MHz. For 9.96 GHz > VCO frequency > 10.87 GHz, 25 MHz ≤ PFD frequency ≤ 770 MHz. | 0x0 | R/W |
| | | | | 0 | Divide by 1. | | |
| | | | | 1 | Divide by 2. | | |
| | | | | 10 | Divide by 3. | | |
| | | | | 11 | Divide by 4. | | |
| 0x794 | DACPLL_CTRL2 | [7:6] | RESERVED | | Reserved. | 0x0 | R/W |
| | | [5:0] | DACPLL_CP | | Charge pump current control. Charge pump current = 100 µA + code × 100 µA. | 0x4 | R/W |
| 0x795 | DACPLL_CTRL3 | [7:4] | RESERVED | | Reserved. | 0x0 | R/W |
| | | [3:0] | D_CP_CALBITS | | DAC PLL optimization control. | 0x8 | R/W |
| 0x796 | DACPLL_CTRL4 | [7:4] | PLL_CTRL0 | | DAC PLL optimization control. | 0xD | R/W |
| | | [3:0] | RESERVED | | Reserved. | 0x2 | R/W |
| 0x797 | DACPLL_CTRL5 | [7:6] | RESERVED | | Reserved. | 0x0 | R/W |
| | | [5:0] | PLL_CTRL1 | | DAC PLL optimization control. | 0x20 | R/W |
| 0x798 | DACPLL_CTRL6 | 7 | RESERVED | | Reserved. | 0x0 | R |
| | | 6 | PLL_CTRL3 | | DAC PLL optimization control. | 0x0 | R/W |
| | | [5:0] | PLL_CTRL2 | | DAC PLL optimization control. | 0x1C | R/W |
| 0x799 | DACPLL_CTRL7 | [7:6] | ADC_CLK_DIVIDER | | ADC clock output divider. | 0x0 | R/W |
| | | | | 0 | Divide by 1. | | |
| | | | | 1 | Divide by 2. | | |
| | | | | 10 | Divide by 3. | | |

## REGISTER DETAILS

*Table 55. Register Details*

| Addr. | Name | Bits | Bit Name | Settings | Description | Reset | Access |
|---|---|---|---|---|---|---|---|
| | | | | 11 | Divide by 4. | | |
| | | [5:0] | N_DIVIDER | | Programmable divide by N value from 2 to 50. N_DIVIDER = (VCO frequency × M_DIVIDER)/(8 × reference clock frequency). | 0x8 | R/W |
| 0x7A0 | DACPLL_CTRL9 | [7:6] | RESERVED | | Reserved. | 0x2 | R/W |
| | | 5 | D_EN_VAR_FINE_PRE | | DAC PLL control. | 0x0 | R/W |
| | | [4:3] | RESERVED | | Reserved. | 0x2 | R/W |
| | | 2 | D_EN_VAR_COARSE_PRE | | DAC PLL control. | 0x0 | R/W |
| | | [1:0] | RESERVED | | Reserved. | 0x0 | R/W |
| 0x7A2 | DACPLL_CTRL10 | 7 | RESERVED | | Reserved. | 0x0 | R |
| | | [6:5] | D_REGULATOR_CAL_WAIT | | DAC PLL optimization control. | 0x1 | R/W |
| | | [4:3] | D_VCO_CAL_WAIT | | DAC PLL optimization control. | 0x2 | R/W |
| | | [2:1] | D_VCO_CAL_CYCLES | | DAC PLL optimization control. | 0x2 | R/W |
| | | 0 | RESERVED | | Reserved. | 0x1 | R/W |
| 0x7B5 | PLL_STATUS | [7:1] | RESERVED | | Reserved. | 0x0 | R |
| | | 0 | PLL_LOCK | | DAC PLL lock status. | 0x0 | R |

## OUTLINE DIMENSIONS

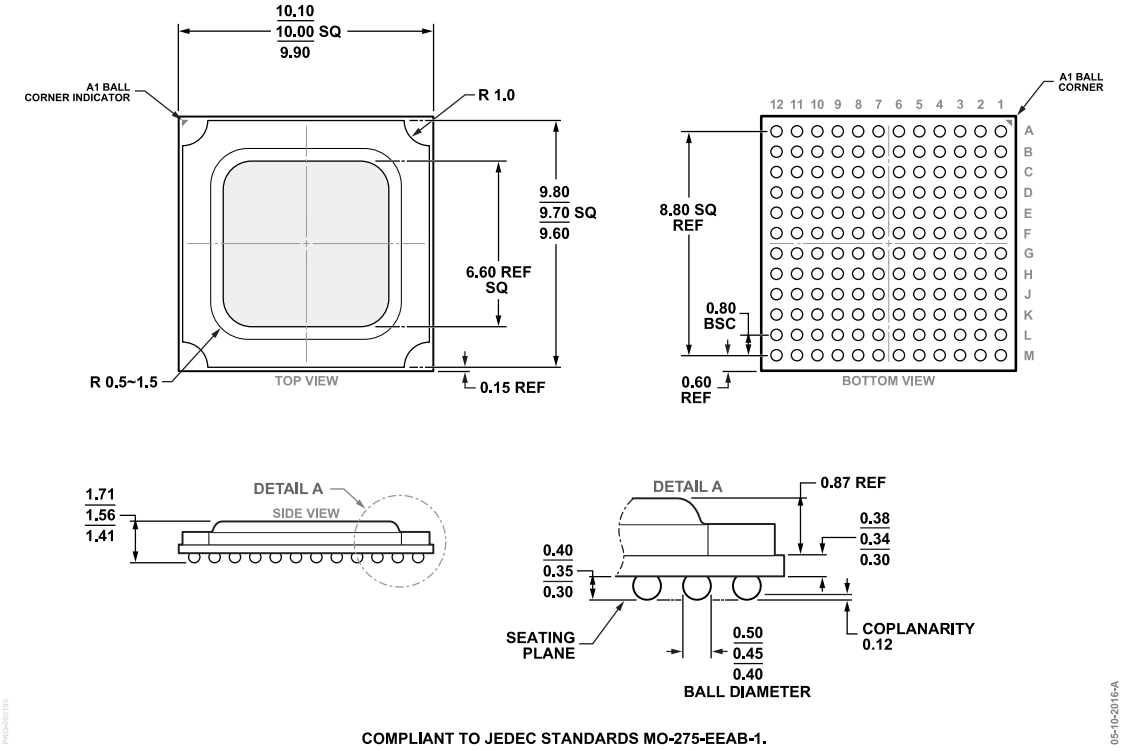


*Figure 66. 144-Ball Ball Grid Array, Thermally Enhanced [BGA_ED]*
*(BP-144-1)*
*Dimensions shown in millimeters*

Updated: May 10, 2022

## ORDERING GUIDE

| Model[1] | Temperature Range | Package Description | Packing Quantity | Package Option |
|---|---|---|---|---|
| AD9171BBPZ | -40°C to +85°C | 144 ball (10x10x1.71 w/6.6 mm EP) | Tray | BP-144-1 |
| AD9171BBPZRL | -40°C to +85°C | 144 ball (10x10x1.71 w/6.6 mm EP) | Reel, 1500 | BP-144-1 |

[1] Z = RoHS Compliant Part.

## EVALUATION BOARDS

| Model[1] | Description |
|---|---|
| AD9171-FMC-EBZ | Evaluation Board |

[1] Z = RoHS Compliant Part.

©2018-2022 Analog Devices, Inc. All rights reserved. Trademarks and registered trademarks are the property of their respective owners.
One Analog Way, Wilmington, MA 01887-2356, U.S.A.

## Looking for pricing, stock, or lifecycle information?

Click below to explore more details on WIN SOURCE:

- View AD9171BBPZ on WIN SOURCE
- Analog Devices Inc. Information

## Optimize Your Supply Chain with WIN SOURCE Solutions

- Global Sourcing Solution
- Obsolete Management
- Cost Control Management
- Shortage Management
- Alternative Solution
- Excess Inventory Management