# Dual, 16-Bit, 2.25 GSPS, TxDAC+ Digital-to-Analog Converter

Data Sheet **AD9152**

## FEATURES

- Supports input data rates up to 1.125 GSPS
- Proprietary low spurious and distortion design
  - Single carrier LTE 20 MHz bandwidth (BW), ACLR = 77 dBc at 180 MHz IF
  - SFDR = 72 dBc at 150 MHz IF, −6 dBFS
- Flexible 4-lane JESD204B interface
- Multiple chip synchronization
  - Fixed latency
  - Data generator latency compensation
- Selectable 1×, 2×, 4×, and 8× interpolation filter
  - Low power architecture
- Input signal power detection
  - Emergency stop for downstream analog circuitry protection
- Transmit enable function allows extra power saving
- High performance, low noise, phase-locked loop (PLL) clock multiplier
- Digital inverse sinc filter and programmable finite impulse response (FIR) filter
- Low power: 1223 mW at 1.5 GSPS, 1406 mW at 2.0 GSPS, full operating conditions
- 56-lead LFCSP with exposed pad

## APPLICATIONS

- Wireless communications
  - Multicarrier LTE and GSM base stations
  - Wideband repeaters
  - Software defined radios
- Wideband communications
  - Point to point microwave radios
  - LMDS/MMDS
- Transmit diversity, multiple input/multiple output (MIMO)
- Instrumentation
- Automated test equipment

## GENERAL DESCRIPTION

The AD9152 is a dual, 16-bit, high dynamic range digital-to-analog converter (DAC) that provides a maximum sample rate of 2.25 GSPS, permitting a multicarrier generation up to the Nyquist frequency. The DAC outputs are optimized to interface seamlessly with the ADRF6720 analog quadrature modulator (AQM) from Analog Devices, Inc. An optional 3-wire or 4-wire serial port interface (SPI) provides for programming/readback of many internal parameters. The full-scale output current can be programmed over a range of 4 mA to 20 mA. The AD9152 is available in a 56-lead LFCSP. The AD9152 is a member of the TxDAC+® family.

## PRODUCT HIGHLIGHTS

1. Ultrawide signal bandwidth enables emerging wideband and multiband wireless applications.
2. Advanced low spurious and distortion design techniques provide high quality synthesis of wideband signals from baseband to high intermediate frequencies.
3. JESD204B Subclass 1 support simplifies multichip synchronization in software and hardware design.
4. Fewer pins for data interface width with the serializer/deserializer (SERDES) JESD204B four-lane interface.
5. Programmable transmit enable function allows easy design balance between power consumption and wake-up time.
6. Small package size with an 8 mm × 8 mm footprint.

## FUNCTIONAL BLOCK DIAGRAM

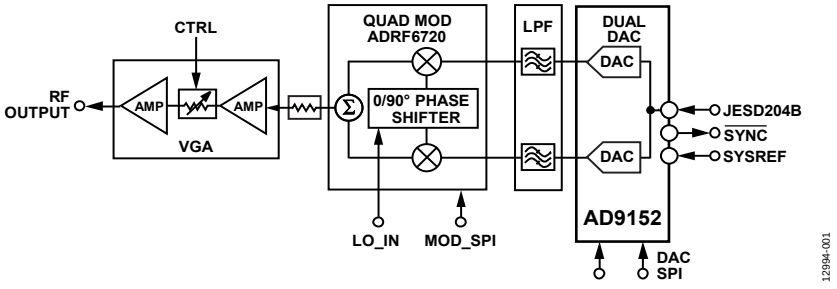


*Figure 1.*




One Technology Way, P.O. Box 9106, Norwood, MA 02062-9106, U.S.A.
Tel: 781.329.4700 
Technical Support www.analog.com

# TABLE OF CONTENTS

## REVISION HISTORY

**8/2017—Rev. A to Rev. B**

Change to Output Compliance Range Parameter; Table 1 ..... 5
Updated Outline Dimensions ..... 103
Changes to Ordering Guide ..... 103

**2/2017—Rev. 0 to Rev. A**

Changes to Figure 2 ..... 4
Change to Device Revision Parameter, Table 14 ..... 23
Changes to Step 1: Start Up the DAC Section and Table 16 ..... 24
Changes to Digital Datapath Section, Address 0x14, Table 17, and Table 18 ..... 26
Change to JESD204B Setup Section ..... 27
Changes to SERDES PLL Fixed Register Writes Section ..... 33
Added Table 36; Renumbered Sequentially ..... 33
Change to Register Block 0x47, Bit 4, Table 69 ..... 63
Deleted DAC PLL Fixed Register Writes Section, Figure 70, and Figure 71; Renumbered Sequentially ..... 64
Changes to Temperature Tracking Section, Starting the PLL Section, and Table 73 ..... 65
Changes to Table 76 and Table 77 ..... 69
Changes to Table 78 ..... 70
Changes to Table 85 ..... 74
Changes to Table 86 ..... 80

**4/2015—Revision 0: Initial Version**

# DETAILED FUNCTIONAL BLOCK DIAGRAM

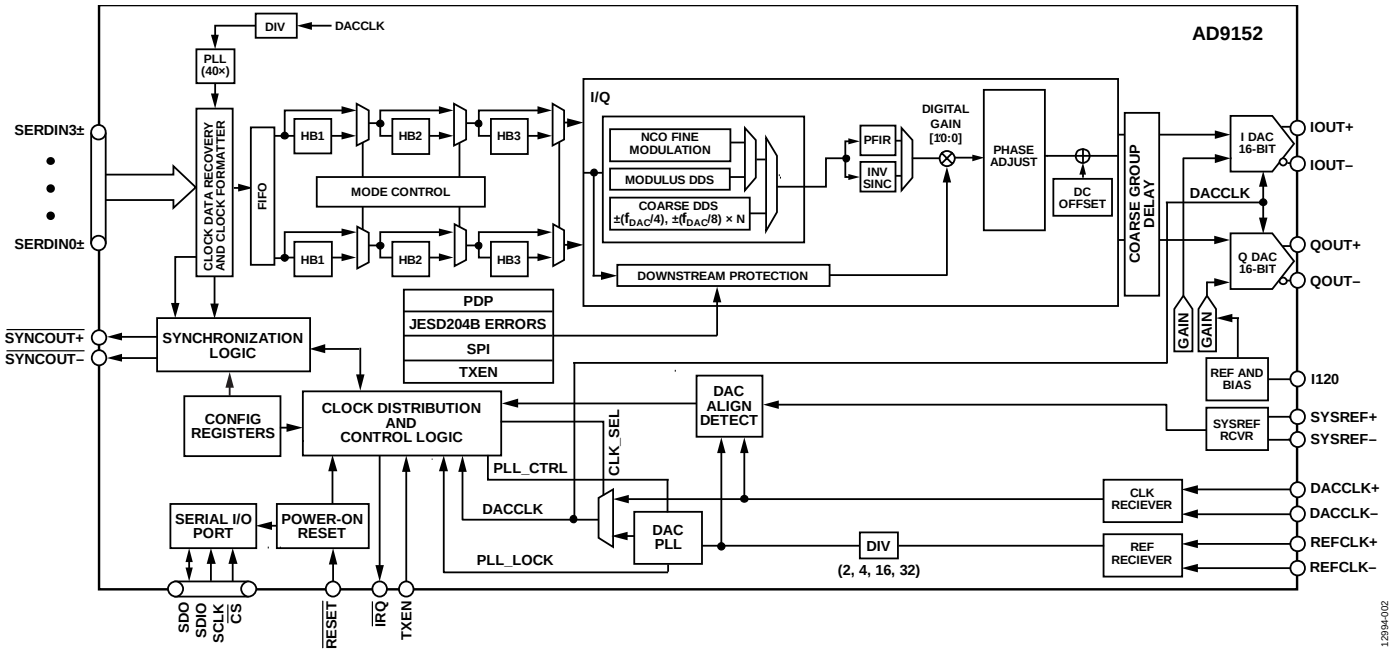


*Figure 2. Detailed Functional Block Diagram*

# SPECIFICATIONS

## DC SPECIFICATIONS

AVDD33 = 3.3 V, SIOVDD33 = 3.3 V, IOVDD = 1.8 V, DVDD12 = 1.2 V, CVDD12 = 1.2 V, PVDD12 = 1.2 V, PLLVDD12 = 1.2 V, SVDD12 = 1.2 V, SDVDD12 = 1.2 V, $V_{TT}$ = 1.2 V, $T_A$ = −40°C to +85°C, $I_{OUTFS}$ = 20 mA, unless otherwise noted.

**Table 1.**

| Parameter | Test Conditions/Comments | Min | Typ | Max | Unit |
|---|---|---|---|---|---|
| RESOLUTION | | | 16 | | Bits |
| ACCURACY | | | | | |
| Differential Nonlinearity (DNL) | | | ±5.0 | | LSB |
| Integral Nonlinearity (INL) | | | ±10.0 | | LSB |
| MAIN DAC OUTPUTS | | | | | |
| Gain Error | With internal reference | −5.5 | −1.3 | +5.5 | % FSR |
| I/Q Gain Mismatch | | −4.5 | | +4.5 | % FSR |
| Full-Scale Output Current ($I_{OUTFS}$) | Based on a 4 kΩ external resistor between I120 and ground | | | | |
| Maximum Setting | | 19.1 | 20.22 | 21.4 | mA |
| Minimum Setting | | 3.8 | 4.04 | 4.3 | mA |
| Output Compliance Range | | 2.3 | | 3.47 | V |
| Output Resistance | | | 15 | | MΩ |
| Output Capacitance | | | 3.0 | | pF |
| Gain DAC Monotonicity | | | Guaranteed | | |
| MAIN DAC TEMPERATURE DRIFT | | | | | |
| Offset | | | 0.1 | | ppm/°C |
| Gain | | | 35 | | ppm/°C |
| Reference Voltage | | | 25 | | ppm/°C |
| REFERENCE | | | | | |
| Internal Reference Voltage | | | 0.5 | | V |
| ANALOG SUPPLY VOLTAGES | | | | | |
| AVDD33 | ±5% | 3.13 | 3.3 | 3.47 | V |
| PVDD12, CVDD12 | ±5% | 1.14 | 1.2 | 1.26 | V |
| | ±2% | 1.274 | 1.3 | 1.326 | V |
| SVDD12, PLLVDD12, $V_{TT}$ | ±5% | 1.14 | 1.2 | 1.26 | V |
| | ±2% | 1.274 | 1.3 | 1.326 | V |
| DIGITAL SUPPLY VOLTAGES | | | | | |
| DVDD12, SDVDD12 | ±5% | 1.14 | 1.2 | 1.26 | V |
| | ±2% | 1.274 | 1.3 | 1.326 | V |
| SIOVDD33 | ±5% | 3.13 | 3.3 | 3.47 | V |
| IOVDD | ±5% | 1.71 | 1.8 | 3.47 | V |
| POWER CONSUMPTION | | | | | |
| Total Power | 2× interpolation mode, $f_{DAC}$ = 1.5 GSPS, IF = 70 MHz, PLL off, INVSINC on, digital gain on, NCO on, JESD204B Mode 4, four SERDES lanes with 7.5 Gbps lane rate, $I_{OUTFS}$ = 20 mA | | 1223 | | mW |
| AVDD33 | | | 87 | | mA |
| PVDD12 | | | 11 | | mA |
| CVDD12 | | | 179 | | mA |
| SDVDD12 and SVDD12 (Includes PLLVDD12 and $V_{TT}$) | | | 328 | | mA |
| DVDD12 | | | 246 | | mA |
| SIOVDD33 and IOVDD | | | 5.7 | | mA |
| OPERATING TEMPERATURE RANGE | | −40 | +25 | +85 | °C |

## DIGITAL SPECIFICATIONS

AVDD33 = 3.3 V, SIOVDD33 = 3.3 V, IOVDD = 1.8 V, DVDD12 = 1.2 V, CVDD12 = 1.2 V, PVDD12 = 1.2 V, PLLVDD12 = 1.2 V, SVDD12 = 1.2 V, SDVDD12 = 1.2 V, $V_{TT}$ = 1.2 V, $T_A$ = −40°C to +85°C, $I_{OUTFS}$ = 20 mA, unless otherwise noted.

**Table 2.**

| Parameter | Symbol | Test Conditions/Comments | Min | Typ | Max | Unit |
|---|---|---|---|---|---|---|
| CMOS INPUT LOGIC LEVEL | | | | | | |
| Input Voltage Logic | $V_{IN}$ | | | | | |
| High | | 1.8 V ≤ IOVDD ≤ 3.3 V | 0.7 × IOVDD | | | V |
| Low | | 1.8 V ≤ IOVDD ≤ 3.3 V | | | 0.3 × IOVDD | V |
| CMOS OUTPUT LOGIC LEVEL | | | | | | |
| Output Voltage Logic | $V_{OUT}$ | | | | | |
| High | | 1.8 V ≤ IOVDD ≤ 3.3 V | 0.7 × IOVDD | | | V |
| Low | | 1.8 V ≤ IOVDD ≤ 3.3 V | | | 0.3 × IOVDD | V |
| MAXIMUM DAC UPDATE RATE[1] | | DVDD12 = CVDD12 = PVDD12 = 1.3 V ± 2% | | | | |
| | | 1× interpolation[2] | 1238 | | | MSPS |
| | | 2× interpolation | 2250 | | | MSPS |
| | | 4× interpolation | 2250 | | | MSPS |
| | | 8× interpolation | 2250 | | | MSPS |
| ADJUSTED DAC UPDATE RATE | | DVDD12 = CVDD12 = PVDD12 = 1.3 V ± 2% | | | | |
| | | 1× interpolation | 1238 | | | MSPS |
| | | 2× interpolation | 1125 | | | MSPS |
| | | 4× interpolation | 562.5 | | | MSPS |
| | | 8× interpolation | 281.25 | | | MSPS |
| INTERFACE[3] | | | | | | |
| Number of JESD204B Lanes | | | | 4 | | Lanes |
| JESD204B Serial Interface Speed | | SVDD12 = SDVDD12 = PLLVDD12 = 1.3 V ± 2% | | | | |
| Minimum | | Per lane | | | 1.44 | Gbps |
| Maximum | | Per lane | 12.38 | | | Gbps |
| DAC CLOCK INPUT (DACCLK±) | | | | | | |
| Differential Peak-to-Peak Voltage | | | 400 | 1000 | 2000 | mV |
| Common-Mode Voltage | | Self biased input, ac-coupled | | 600 | | mV |
| Maximum Clock Rate | | DVDD12 = CVDD12 = PVDD12 = 1.3 V ± 2% | 2250 | | | MHz |
| REFERENCE CLOCK INPUT (REFCLK±) | | | | | | |
| Differential Peak-to-Peak Voltage | | | 400 | 1000 | 2000 | mV |
| Common-Mode Voltage | | Self biased input, ac-coupled | | 600 | | mV |
| Input Clock Frequency (PLL Mode) | | 6 GHz ≤ $f_{VCO}$ ≤ 12 GHz | 70 | | 1000 | MHz |
| SYSTEM REFERENCE INPUT (SYSREF±) | | | | | | |
| Differential Peak-to-Peak Voltage | | | 400 | 1000 | 2000 | mV |
| Common-Mode Voltage | | | 0 | | 2000 | mV |
| SYSREF± Frequency[4] | | | | | $f_{DATA}/(K \times S)$ | Hz |
| SYSREF± TO DAC CLOCK[5] | | SYSREF± differential swing = 1.2 V, slew rate = 6.3 V/ns, hysteresis off (ac-coupled, and 0 V, 0.6 V, 1.25 V, 2.0 V dc-coupled common-mode voltages) | | | | |
| Setup Time | $t_{SSD}$ | | −6 | | | ps |
| Hold Time | $t_{HSD}$ | | 224 | | | ps |
| Keep Out Window | KOW | | | 218 | | ps |

| Parameter | Symbol | Test Conditions/Comments | Min | Typ | Max | Unit |
|---|---|---|---|---|---|---|
| SPI | | | | | | |
| Maximum Clock Rate | SCLK | IOVDD = 1.8 V | 10 | | | MHz |
| Minimum SCLK Pulse Width | | | | | | |
| High | $t_{PWH}$ | | | | 8 | ns |
| Low | $t_{PWL}$ | | | | 12 | ns |
| SDIO to SCLK | | | | | | |
| Setup Time | $t_{DS}$ | | 5 | | | ns |
| Hold Time | $t_{DH}$ | | 2 | | | ns |
| SDO to SCLK | | | | | | |
| Data Valid Window | $t_{DV}$ | | 26 | | | ns |
| $\overline{CS}$ to SCLK | | | | | | |
| Setup Time | $t_{S\overline{CS}}$ | | 5 | | | ns |
| Hold Time | $t_{H\overline{CS}}$ | | 2 | | | ns |

[1] See Table 3 and Table 4 for detailed specifications for the DAC update rate conditions.
[2] The maximum speed for 1× interpolation is limited by the JESD2040B interface. See Table 4 for details.
[3] See Table 4 for detailed specifications for JESD2040B speed conditions.
[4] K and S are JESD204B transport layer parameters. See Table 41 for the full definitions.
[5] See Table 5 for detailed specifications for SYSREF± to DAC clock timing conditions.

## MAXIMUM DAC UPDATE RATE SPEED SPECIFICATIONS BY SUPPLY

AVDD33 = 3.3 V, SIOVDD33 = 3.3 V, IOVDD = 1.8 V, DVDD12 = 1.2 V, CVDD12 = 1.2 V, PVDD12 = 1.2 V, PLLVDD12 = 1.2 V, SVDD12 = 1.2 V, SDVDD12 = 1.2 V, $V_{TT}$ = 1.2 V, $T_A$ = −40°C to +85°C, $I_{OUTFS}$ = 20 mA, unless otherwise noted.

**Table 3.**

| Parameter | Test Conditions/Comments | Min | Typ | Max | Unit |
|---|---|---|---|---|---|
| MAXIMUM DAC UPDATE RATE | DVDD12, CVDD12 = 1.2 V ± 5% | 1.85 | | | GSPS |
| | DVDD12, CVDD12 = 1.3 V ± 2% | 2.25 | | | GSPS |

## JESD204B SERIAL INTERFACE SPEED SPECIFICATIONS

AVDD33 = 3.3 V, SIOVDD33 = 3.3 V, IOVDD = 1.8 V, DVDD12 = 1.2 V, CVDD12 = 1.2 V, PVDD12 = 1.2 V, PLLVDD12 = 1.2 V, SVDD12 = 1.2 V, SDVDD12 = 1.2 V, $V_{TT}$ = 1.2 V, $T_A$ = −40°C to +85°C, $I_{OUTFS}$ = 20 mA, unless otherwise noted.

**Table 4.**

| Parameter | Test Conditions/Comments | Min | Typ | Max | Unit |
|---|---|---|---|---|---|
| HALF RATE | SVDD12, SDVDD12, PLLVDD12 = 1.2 V ± 5% | 5.75 | | 11.00 | Gbps |
| | SVDD12, SDVDD12, PLLVDD12 = 1.3 V ± 2% | 5.75 | | 12.38 | Gbps |
| FULL RATE | SVDD12, SDVDD12, PLLVDD12 = 1.2 V ± 5% | 2.88 | | 5.53 | Gbps |
| | SVDD12, SDVDD12, PLLVDD12 = 1.3 V ± 2% | 2.88 | | 6.19 | Gbps |
| OVERSAMPLING | SVDD12, SDVDD12, PLLVDD12 = 1.2 V ± 5% | 1.44 | | 2.69 | Gbps |
| | SVDD12, SDVDD12, PLLVDD12 = 1.3 V ± 2% | 1.44 | | 3.09 | Gbps |

## SYSREF± TO DAC CLOCK TIMING SPECIFICATIONS

AVDD33 = 3.3 V, SIOVDD33 = 3.3 V, IOVDD = 1.8 V, DVDD12 = 1.2 V, CVDD12 = 1.2 V, PVDD12 = 1.2 V, PLLVDD12 = 1.2 V, SVDD12 = 1.2 V, SDVDD12 = 1.2 V, $V_{TT}$ = 1.2 V, $T_A$ = −40°C to +85°C, $I_{OUTFS}$ = 20 mA, SYSREF± common-mode voltages = 0.0 V, 0.6 V, 1.25 V, and 2.0 V, unless otherwise noted.

**Table 5.**

| Parameter | Test Conditions/Comments | Min | Typ | Max | Unit |
|---|---|---|---|---|---|
| SYSREF | Differential swing = 1.2 V, slew rate = 6.3 V/ns | | | | |
| Hysteresis Off | | | | | |
| Setup Time | AC-coupled | −9 | | | ps |
| | DC-coupled | −6 | | | ps |
| Hold Time | AC-coupled | 199 | | | ps |
| | DC-coupled | 224 | | | ps |
| Hysteresis On (HYS_CNTRL = 0x3FF) | | | | | |
| Setup Time | AC-coupled | 143 | | | ps |
| | DC-coupled | 145 | | | ps |
| Hold Time | AC-coupled | 97 | | | ps |
| | DC-coupled | 123 | | | ps |

## DIGITAL INPUT DATA TIMING SPECIFICATIONS

AVDD33 = 3.3 V, SIOVDD33 = 3.3 V, IOVDD = 1.8 V, DVDD12 = 1.2 V, CVDD12 = 1.2 V, PVDD12 = 1.2 V, PLLVDD12 = 1.2 V, SVDD12 = 1.2 V, SDVDD12 = 1.2 V, $V_{TT}$ = 1.2 V, $T_A$ = −40°C to +85°C, $I_{OUTFS}$ = 20 mA, unless otherwise noted.

**Table 6.**

| Parameter | Test Conditions/Comments | Min | Typ | Max | Unit |
|---|---|---|---|---|---|
| LATENCY | | | | | |
| Interface | | | 17 | | PClock[1] cycles |
| Interpolation | | | | | |
| 1× | | | 143 | | DAC clock cycles |
| 2× | | | 163 | | DAC clock cycles |
| 4× | | | 287 | | DAC clock cycles |
| 8× | | | 557 | | DAC clock cycles |
| Inverse Sinc | | | 17 | | DAC clock cycles |
| Fine Modulation | | | 20 | | DAC clock cycles |
| Coarse Modulation | | | | | |
| $f_S/8$ | | | 8 | | DAC clock cycles |
| $f_S/4$ | | | 4 | | DAC clock cycles |
| Digital Phase Adjust | | | 12 | | DAC clock cycles |
| Digital Gain Adjust | | | 12 | | DAC clock cycles |
| Power-Up Time | Register 0x011 from 0x60 to 0x00 | | 60 | | µs |

[1] PClock is the AD9152 internal processing clock and equals the lane rate ÷ 40.

## LATENCY VARIATION SPECIFICATIONS

AVDD33 = 3.3 V, SIOVDD33 = 3.3 V, IOVDD = 1.8 V, DVDD12 = 1.2 V, CVDD12 = 1.2 V, PVDD12 = 1.2 V, PLLVDD12 = 1.2 V, SVDD12 = 1.2 V, SDVDD12 = 1.2 V, $V_{TT}$ = 1.2 V, $T_A$ = −40°C to +85°C, $I_{OUTFS}$ = 20 mA, unless otherwise noted.

**Table 7.**

| Parameter | Test Conditions/Comments | Min | Typ | Max | Unit |
|---|---|---|---|---|---|
| DAC LATENCY VARIATION | Given proper calibration of the local multiframe clock (LMFC) delay | | | | |
| Subclass 1 | | | | | DAC clock cycles |
| PLL Off | | −3 | 0 | +3 | DAC clock cycles |
| PLL On | | −4 | | +4 | DAC clock cycles |

## JESD204B INTERFACE ELECTRICAL SPECIFICATIONS

AVDD33 = 3.3 V, SIOVDD33 = 3.3 V, IOVDD = 1.8 V, DVDD12 = 1.2 V, CVDD12 = 1.2 V, PVDD12 = 1.2 V, PLLVDD12 = 1.2 V, SVDD12 = 1.2 V, SDVDD12 = 1.2 V, $V_{TT}$ = 1.2 V, $T_A$ = −40°C to +85°C, $I_{OUTFS}$ = 20 mA, unless otherwise noted.

**Table 8.**

| Parameter | Symbol | Test Conditions/Comments | Min | Typ | Max | Unit |
|---|---|---|---|---|---|---|
| JESD204B DATA INPUTS | | | | | | |
| Input Leakage Current | | $T_A$ = 25°C | | | | |
| Logic High | | Input level = 1.2 V ± 0.25 V, $V_{TT}$ = 1.2 V | | 10 | | µA |
| Logic Low | | Input level = 0 V | | −4 | | µA |
| Unit Interval | UI | | 81 | | 694 | ps |
| Common-Mode Voltage | $V_{RCM}$ | AC-coupled<br>$V_{TT}$ = SVDD12[1] | −0.05 | | +1.85 | V |
| Differential Voltage | $R\_V_{DIFF}$ | | 110 | | 1050 | mV |
| $V_{TT}$ Source Impedance | $Z_{TT}$ | At dc | | | 30 | Ω |
| Differential Impedance | $Z_{RDIFF}$ | At dc | 80 | 100 | 120 | Ω |
| Differential Return Loss | $RL_{RDIF}$ | | | 8 | | dB |
| Common-Mode Return Loss | $RL_{RCM}$ | | | 6 | | dB |
| DIFFERENTIAL OUTPUTS ($\overline{\text{SYNCOUT}}$±)[2] | | | | | | |
| Output Offset Voltage | $V_{OS}$ | | 1.15 | | 1.25 | V |
| Output Differential Voltage | $V_{OD}$ | High swing mode: Register 0x230, Bit 0 = 1 | 350 | | 410 | mV |
| DETERMINISTIC LATENCY | | | | | | |
| Fixed | | | | | 17 | PClock[3] cycles |
| Variable | | | | | 2 | PClock[3] cycles |
| SYSREF± TO LMFC DELAY | | | | 4 | | DAC clock cycles |

[1] As measured on the input side of the ac coupling capacitor.
[2] IEEE Standard 1596.3 LVDS compatible.
[3] PClock is the AD9152 internal processing clock and equals the lane rate ÷ 40.

## AC SPECIFICATIONS

AVDD33 = 3.3 V, SIOVDD33 = 3.3 V, IOVDD = 1.8 V, DVDD12 = 1.2 V, CVDD12 = 1.2 V, PVDD12 = 1.2 V, PLLVDD12 = 1.2 V, SVDD12 = 1.2 V, SDVDD12 = 1.2 V, $V_{TT}$ = 1.2 V, $T_A$ = −40°C to +85°C, $I_{OUTFS}$ = 20 mA, unless otherwise noted.

**Table 9.**

| Parameter | Test Conditions/Comments | Min | Typ | Max | Unit |
|---|---|---|---|---|---|
| SPURIOUS-FREE DYNAMIC RANGE (SFDR) | −6 dBFS single tone | | | | |
| $f_{DAC}$ = 1966.08 MSPS | $f_{OUT}$ = 20 MHz | | 76 | | dBc |
| | $f_{OUT}$ = 150 MHz | | 72 | | dBc |
| | $f_{OUT}$ = 180 MHz | | 68 | | dBc |
| TWO-TONE THIRD INTERMODULATION DISTORTION (IMD) | −6 dBFS | | | | |
| $f_{DAC}$ = 983.04 MSPS | $f_{OUT}$ = 30 MHz | | 86 | | dBc |
| | $f_{OUT}$ = 150 MHz | | 79 | | dBc |
| $f_{DAC}$ = 1966.08 MSPS | $f_{OUT}$ = 30 MHz | | 86 | | dBc |
| | $f_{OUT}$ = 180 MHz | | 78 | | dBc |
| NOISE SPECTRAL DENSITY (NSD), SINGLE TONE | 0 dBFS | | | | |
| $f_{DAC}$ = 983.04 MSPS | $f_{OUT}$ = 150 MHz | | −162.5 | | dBm/Hz |
| $f_{DAC}$ = 1966.08 MSPS | $f_{OUT}$ = 180 MHz | | −163 | | dBm/Hz |
| 5 MHz BW LTE FIRST ADJACENT CHANNEL LEAKAGE RATIO (ACLR), SINGLE CARRIER | 0 dBFS, PLL off | | | | |
| $f_{DAC}$ = 1966.08 MSPS | $f_{OUT}$ = 50 MHz | | 79 | | dBc |
| | $f_{OUT}$ = 150 MHz | | 77 | | dBc |
| | $f_{OUT}$ = 180 MHz | | 77 | | dBc |
| 5 MHz BW LTE SECOND ACLR, SINGLE CARRIER | 0 dBFS, PLL off | | | | |
| $f_{DAC}$ = 1966.08 MSPS | $f_{OUT}$ = 50 MHz | | 82 | | dBc |
| | $f_{OUT}$ = 150 MHz | | 81 | | dBc |
| | $f_{OUT}$ = 180 MHz | | 81 | | dBc |

# ABSOLUTE MAXIMUM RATINGS

**Table 10.**

| Parameter | Rating |
|---|---|
| I120 to Ground | −0.3 V to AVDD33 + 0.3 V |
| SERDINx±, $V_{TT}$, $\overline{\text{SYNCOUT}}$±, TXEN | −0.3 V to SIOVDD33 + 0.3 V |
| IOUT±, QOUT± | −0.3 V to AVDD33 + 0.3 V |
| SYSREF± | GND − 0.5 V to +2.5 V |
| DACCLK± and REFCLK± to Ground | −0.3 V to PVDD12 + 0.3 V |
| $\overline{\text{RESET}}$, $\overline{\text{IRQ}}$, $\overline{\text{CS}}$, SCLK, SDIO, SDO, PROTECT_OUT to Ground | −0.3 V to IOVDD + 0.3 V |
| LDO_BYP1 | −0.3 V to SVDD12 + 0.3 V |
| LDO_BYP2 | −0.3 V to PVDD12 + 0.3 V |
| Ambient Operating Temperature ($T_A$) | −40°C to +85°C |
| Junction Temperature | 125°C |
| Storage Temperature Range | −65°C to +150°C |

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.

## THERMAL RESISTANCE

The exposed pad (EPAD) must be soldered to the ground plane for the 56-lead LFCSP. The EPAD provides an electrical, thermal, and mechanical connection to the board.

Typical $\theta_{JA}$, $\theta_{JB}$, and $\theta_{JC}$ values are specified for a 4-layer JESD51-7 high effective thermal conductivity test board for leaded surface-mount packages. $\theta_{JA}$ is obtained in still air conditions (JESD51-2). Airflow increases heat dissipation, effectively reducing $\theta_{JA}$. $\theta_{JB}$ is obtained following double-ring cold plate test conditions (JESD51-8). $\theta_{JC}$ is obtained with the test case temperature monitored at the bottom of the exposed pad.

$\Psi_{JT}$ and $\Psi_{JB}$ are thermal characteristic parameters obtained with $\theta_{JA}$ in still air test conditions.

Junction temperature ($T_J$) can be estimated using the following equations:

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

or

$$T_J = T_B + (\Psi_{JB} \times P)$$

where:
$T_T$ is the temperature measured at the top of the package.
$P$ is the total device power dissipation.
$T_B$ is the temperature measured at the board.

**Table 11. Thermal Resistance**

| Package | $\theta_{JA}$ | $\theta_{JB}$ | $\theta_{JC}$ | $\Psi_{JT}$ | $\Psi_{JB}$ | Unit |
|---|---|---|---|---|---|---|
| 56-Lead LFCSP[1] | 25.5 | 4.8 | 1.7 | 0.1 | 4.8 | °C/W |

[1] The exposed pad must be securely connected to the ground plane.

## ESD CAUTION

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

# PIN CONFIGURATION AND FUNCTION DESCRIPTIONS

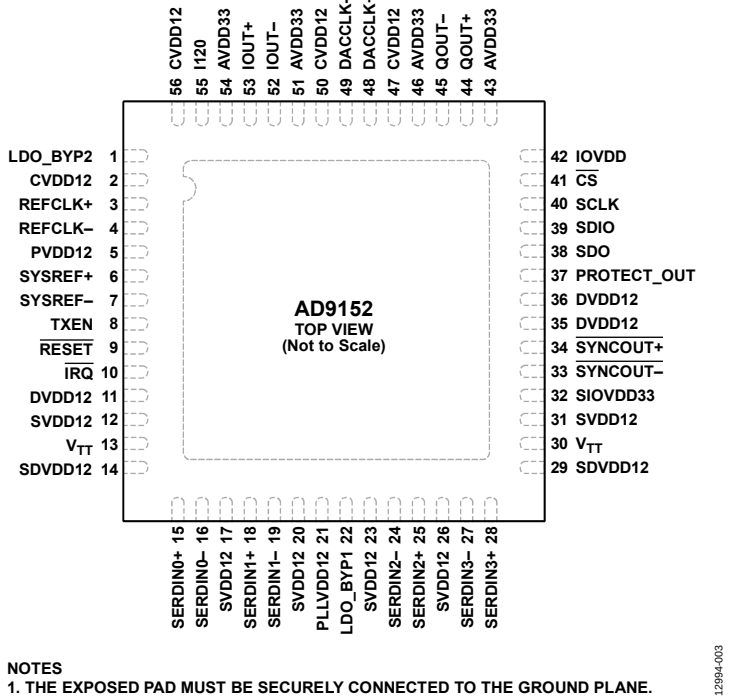


*Figure 3. Pin Configuration*

**Table 12. Pin Function Descriptions**

| Pin No. | Mnemonic | Description |
|---|---|---|
| 1 | LDO_BYP2 | LDO Clock Bypass for the DAC PLL. This pin requires a 1 Ω resistor in series with a 1 µF capacitor to ground. |
| 2 | CVDD12 | 1.2 V Clock Supply. |
| 3 | REFCLK+ | PLL Reference Clock Input, Positive. |
| 4 | REFCLK– | PLL Reference Clock Input, Negative. |
| 5 | PVDD12 | 1.2 V Supply. This pin supplies the DAC PLL and clock receiver circuitry. |
| 6 | SYSREF+ | Positive Reference Clock for Deterministic Latency. This pin is self biased for ac coupling. This pin may be ac-coupled or dc-coupled. |
| 7 | SYSREF– | Negative Reference Clock for Deterministic Latency. This pin is self biased for ac coupling. This pin may be ac-coupled or dc-coupled. |
| 8 | TXEN | Transmitter (Tx) Enable for I DAC and Q DAC. CMOS levels are determined with respect to IOVDD. |
| 9 | $\overline{\text{RESET}}$ | Reset (Active Low). CMOS levels are determined with respect to IOVDD. |
| 10 | $\overline{\text{IRQ}}$ | Interrupt Request (Active Low, Open Drain). |
| 11 | DVDD12 | 1.2 V Digital Supply. |
| 12 | SVDD12 | 1.2 V JESD204B Receiver (Rx) Analog Supply. |
| 13 | $V_{TT}$ | 1.2 V Termination Voltage. Connect this pin to the SVDD12 pin externally. |
| 14 | SDVDD12 | 1.2 V JESD204B Rx Digital Supply. |
| 15 | SERDIN0+ | Serial Channel Input 0, Positive. CML compliant. SERDIN0+ is 50 Ω terminated to the $V_{TT}$ pin voltage. This pin is ac-coupled only. Resistance calibrated. |
| 16 | SERDIN0– | Serial Channel Input 0, Negative. CML compliant. SERDIN0– is 50 Ω terminated to the $V_{TT}$ pin voltage. This pin is ac-coupled only. Resistance calibrated. |
| 17 | SVDD12 | 1.2 V JESD204B Rx Analog Supply. |
| 18 | SERDIN1+ | Serial Channel Input 1, Positive. CML compliant. SERDIN1+ is 50 Ω terminated to the $V_{TT}$ pin voltage. This pin is ac-coupled only. Resistance calibrated. |
| 19 | SERDIN1– | Serial Channel Input 1, Negative. CML compliant. SERDIN1– is 50 Ω terminated to the $V_{TT}$ pin voltage. This pin is ac-coupled only. Resistance calibrated. |
| 20 | SVDD12 | 1.2 V JESD204B Rx Analog Supply. |
| 21 | PLLVDD12 | 1.2 V SERDES PLL Supply. |
| 22 | LDO_BYP1 | LDO SERDES Bypass. This pin requires a 1 Ω resistor in series with a 1 µF capacitor to ground. |

| Pin No. | Mnemonic | Description |
|---|---|---|
| 23 | SVDD12 | 1.2 V JESD204B Rx Analog Supply. |
| 24 | SERDIN2− | Serial Channel Input 2, Negative. CML compliant. SERDIN2− is 50 Ω terminated to the $V_{TT}$ pin voltage. This pin is ac-coupled only. Resistance calibrated |
| 25 | SERDIN2+ | Serial Channel Input 2, Positive. CML compliant. SERDIN2+ is 50 Ω terminated to the $V_{TT}$ pin voltage. This pin is ac-coupled only. Resistance calibrated. |
| 26 | SVDD12 | 1.2 V JESD204B Rx Analog Supply. |
| 27 | SERDIN3− | Serial Channel Input 3, Negative. CML compliant. SERDIN3− is 50 Ω terminated to the $V_{TT}$ pin voltage. This pin is ac-coupled only. Resistance calibrated. |
| 28 | SERDIN3+ | Serial Channel Input 3, Positive. CML compliant. SERDIN3+ is 50 Ω terminated to the $V_{TT}$ pin voltage. This pin is ac-coupled only. Resistance calibrated. |
| 29 | SDVDD12 | 1.2 V JESD204B Rx Digital Supply. |
| 30 | $V_{TT}$ | 1.2 V Termination Voltage. Connect $V_{TT}$ to the SVDD12 pin externally. |
| 31 | SVDD12 | 1.2 V JESD204B Rx Analog Supply. |
| 32 | SIOVDD33 | 3.3 V Supply for Equalizers. |
| 33 | $\overline{\text{SYNCOUT}}$− | Negative LVDS Sync Output Signal. |
| 34 | $\overline{\text{SYNCOUT}}$+ | Positive LVDS Sync Output Signal. |
| 35 | DVDD12 | 1.2 V Digital Supply. |
| 36 | DVDD12 | 1.2 V Digital Supply. |
| 37 | PROTECT_OUT | Protection Indicator for I DAC and Q DAC. CMOS levels are determined with respect to IOVDD. |
| 38 | SDO | Serial Port Data Output. CMOS levels are determined with respect to IOVDD. |
| 39 | SDIO | Serial Port Data Input/Output. CMOS levels are determined with respect to IOVDD. |
| 40 | SCLK | Serial Port Clock Input. CMOS levels are determined with respect to IOVDD. |
| 41 | $\overline{\text{CS}}$ | Serial Port Chip Select, Active Low. CMOS levels are determined with respect to IOVDD. |
| 42 | IOVDD | 1.8 V IOVDD Supply for CMOS Input/Output and SPI. |
| 43 | AVDD33 | 3.3 V Analog Supply for the DAC Cores. |
| 44 | QOUT+ | Q DAC Positive Current Output. |
| 45 | QOUT− | Q DAC Negative Current Output. |
| 46 | AVDD33 | 3.3 V Analog Supply for the DAC Cores. |
| 47 | CVDD12 | 1.2 V Clock Supply. |
| 48 | DACCLK+ | Positive Device Clock When PLL Is Not Used. |
| 49 | DACCLK− | Negative Device Clock When PLL Is Not Used. |
| 50 | CVDD12 | 1.2 V Clock Supply. |
| 51 | AVDD33 | 3.3 V Analog Supply for the DAC Cores. |
| 52 | IOUT− | I DAC Negative Current Output. |
| 53 | IOUT+ | I DAC Positive Current Output. |
| 54 | AVDD33 | 3.3 V Analog Supply for the DAC Cores. |
| 55 | I120 | Output Current Generation Pin for the DAC Full-Scale Current. Tie a 4 kΩ resistor from this pin to the ground plane. |
| 56 | CVDD12 | 1.2 V Clock Supply. |
| | EPAD | Exposed Pad. The exposed pad must be securely connected to the ground plane. |

# TERMINOLOGY

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

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

**Offset Error**
Offset error is the deviation of the output current from the ideal of 0 mA. For IOUT+/QOUT+, 0 mA output is expected when all inputs are set to 0. For IOUT−/QOUT−, 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 ppm of full-scale range (FSR) per degree Celsius. For reference drift, the drift is reported in ppm per degree Celsius.

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

**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 $f_{DATA}$ (interpolation rate), a digital filter can be constructed that has a sharp transition band near $f_{DATA}/2$. Images that typically appear around $f_{DAC}$ (output data rate) can be greatly suppressed.

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

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

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

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

**Logical Lane**
Logical Lane x refers to physical lanes after optionally being remapped by the crossbar block (Register 0x308 and Register 0x309).

# TYPICAL PERFORMANCE CHARACTERISTICS

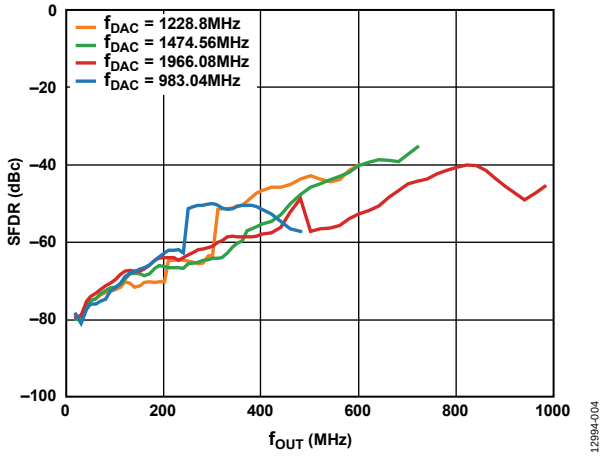


*Figure 4. Single Tone (0 dBFS) SFDR vs. $f_{OUT}$ in the First Nyquist Zone over $f_{DAC}$*

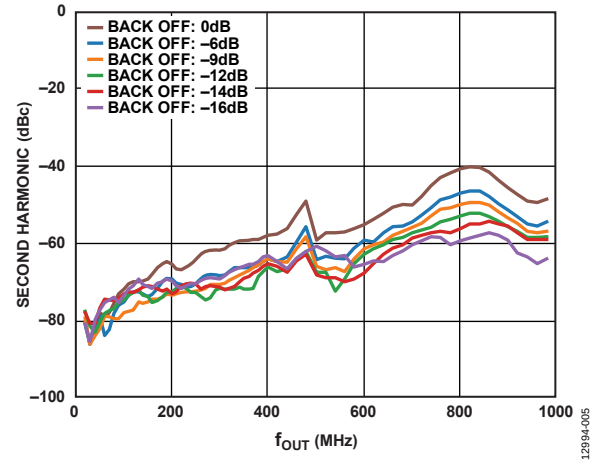


*Figure 5. Single Tone Second Harmonic vs. $f_{OUT}$ in the First Nyquist Zone over Digital Back Off, $f_{DAC}$ = 1966.08 MHz*

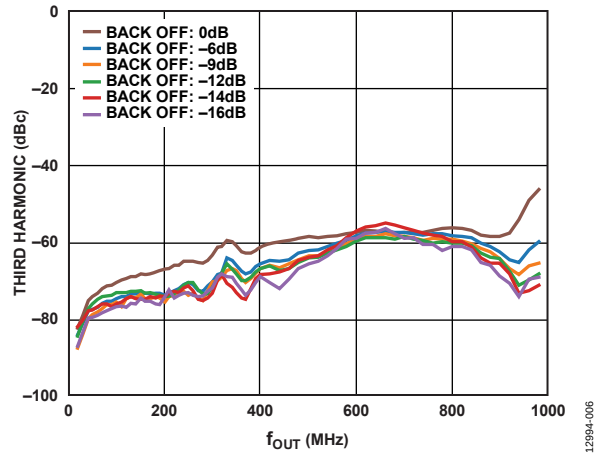


*Figure 6. Single Tone Third Harmonic vs. $f_{OUT}$ in the First Nyquist Zone over Digital Back Off, $f_{DAC}$ = 1966.08 MHz*

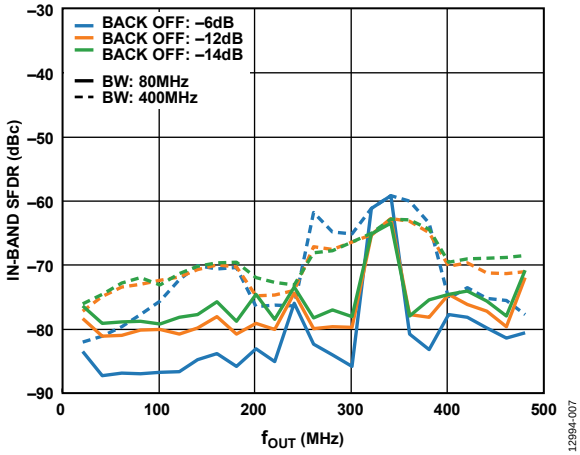


*Figure 7. In-Band, Single Tone SFDR vs. $f_{OUT}$ in 80 MHz and 400 MHz Bandwidths, $f_{DAC}$ = 983.04 MHz*

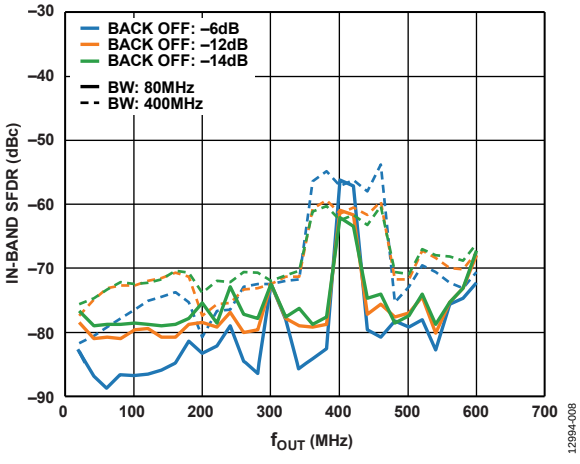


*Figure 8. In-Band, Single Tone SFDR vs. $f_{OUT}$ in 80 MHz and 400 MHz Bandwidths, $f_{DAC}$ = 1228.8 MHz*

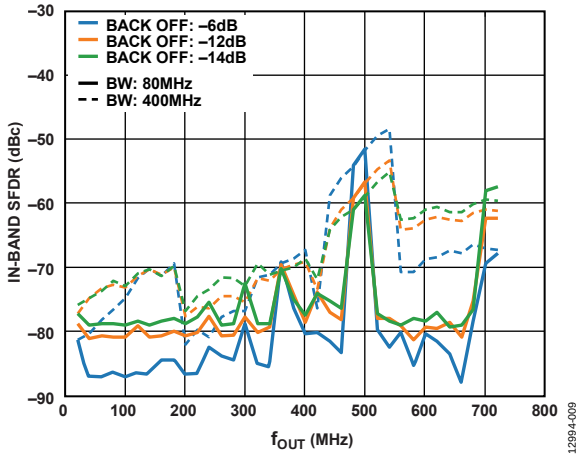


*Figure 9. In-Band, Single Tone SFDR vs. $f_{OUT}$ in 80 MHz and 400 MHz Bandwidths, $f_{DAC}$ = 1474.56 MHz*

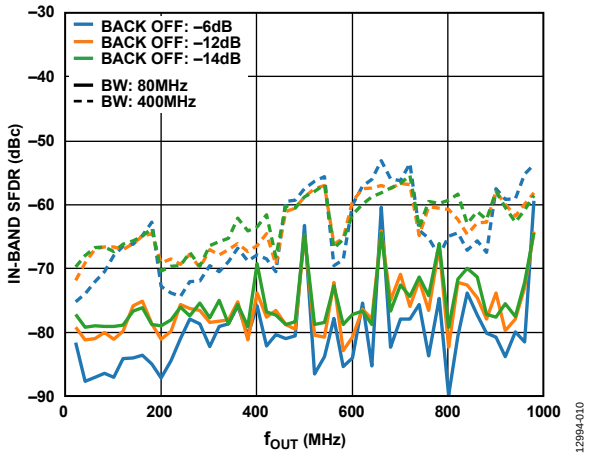


Figure 10. In-Band, Single Tone SFDR vs. $f_{OUT}$ in 80 MHz and 400 MHz Bandwidths, $f_{DAC}$ = 1966.08 MHz

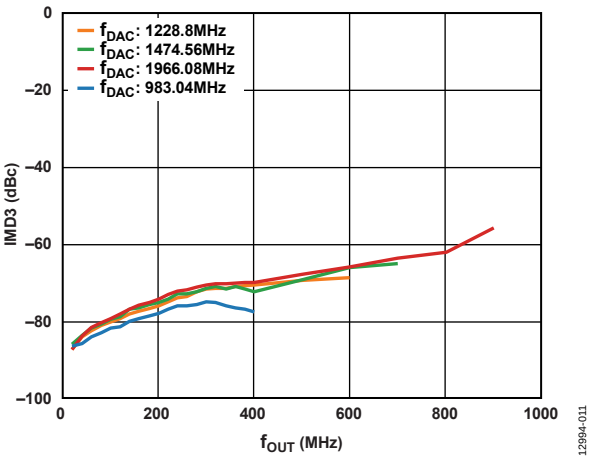


Figure 11. Two-Tone, Third-Order IMD (IMD3) vs. $f_{OUT}$ over $f_{DAC}$

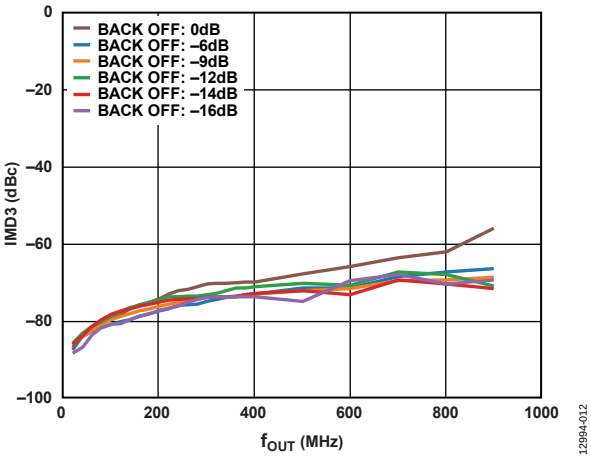


Figure 12. Two-Tone, Third-Order IMD (IMD3) vs. $f_{OUT}$ over Digital Back Off, $f_{DAC}$ = 1966.08 MHz

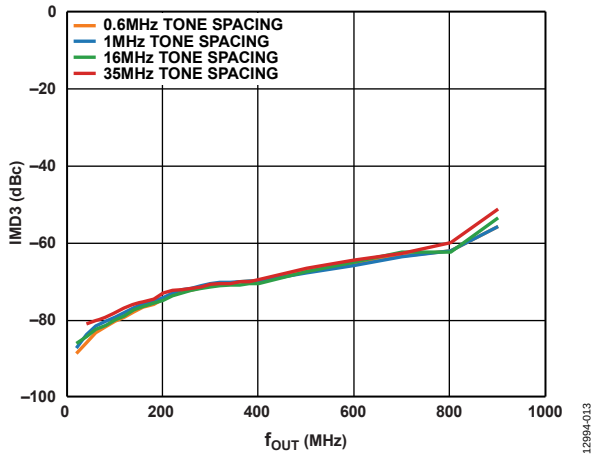


Figure 13. Two-Tone, Third-Order IMD (IMD3) vs. $f_{OUT}$ over Tone Spacing, $f_{DAC}$ = 1966.08 MHz

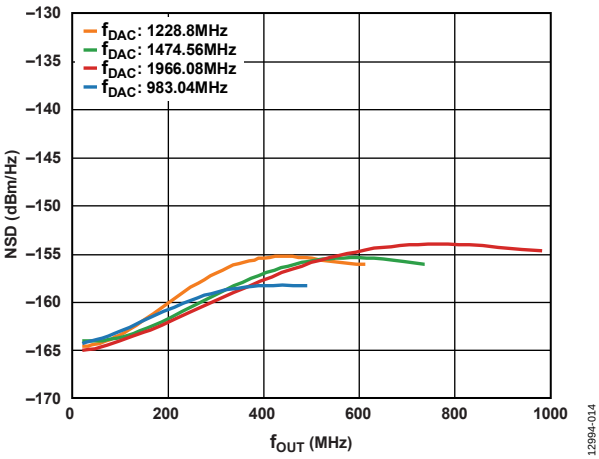


Figure 14. Single Tone (0 dBFS) NSD vs. $f_{OUT}$ over $f_{DAC}$

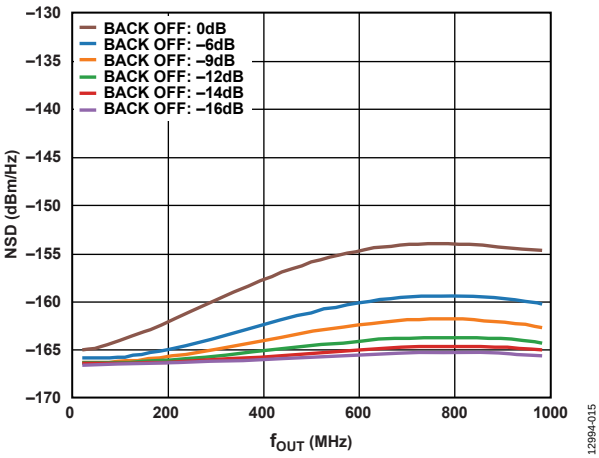


Figure 15. Single Tone NSD vs. $f_{OUT}$ over Digital Back Off, $f_{DAC}$ = 1966.08 MHz

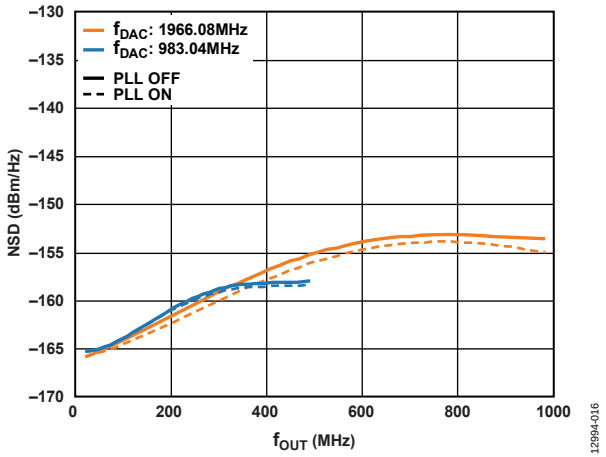


Figure 16. Single Tone NSD vs. $f_{OUT}$, PLL On and Off

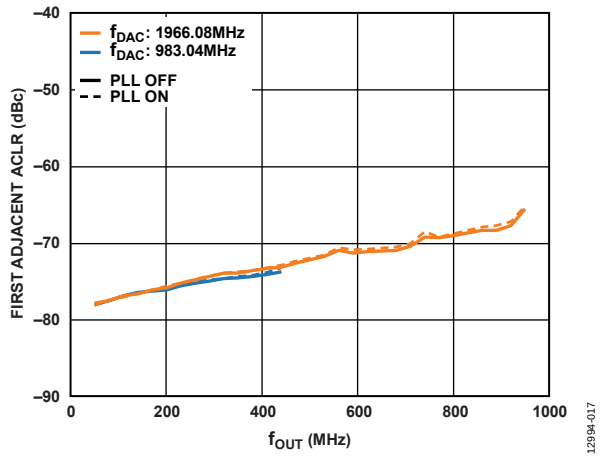


Figure 17. One-Carrier (1C) 5 MHz Bandwidth LTE, First Adjacent ACLR vs. $f_{OUT}$, PLL On and Off

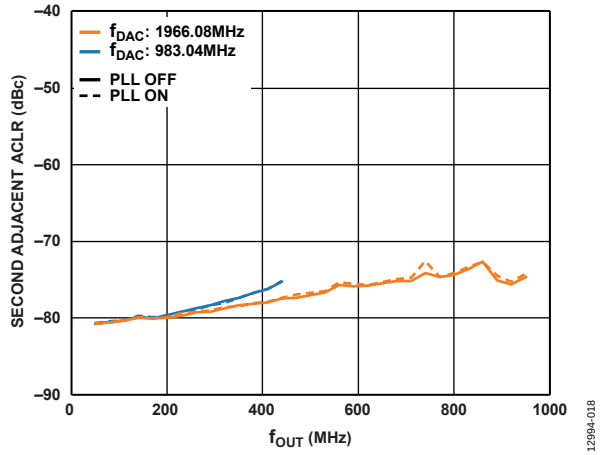


Figure 18. 1C 5 MHz Bandwidth LTE, Second Adjacent ACLR vs. $f_{OUT}$, PLL On and Off

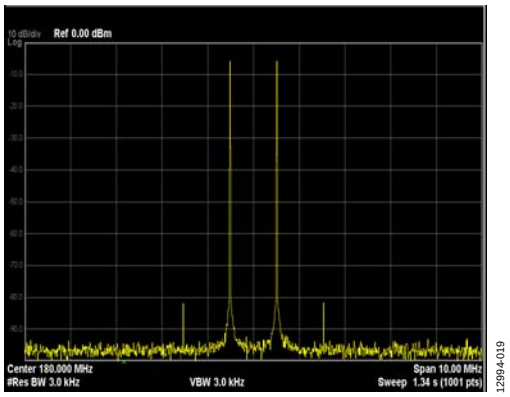


Figure 19. Two-Tone, Third-Order IMD Performance, IF = 180 MHz, $f_{DAC}$ = 1966.08 MHz

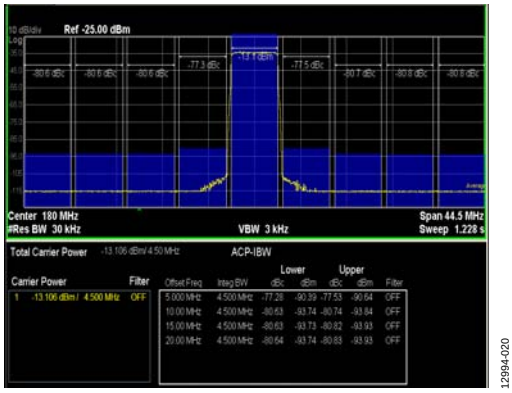


Figure 20. 1C 5 MHz Bandwidth LTE ACLR Performance, IF = 180 MHz, $f_{DAC}$ = 1966.08 MHz

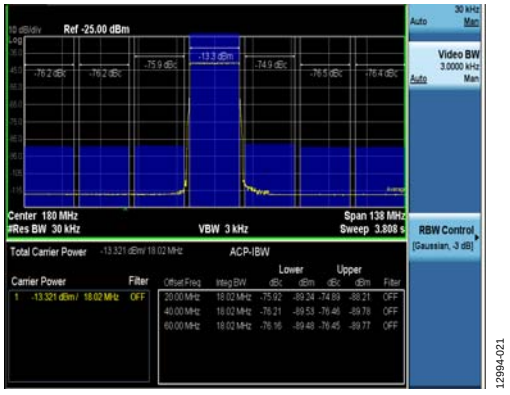


Figure 21. 1C 20 MHz Bandwidth LTE ACLR Performance, IF = 180 MHz, $f_{DAC}$ = 1966.08 MHz

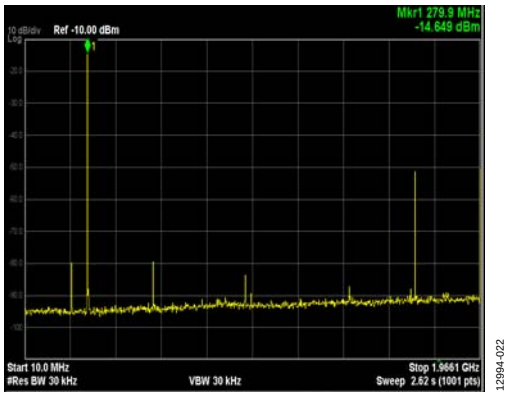


Figure 22. Single Tone, $f_{DAC}$ =1966.08 MHz, $f_{OUT}$ = 280 MHz, −14 dBFS

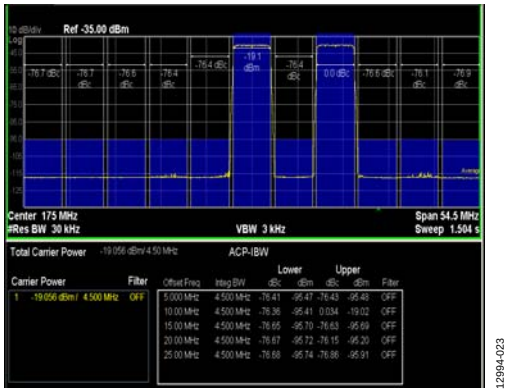


Figure 23. Two-Carrier (2C) 2× 5 MHz Bandwidth with 5 MHz Gap LTE ACLR Performance, IF = 180 MHz, $f_{DAC}$ = 1966.08 MHz

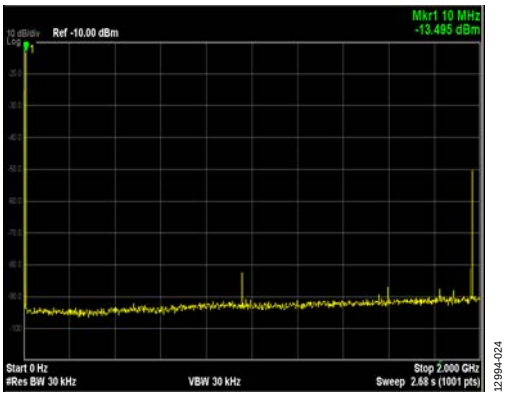


Figure 24. Single Tone SFDR, $f_{DAC}$ =1966.08 MHz, 4× Interpolation, $f_{OUT}$ = 10 MHz, −14 dBFS

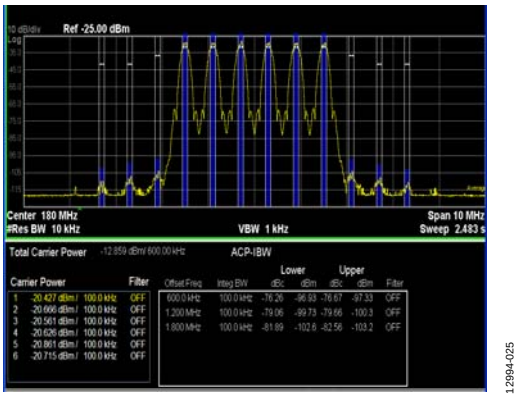


Figure 25. Six-Carrier (6C) Spaced by 600 kHz GSM Edge ACP Performance, IF = 180 MHz, $f_{DAC}$ = 1966.08 MHz

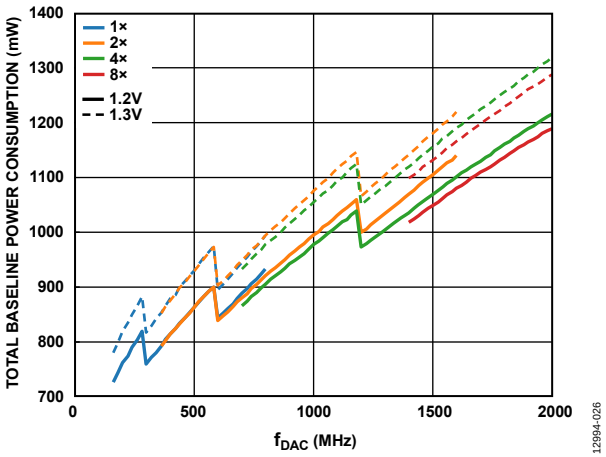


Figure 26. Total Baseline Power Consumption vs. $f_{DAC}$ over Interpolation

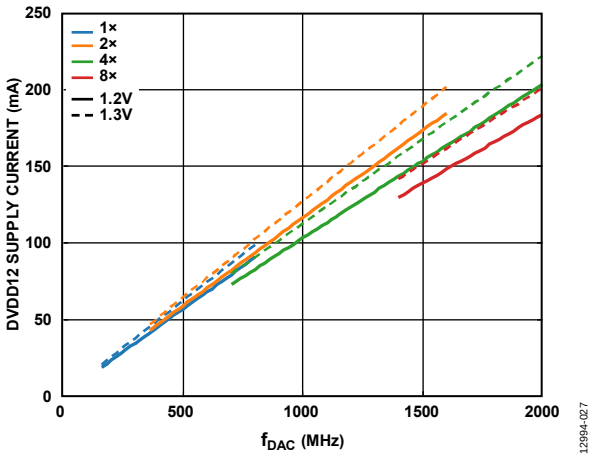


Figure 27. DVDD12 Supply Current vs. $f_{DAC}$ over Interpolation

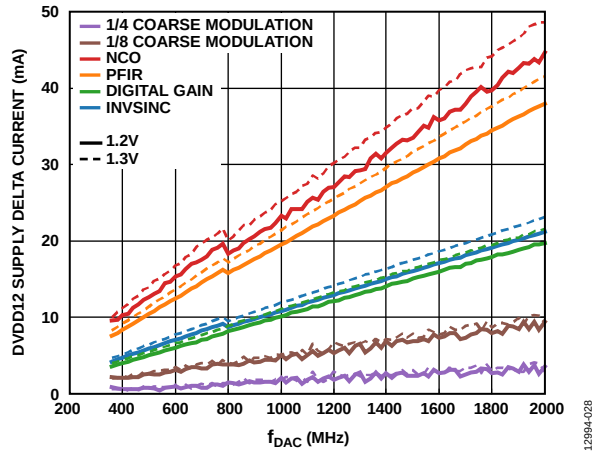


Figure 28. DVDD12 Supply Delta Current vs. $f_{DAC}$ over Digital Functions

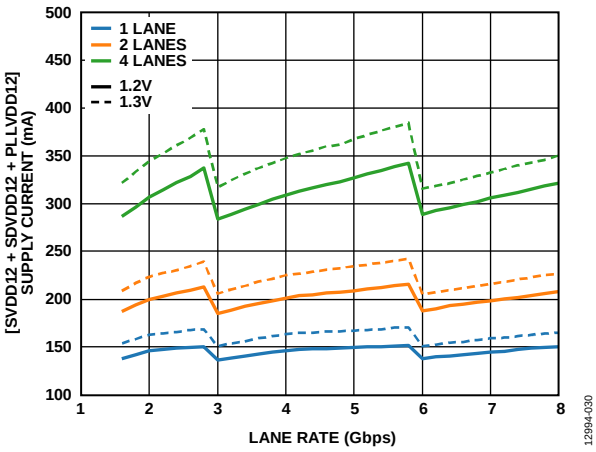


Figure 30. Total SERDES Supply Current (SVDD12, SDVDD12, PLLVDD12) vs. Lane Rate; One, Two, and Four Lanes

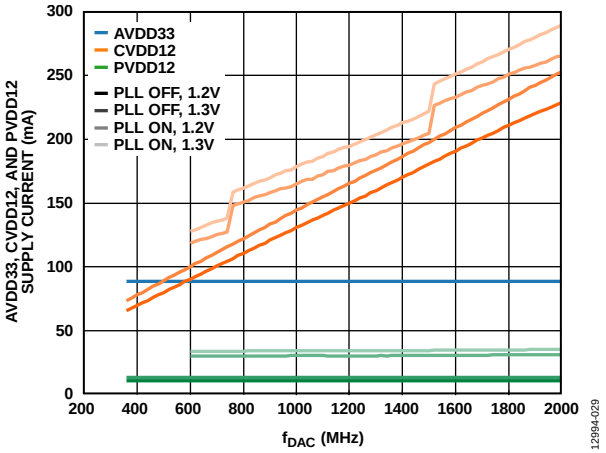


Figure 29. AVDD33, CVDD12, PVDD12 Supply Current vs. $f_{DAC}$

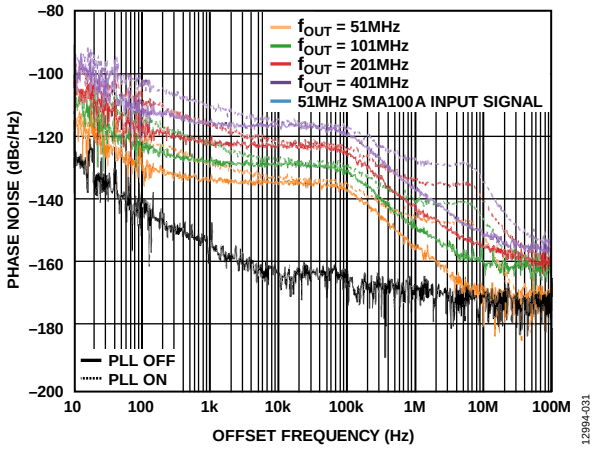


Figure 31. Single Tone Phase Noise vs. Offset Frequency at Four $f_{OUT}$ Values and with an SMA100A Signal Generator, $f_{DAC}$ = 1.96608 GHz, PLL On and Off

# THEORY OF OPERATION

The AD9152 is a 16-bit, dual DAC with a SERDES interface. Figure 2 shows a detailed functional block diagram of the AD9152. Four high speed serial lanes carry data at a maximum speed of 12.38 Gbps, and a 1.238 GSPS input data rate to the DACs. 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 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, a high fidelity direct external DAC sampling clock, or a 2× DAC frequency RF clock. The device can be configured to operate in one-, two-, or four-lane modes, depending on the required input data rate.

The digital datapath of the AD9152 offers four interpolation modes (1×, 2×, 4×, and 8×) through three half-band filters with a maximum DAC sample rate of 2.25 GSPS. An inverse sinc filter compensates for sinc related roll-off. The PFIR filter compensates the gain over frequency in a more flexible way.

The AD9152 DAC cores provide a fully differential current output with a nominal full-scale current of 20 mA. The full-scale current, $I_{OUTFS}$, is user adjustable to between 4.04 mA and 20.22 mA, typically. The differential current outputs are complementary and are optimized for easy integration with the Analog Devices ADRF6720 AQM. The AD9152 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 from link establishment to link establishment. An external alignment (SYSREF±) signal makes the AD9152 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 Device Setup Guide section). Simple SPI initialization routines set up the JESD204B link and are included in the evaluation board package. The following sections describe the various blocks of the AD9152 in greater detail. Descriptions of the JESD204B interface, control parameters, and various registers to set up and monitor the device are provided. The recommended start-up routine reliably sets up the data link.

# SERIAL PORT OPERATION

The serial port is a flexible, synchronous serial communications port that allows easy interfacing with many industry-standard microcontrollers and microprocessors. The serial input/output is compatible with most synchronous transfer formats, including both the Motorola SPI and Intel® SSR protocols. The interface allows read/write access to all registers that configure the AD9152. MSB first or LSB first transfer formats are supported. The serial port interface is a 4-wire or a 3-wire (by default) interface in which the input and output share a single-pin input/output (SDIO).

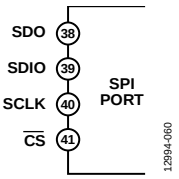


*Figure 32. Serial Port Interface Pins*

There are two phases to a communication cycle with the AD9152. 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 needed to transfer N bytes during the transfer cycle. Registers change immediately upon writing to the last bit of each transfer byte, except for the frequency tuning word (FTW) and the numerically controlled oscillator (NCO) phase offsets, which change only when the frequency tuning word (FTW) FTW_UPDATE_REQ bit (Register 0x113, Bit 0) 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 14 to Bit 0 of the instruction word, determine the register that is accessed during the data transfer portion of the communication cycle. For multibyte transfers, [A14:0] is the starting address. The remaining register addresses are generated by the device based on the address increment bits (Register 0x000, Bit 5 and Bit 2). If the address increment bits are set high, multibyte SPI writes start on A[14:0] and increment by 1 every 8 bits sent/received. If the address increment bits are set to 0, the address decrements by 1 every 8 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 10 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. It allows more than one device to be used on the same serial communications lines. The SDIO pin goes to a high impedance state when this input is high. During the communication cycle, 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 bits (Register 0x000, Bit 6 and Bit 1). The default is MSB first (the LSB first bits = 0).

When the LSB first bits = 0 (MSB first), the instruction and data bits are 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 input and output.

Multibyte data transfers can be performed as well. This is achieved 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 the address increment bits (Register 0x000, Bit 5 and Bit 2). When the address increment bits are 1, the multicycle addresses are incremented. When the address increment bits are 0, the addresses are decremented. A new write cycle can always be initiated by bringing $\overline{CS}$ high and then low again.

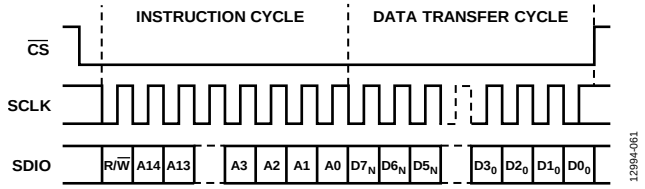


*Figure 33. Serial Register Interface Timing, MSB First, Register 0x000, Bit 5 and Bit 2 = 0*

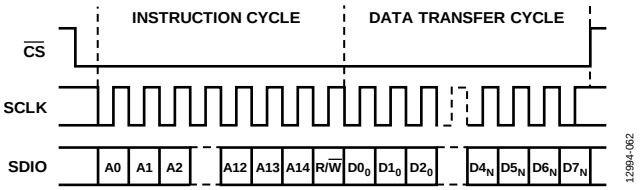


*Figure 34. Serial Register Interface Timing, LSB First, Register 0x000, Bit 5 and Bit 2 = 1*

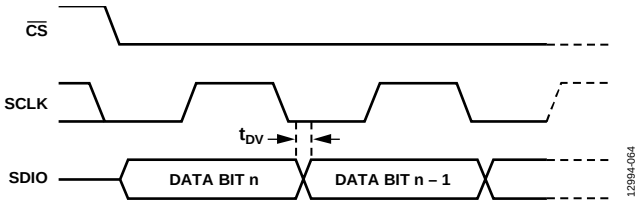


*Figure 35. Timing Diagram for Serial Port Register Read*

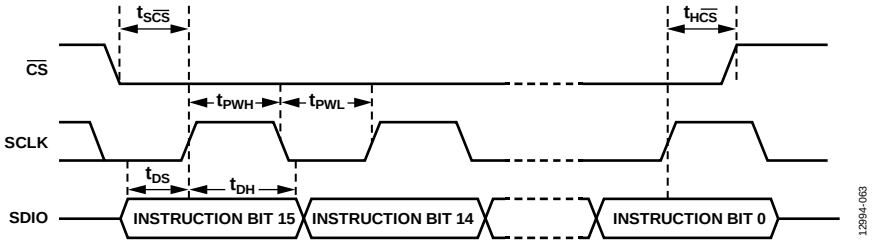


*Figure 36. Timing Diagram for Serial Port Register Write*

# CHIP INFORMATION

Register 0x003 to Register 0x006 contain chip information, as shown in Table 14.

**Table 14. Chip Information**

| Information | Description |
|---|---|
| Chip Type | Register 0x003. The product type is high speed DAC, which is represented by a code of 0x04. |
| Product ID | 8 MSBs in Register 0x005 and 8 LSBs in Register 0x004. The product ID is 0x9152. |
| Device Revision | Register 0x006, Bits[4:0]. The device revision is 0x8. |

# DEVICE SETUP GUIDE

## OVERVIEW

The sequence of steps to properly set up the AD9152 is as follows:

1. Set up the SPI interface, power up necessary circuit blocks, make the required writes to the configuration registers, and set up the DAC clocks (see Step 1: Start Up the DAC).
2. Set the digital features (see Step 2: Digital Datapath).
3. Set up the JESD204B links (see Step 3: Transport Layer).
4. Set up the physical layer of the SERDES interface (see Step 4: Physical Layer).
5. Set up the data link layer of the SERDES interface (see Step 5: Data Link Layer).
6. Check for errors (see Step 6: Optional Error Monitoring).
7. Optionally, enable any needed features as described in Step 7: Optional Features.

A specific working start-up sequence example is given in the Example Start-Up Sequence section.

The register writes listed in Table 15 to Table 23 give the register writes necessary to set up the AD9152. Consider printing this setup guide and filling in the Value column with appropriate variable values for the conditions of the desired application.

The notation 0x indicates register settings that the user must fill in. To fill in the unknown register values, select the correct settings for each variable listed in the Variable column. The Description column describes how to set variables or provides a link to a section where this is described. Register settings with specified values are fixed settings to be used in all cases. A variable is noted by concatenating multiple terms. For example, PdDACs is a variable that corresponds to the value determined for Register 0x011, Bits[6:5].

## STEP 1: START UP THE DAC

This section describes how to set up the SPI interface, power up necessary circuit blocks, write to the required configuration registers, and set up the DAC clocks.

**Table 15. Power-Up and DAC Initialization Settings**

| Addr. | Bit No. | Value[1] | Variable | Description |
|---|---|---|---|---|
| 0x000 | | 0xBD | | Soft reset. |
| 0x000 | | 0x3C | | Deassert reset, set 4-wire SPI. |
| 0x011 | | 0x | | |
| | 7 | 0 | | Power up band gap. |
| | [6:5] | | PdDACs | PdDACs = 0 if both DACs are used. If not, see the DAC Power-Down Setup section. |
| | 4 | 0 | | Power up digital clocks. |
| | [3:2] | | PdCLKs | PdCLKs = 0 if both DACs are used. |
| | 1 | 0 | | Power up the PCLK. |
| | 0 | 0 | | Power up the clock receiver. |
| 0x080 | | 0x | | |
| | 2 | | DUTY_EN | DUTY_EN = 1 if using the duty function. |
| 0x081 | | 0x | | |
| | 4 | | PdSysref | PdSysref = 0x0 for Subclass 1. PdSysref = 0x1 for Subclass 0. See the Subclass Setup section. |
| 0x1CD[2] | | 0xD8 | | Band gap configuration. |

[1] 0x denotes a register value that the user must fill in. See the Variable and Description columns for information on selecting the appropriate register value.
[2] Register 0x1CD must be set to the recommended value and does not appear in the register map.

The following registers must be written to and values changed from default for the device to work correctly and must be written after any soft reset, hard reset, or power-up occurs. All registers in Table 16 do not appear in the register map.

**Table 16. Required SERDES PLL Register Settings**

| Address | Value[1] | Description |
|---|---|---|
| 0x284 | 0x62 | SERDES PLL configuration |
| 0x285 | 0xC9 | SERDES PLL configuration |
| 0x286 | 0xE | SERDES PLL configuration |
| 0x287 | 0x12 | SERDES PLL configuration |
| 0x28A | 0x | See Table 36 |
| 0x28B | 0x0 | SERDES PLL configuration |
| 0x290 | 0x89 | SERDES PLL configuration |
| 0x291 | 0x | See Table 36 |
| 0x294 | 0x24 | SERDES PLL configuration |
| 0x296 | 0x | See Table 36 |
| 0x297 | 0xD | SERDES PLL configuration |
| 0x299 | 0x2 | SERDES PLL configuration |
| 0x29A | 0x8E | SERDES PLL configuration |
| 0x29C | 0x2A | SERDES PLL configuration |
| 0x29F | 0x7E | SERDES PLL configuration |
| 0x2A0 | 0x6 | SERDES PLL configuration |

[1] 0x denotes a register value that the user must fill in. See the Variable and Description columns for information on selecting the appropriate register value.

If using the optional DAC PLL, also set the registers in Table 17 and Table 18. The registers in Table 17 optimize the performance of the SERDES PLL and must be set to the fixed value as required. Some registers in Table 17 do not appear in the register map.

**Table 17. Required DAC PLL Configurations**

| Address | Value | Description |
|---|---|---|
| 0x08D | 0x7B | DAC PLL configuration |
| 0x1B0 | 0x0 | DAC PLL configuration |
| 0x1B9 | 0x24 | DAC PLL configuration |
| 0x1BC | 0xD | DAC PLL configuration |
| 0x1BE | 0x2 | DAC PLL configuration |
| 0x1BF | 0x8E | DAC PLL configuration |
| 0x1C0 | 0x2A | DAC PLL configuration |
| 0x1C4 | 0x7E | DAC PLL configuration |
| 0x1C1 | 0x2C | DAC PLL configuration |

**Table 18. Optional DAC PLL Configuration Procedure**

| Addr. | Value[1] | Variable | Description |
|---|---|---|---|
| 0x08B | 0x | LODivMode | See the DAC PLL Setup section |
| 0x08C | 0x | RefDivMode | See the DAC PLL Setup section |
| 0x085 | 0x | BCount | See the DAC PLL Setup section |
| 0x1B6 | 0x | LookUpVals | See Table 73 |
| 0x1B5 | 0x | | See Table 73 |
| 0x1BB | 0x | LookUpVals | See Table 73 |
| 0x1B4 | 0x78 | | Optimal DAC PLL VCO settings |
| 0x1C5 | 0x | | See Table 73 |
| 0x08A | 0x12 | | Optimal DAC PLL VCO settings |
| 0x087 | 0x62 | | Optimal DAC PLL loop filter settings |
| 0x088 | 0xC9 | | Optimal DAC PLL loop filter settings |
| 0x089 | 0x0E | | Optimal DAC PLL loop filter settings |
| 0x083 | 0x10 | | Enable the DAC PLL[2] |

[1] 0x denotes a register value that the user must fill in. See the Variable and Description columns for information on selecting the appropriate register value.
[2] Verify that Register 0x084, Bit 1 reads back 1 after enabling the DAC PLL to indicate that the DAC PLL has locked.

## STEP 2: DIGITAL DATAPATH

This section describes which interpolation filters to use and sets the data format being used. Additional digital features are available including fine and coarse modulation, digital gain scaling, and an inverse sinc filter used to improve pass-band flatness. Table 23 provides further details on the feature blocks available.

**Table 19. Digital Datapath Settings**

| Addr. | Bit No. | Value[1] | Variable | Description |
|---|---|---|---|---|
| 0x112 | | 0x | InterpMode | Select interpolation mode; see the Interpolation section. |
| 0x110 | | 0x | | |
| | 7 | | DataFmt | DataFmt = 0 if twos complement; DataFmt = 1 if unsigned binary. |

[1] 0x denotes a register value that the user must fill in. See the Variable and Description columns for information on selecting the appropriate register value.

## STEP 3: TRANSPORT LAYER

This section describes how to set up the JESD204B links. The parameters are determined by the desired JESD204B operating mode. See the JESD204B Setup section for details.

**Table 20. Transport Layer Settings**

| Addr. | Bit No. | Value[1] | Variable | Description |
|---|---|---|---|---|
| 0x200 | | 0x00 | | Power up the interface. |
| 0x201 | | 0x | UnusedLanes | See the JESD204B Setup section. |
| 0x300 | | 0x | | |
| | 6 | | CheckSumMode | See the JESD204B Setup section. |
| 0x450 | | 0x | DID | Set DID to match the device ID sent by the transmitter. |
| 0x451 | | 0x | BID | Set BID to match the bank ID sent by the transmitter. |
| 0x452 | | 0x | LID | Set LID to match the lane ID sent by the transmitter. |
| 0x453 | | 0x | | |
| | 7 | | Scrambling | See the JESD204B Setup section. |
| | [4:0] | | L − 1[2] | See the JESD204B Setup section. |
| 0x454 | | 0x | F − 1[2] | See the JESD204B Setup section. |
| 0x455 | | 0x | K − 1[2] | See the JESD204B Setup section. |
| 0x456 | | 0x | M − 1[2] | See the JESD204B Setup section. |
| 0x457 | | 0x | N − 1[2] | N = 16. |
| 0x458 | | 0x | | |
| | 5 | | Subclass | See the JESD204B Setup section. |
| | [4:0] | | Np − 1[2] | Np = 16. |
| 0x459 | | 0x | | |
| | 5 | | JESDVer | JESDVer = 1 for JESD204B, JESDVer = 0 for JESD204A. |
| | [4:0] | | S − 1[2] | See the JESD204B Setup section. |
| 0x45A | | 0x | | |
| | 7 | | HD | See the JESD204B Setup section. |
| | [4:0] | | CF | CF must equal 0. |
| 0x45D | | 0x | Lane0Checksum | See the JESD204B Setup section. |
| 0x46C | | 0x | Lanes | Deskew lanes. |
| 0x476 | | 0x | F | See the JESD204B Setup section. |
| 0x47D | | 0x | Lanes | Enable lanes. See the JESD204B Setup section. |

[1] 0x denotes a register value that the user must fill in. See the Variable and Description columns for information on selecting the appropriate register value.
[2] This JESD204B link parameter is programmed in n − 1 notation as noted. For example, if the setup requires L = 4 (4 lanes per link), program L − 1 or 3 into Register 0x453, Bits[4:0].

## STEP 4: PHYSICAL LAYER

This section describes how to set up the physical layer of the SERDES interface. In this section, the input termination settings are configured along with the CDR sampling and SERDES PLL.

**Table 21. Device Configurations and Physical Layer Settings**

| Addr. | Bit No. | Value[1] | Variable | Description |
|---|---|---|---|---|
| 0x2A7 | | 0x01 | | Autotune PHY setting. |
| 0x314 | | 0x01 | | SERDES SPI configuration. |
| 0x230 | | 0x | | |
| | 5 | | Halfrate | Set up the CDR; see the SERDES Clocks Setup section. |
| | [4:2] | 0x2 | | SERDES PLL default configuration. |
| | 1 | | OvSmp | Set up the CDR; see the SERDES Clocks Setup section. |
| | 0 | 1 | | $\overline{\text{SYNCOUT}}$± swing VOD is set to 350 mV. |
| 0x206 | | 0x00 | | Reset the CDR. |
| 0x206 | | 0x01 | | Release CDR reset. |
| 0x289 | | 0x | | |
| | 2 | 1 | | SERDES PLL configuration. |
| | [1:0] | | PLLDiv | Set the CDR oversampling for PLL; see the SERDES Clocks Setup section. |
| 0x280 | | 0x01 | | Enable the SERDES PLL.[2] |
| 0x268 | | 0x | | |
| | [7:6] | | EqMode | See the Equalization Mode Setup section. |
| | [5:0] | 0x22 | | Required value (default). |

[1] 0x denotes a register value that the user must fill in. See the Variable and Description columns for information on selecting the appropriate register value.
[2] Verify that Register 0x281, Bit 0 reads back 1 after enabling the SERDES PLL to indicate that the SERDES PLL has locked.

## STEP 5: DATA LINK LAYER

This section describes how to set up the data link layer of the SERDES interface. This section deals with SYSREF processing, setting deterministic latency, and establishing the link.

**Table 22. Data Link Layer Settings**

| Addr. | Bit No. | Value[1] | Variable | Description |
|---|---|---|---|---|
| 0x301 | | 0x | Subclass | See the JESD204B Setup section. |
| 0x304 | | 0x | LMFCDel | See the Link Latency Setup section. |
| 0x306 | | 0x | LMFCVar | See the Link Latency Setup section. |
| 0x03A | | 0x01 | | Set sync mode = one shot sync; see the Syncing LMFC Signals section for other sync options. |
| 0x03A | | 0x81 | | Enable the sync machine. |
| 0x03A | | 0xC1 | | Arm the sync machine. |
| SYSREF± | | | | If Subclass = 1, ensure that at least one SYSREF± edge is sent to the device.[2] |
| 0x308 to 0x309 | | 0x | XBarVals | If remapping lanes, set up cross-bar; see the Crossbar Setup section. |
| 0x334 | | 0x | InvLanes | Invert the polarity of the desired logical lanes. Bit x of InvLanes must be a 1 for each Logical Lane x to invert. |
| 0x300 | | 0x | | Enable the link. |
| | 6 | | ChkSmMd | See the JESD204B Setup section. |
| | 0 | 1 | | Enable the link. |

[1] 0x denotes a register value that the user must fill in. See the Variable and Description columns for information on selecting the appropriate register value.
[2] Verify that Register 0x03B, Bit 3 reads back 1 after sending at least one SYSREF± edge to the device to indicate that the LMFC sync machine has properly locked.

## STEP 6: OPTIONAL ERROR MONITORING

For JESD204B error monitoring, see the JESD204B Error Monitoring section. For other error checks, see the Interrupt Request Operation section.

## STEP 7: OPTIONAL FEATURES

A number of optional features can be enabled. Table 22 provides links to the sections describing each feature.

**Table 23. Optional Features**

| Feature | Default | Description |
|---|---|---|
| Digital Modulation | Off | Modulates the data with a desired carrier. See the Digital Modulation section. |
| Inverse Sinc | Off | Improves pass-band flatness. See the Inverse Sinc section. |
| Digital Gain | 0 dB | Multiplies data by a factor. Can compensate inverse sinc usage or balance I/Q amplitude. See the Digital Gain section. |
| Phase Adjust | Off | Used to balance I/Q phase. See the Phase Adjust section. |
| DC Offset | Off | Used to cancel LO leakage. See the DC Offset section. |
| Coarse Group Delay Adjustment | 0 | Used to control overall latency. See the Coarse Group Delay Adjustment section. |
| Downstream Protection | Off | Used to protect downstream components. See the Downstream Protection section. |

## DAC PLL SETUP

This section explains how to select appropriate LODivMode, RefDivMode, and BCount values in the Device Setup Guide section. These parameters depend on the desired DAC clock frequency ($f_{DAC}$) and DAC reference clock frequency ($f_{REF}$). When using the DAC PLL, the reference clock signal is applied to the REFCLK± differential pins (Pin 3 and Pin 4).

**Table 24. DAC PLL LODivMode Settings**

| DAC Frequency Range (MHz) | LO_DIV_MODE, Register 0x08B, Bits[1:0] |
|---|---|
| 1500 to 2250 | 1 |
| 750 to 1500 | 2 |
| 420 to 750 | 3 |

**Table 25. DAC PLL RefDivMode Settings**

| DAC PLL Reference Frequency ($f_{REF}$) (MHz) | Divide by Factor (RefDivFactor) | REF_DIV_MODE, Register 0x08C, Bits[2:0] |
|---|---|---|
| 80 to 160 | 2 | 1 |
| 160 to 320 | 4 | 2 |
| 320 to 640 | 8 | 3 |
| 640 to 1000 | 16 | 4 |

The VCO frequency ($f_{VCO}$) is related to the DAC clock frequency according to the following equation:

$$f_{VCO} = f_{DAC} \times 2^{LODivMode + 1}$$

where 6 GHz ≤ $f_{VCO}$ ≤ 12 GHz.

BCount must be between 6 and 127 and is calculated based on $f_{DAC}$ and $f_{REF}$ as follows:

$$BCount = \text{floor}((f_{DAC})/(2 \times f_{REF}/RefDivFactor))$$

where $RefDivFactor = 2^{RefDivMode}$ (see Table 25).

Finally, to finish configuring the DAC PLL, set the VCO control registers up as described in Table 73 based on the VCO frequency ($f_{VCO}$).

For more information on the DAC PLL, see the DAC Input Clock Configurations section.

## INTERPOLATION

The transmit path can use zero to three cascaded interpolation filters, which each provide a 2× increase in output data rate and a low-pass function. Table 26 shows the different interpolation modes and the respective usable bandwidth along with the maximum $f_{DATA}$ rate attainable when the power supply is 1.2 V.

**Table 26. Interpolation Modes and Their Usable Bandwidth**

| Interpolation Mode | InterpMode | Usable Bandwidth | Maximum $f_{DATA}$ (MHz) |
|---|---|---|---|
| 1× (bypass) | 0x00 | $f_{DATA}$ | 1238 (JESD204B limited) |
| 2× | 0x01 | 0.4 × $f_{DATA}$ | 1125 |
| 4× | 0x02 | 0.4 × $f_{DATA}$ | 562.5 |
| 8× | 0x03 | 0.4 × $f_{DATA}$ | 281.25 |

The usable bandwidth is defined for 1×, 2×, 4×, and 8× modes 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. For more information, see the Interpolation Filters section.

## JESD204B SETUP

This section explains how to select a JESD204B operating mode for a desired application. This in turn defines appropriate values for CheckSumMode, UnusedLanes, DualLink, CurrentLink, Scrambling, L, F, K, M, N, Np, Subclass, S, HD, Lane0Checksum, and Lanes needed for the Device Setup Guide section.

Note that DualLink, Scrambling, L, F, K, M, N, Np, S, HD, and Subclass must be set the same on the transmit side.

For a summary of how a JESD204B system works and what each parameter means, see the JESD204B Serial Data Interface section.

### *Available Operating Modes*

**Table 27. JESD204B Operating Modes (Single Link)**

| | Mode | | | | | |
|---|---|---|---|---|---|---|
| **Parameter** | **4** | **5** | **6** | **7** | **9** | **10** |
| M (Converter Count) | 2 | 2 | 2 | 2 | 1 | 1 |
| L (Lane Count) | 4 | 4 | 2 | 1 | 2 | 1 |
| S ((Samples per Converter) per Frame) | 1 | 2 | 1 | 1 | 1 | 1 |
| F ((Octets per Frame) per Lane) | 1 | 2 | 2 | 4 | 1 | 2 |

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

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

where *LaneRate* is between 1.44 Gbps and 12.38 Gbps at 1.3 V.

Octets per frame per lane (F) and samples per convertor per frame (S) define how the data is packed. If F = 1, the high density setting must be set to one (HD = 1). Otherwise, set HD = 0.

Converter resolution and bits per sample (N and Np) must both be set to 16. Frames per multiframe (K) must be set to 32 for Mode 4 and Mode 9. Other modes may use either K = 16 or K = 32.

#### Scrambling

Scrambling is a feature that makes the spectrum of the link data independent. This avoids spectral peaking and provides some protection against data dependent errors caused by frequency selective effects in the electrical interface. Set to 1 if scrambling is being used, or to 0 if it is not.

#### Subclass

Subclass determines whether the latency of the device is deterministic, meaning it requires an external synchronization signal. See the Subclass Setup section for more information.

#### CurrentLink

Set CurrentLink to 0 to configure Link 0.

**Lanes**

Use Lanes to enable and deskew particular lanes in two thermometer coded registers.

$$Lanes = (2^L) - 1$$

**UnusedLanes**

UnusedLanes is used to turn off unused circuit blocks to save power. Each physical lane that is not being used (SERDINx±) must be powered off by writing a 1 to the corresponding bit of Register 0x201.

For example, if using Mode 6 in single link mode and sending data on SERDIN0± and SERDIN2±, set UnusedLanes = 0x0A to power off Physical Lane 1 and Physical Lane 3.

**CheckSumMode**

CheckSumMode must match the checksum mode used on the transmit side. If the checksum used is the sum of the fields in the link configuration table, CheckSumMode = 0. If summing the registers containing the packed link configuration fields, CheckSumMode = 1. For more information on the how to calculate the two checksum modes, see the Lane0Checksum section.

**Lane0Checksum**

Lane0Checksum may be used for error checking purposes to ensure that the transmitter is set up as expected.

If CheckSumMode = 0, the checksum is the lower 8 bits of the sum of the L − 1, M − 1, K − 1, N − 1, Np − 1, S − 1, Scrambling, HD, Subclass, and JESDVer variables.

If CheckSumMode = 1, Lane0Checksum is the lower 8 bits of the sum of Register 0x450 to Register 0x45A. Select whether to sum by fields or by registers, matching the setting on the transmitter.

### *DAC Power-Down Setup*

As described in the Step 1: Start Up the DAC section, PdDACs must be set to 0 if both converters are being used. If only one of the converters is being used, the unused converter must be powered down. Table 28 can be used to determine which DAC is powered down based on the number of converters (M) and which converter to use (I DAC or Q DAC).

**Table 28. DAC Power-Down Configuration Settings**

| M (Converters per Link) | DACs to Power Down | | PdCLKs |
|---|---|---|---|
| | I | Q | |
| 1 | 0 | 1 | 0b01 |
| 1 | 1 | 0 | 0b10 |
| 2 | 0 | 0 | 0b00 |

**PdClocks**

If one of the two DACs is powered down, the clock for that DAC can be powered down. If the I DAC is powered down, PdClocks = 0b10. If the Q DAC is powered down, PdClocks = 0b01.

## SERDES CLOCKS SETUP

This section describes how to select the appropriate Halfrate, OvSmp, and PLLDiv settings in the Device Setup Guide section. These parameters depend solely on the lane rate (the lane rate is established in the JESD204B Setup section).

**Table 29. SERDES Lane Rate Configuration Settings**

| Lane Rate (Gbps) | Halfrate | OvSmp | PLLDiv |
|---|---|---|---|
| 1.44 to 3.09 | 0 | 1 | 2 |
| 2.88 to 6.19 | 0 | 0 | 1 |
| 5.75 to 12.38 | 1 | 0 | 0 |

Halfrate and OvSmp set how the clock detect and recover (CDR) circuit samples. See the SERDES PLL section for an explanation of how that circuit blocks works and the role of PLLDiv in the block.

## EQUALIZATION MODE SETUP

Set EqMode = 1 for a low power setting. Select this mode if the insertion loss in your printed circuit board (PCB) is less than 12 dB. For insertion losses greater than 12 dB, but less than 17.5 dB, set EqMode = 0. More details can be found in the Equalization section.

## LINK LATENCY SETUP

This section describes the steps necessary to guarantee multichip deterministic latency in Subclass 1 and guarantee synchronization of links within a device in Subclass 0. Use this section to fill in LMFCDel, LMFCVar, and Subclass in the Device Setup Guide section. For more information, see the Syncing LMFC Signals section.

### *Subclass Setup*

The AD9152 supports JESD204B Subclass 1 and Subclass 0 operation.

**Subclass 1**

Subclass 1 mode achieves deterministic latency and allows the synchronization of links to within the limits listed in Table 7. It requires an external SYSREF± signal that is accurately phase aligned to the DAC clock.

**Subclass 0**

Subclass 0 mode gives deterministic latency to within 4 DAC clock periods. It does not require any signal on the SYSREF± pins (the pins can be left disconnected).

Subclass 0 still requires that all lanes arrive within the same LMFC cycle and the dual DACs must be synchronized to each other (they are synchronized to an internal clock instead of the SYSREF± signal).

Set Subclass to 0 or 1 as desired.

### Link Delay Setup

Use LMFCVar and LMFCDel to impose delays such that all lanes in a system arrive in the same LMFC cycle.

The unit used internally for delays is the period of the internal processing clock (PClock), whose rate is 1/40th of the lane rate. Delays that are not in PClock cycles must be converted before they are used.

Some useful internal relationships are defined below:

*PClock period* = *40/LaneRate*

The PClock period can be used to convert from time to PClock cycles when needed.

*PClockFactor* = 4/F (Frames per PClock)

PClockFactor is used to convert from units of PClock cycles to frame clock cycles, which is needed to set LMFCDel in Subclass 1.

*PClocksPerMF= K/PClockFactor* (PClocks per LMFC cycle)

where *PClocksPerMF* is the number or PClock cycles in a multiframe cycle.

The values for PClockFactor and PClockPerMF are given per JESD204B mode in Table 30.

**Table 30. PClockFactor and PClockPerMF Per JESD204B Mode**

| JESD204B Mode ID | 4 | 5 | 6 | 7 | 9 | 10 |
|---|---|---|---|---|---|---|
| PClockFactor | 4 | 2 | 2 | 1 | 4 | 2 |
| PClockPerMF (K = 32) | 8 | 16 | 16 | 32 | 8 | 16 |
| PClockPerMF (K = 16) | N/A[1] | 8 | 8 | 16 | N/A[1] | 8 |

[1] N/A means not applicable.

#### With Known Delays

With information about all the system delays, LMFCVar and LMFCDel can be calculated directly.

RxFixed (the fixed receiver delay in PClock cycles) and RxVar (the variable receiver delay in PClock cycles) can be found in Table 8. TxFixed (the fixed transmitter delay in PClock cycles) and TxVar (the variable receiver delay in PClock cycles) can be found in the data sheet of the transmitter used. PCBFixed (the fixed PCB trace delay in PClock cycles) can be extracted from the software; because this is generally much smaller than a PClock cycle, it can also be omitted. For both the PCB and transmitter delays, convert the delays into PClock cycles.

For each lane,

*MinDelayLane* = floor(*RxFixed* + *TxFixed* + *PCBFixed*)

*MaxDelayLane* = ceiling(*RxFixed* + *RxVar* + *TxFixed* + *TxVar* + *PCBFixed*))

Across lanes, links, and devices:
*MinDelay* is the minimum of all *MinDelayLane* values.
*MaxDelay* is the maximum of all *MaxDelayLane* values.

For safety, add a guard band of 1 PClock cycle to each end of the link delay as in the following equations:

*LMFCVar* = (*MaxDelay* + 1) − (*MinDelay* − 1)

Note that if LMFCVar must be more than 10, the AD9152 cannot tolerate the variable delay in the system.

For Subclass 1,

*LMFCDel* = ((*MinDelay* − 1) × *PClockFactor*) % *K*

For Subclass 0,

*LMFCDel* = (*MinDelay* − 1) % *PClockPerMF*

Program the same LMFCDel and LMFCVar across all links and devices.

See the Link Delay Setup Example, with Known Delays section for an example calculation.

#### Without Known Delays

If comprehensive delay information is not available or known, the AD9152 can read back the link latency between the $LMFC_{RX}$ and the last arriving LMFC boundary in PClock cycles. This information is then used to calculate LMFCVar and LMFCDel.

For each link (on each device),

1. Power up the board.
2. Follow the steps in Table 15 through Table 23 of the Device Setup Guide.
3. Set the subclass and perform a sync. For one shot sync, perform the writes in Table 31. See the Syncing LMFC Signals section for alternate sync modes.
4. Record DYN_LINK_LATENCY_0 (Register 0x302) as a value of Delay for that link and power cycle.

Repeat Steps 1 to Step 4 twenty times for each device in the system. Keep a single list of the Delay values across all runs and devices.

**Table 31. Register Configuration and Procedure for One Shot Sync**

| Addr. | Bit. No. | Value[1] | Variable | Description |
|---|---|---|---|---|
| 0x301 | | 0x | Subclass | Set the subclass |
| 0x03A | | 0x01 | | Set sync mode to one shot sync |
| 0x03A | | 0x81 | | Enable the sync machine |
| 0x03A | | 0xC1 | | Arm the sync machine |
| SYSREF± | | | | If Subclass = 1, ensure that at least one SYSREF± edge is sent to the device |
| 0x300 | | 0x | | Enable the link |
| | 6 | | ChkSmMd | See the JESD204B Setup section |
| | 0 | | 1 | Enable the link |

[1] 0x denotes a register value that the user must fill in. See the Variable and Description columns for information on selecting the appropriate register value.

The list of delay values is used to calculate LMFCDel and LMFCVar, but first some of the delay values may need to be remapped.

The maximum possible value for DYN_LINK_LATENCY is one less than the number of PClocks in a multiframe (PClocksPerMF). It is possible that a rollover condition may be encountered, meaning the set of recorded Delay values might roll over the edge of a multiframe. If so, Delay values may be near both 0 and PClocksPerMF. If this occurs, add PClocksPerMF to the set of values near 0.

For example, for Delay value readbacks of 6, 7, 0, and 1, the 0 and 1 Delay values must be remapped to 8 and 9, making the new set of Delay values 6, 7, 8, and 9.

Across power cycles, links, and devices.

- MinDelay is the minimum of all Delay measurements.
- MaxDelay is the maximum of all Delay measurements.

For safety, a guard band of 1 PClock cycle is added to each end of the link delay and calculate LMFCVar and LMFCDel with the following equation:

$$LMFCVar = (MaxDelay + 1) - (MinDelay - 1)$$

Note that if LMFCVar must be more than 10, the AD9152 cannot tolerate the variable delay in the system.

For Subclass 1,

$$LMFCDel = ((MinDelay - 1) \times PClockFactor) \% K$$

For Subclass 0,

$$LMFCDel = (MinDelay - 1) \% PClockPerMF$$

Program the same LMFCDel and LMFCVar across all links and devices.

See the Link Delay Setup Example, Without Known Delays section for an example calculation.

## CROSSBAR SETUP

Register 0x308 and Register 0x309 allow arbitrary mapping of physical lanes (SERDINx±) to logical lanes used by the SERDES deframers.

**Table 32. Crossbar Registers**

| Address | Bits | Logical Lane |
|---|---|---|
| 0x308 | [2:0] | LOGICAL_LANE0_SRC |
| 0x308 | [5:3] | LOGICAL_LANE1_SRC |
| 0x309 | [2:0] | LOGICAL_LANE2_SRC |
| 0x309 | [5:3] | LOGICAL_LANE3_SRC |

Write each LOGICAL_LANEx_SRC with the number (x) of the desired physical lane (SERDINx±) from which to receive data. By default, all logical lanes use the corresponding physical lane as their data source. For example, by default LOGICAL_LANE0_SRC = 0, meaning Logical Lane 0 receives data from Physical Lane 0 (SERDIN0±). If instead the user wants to use SERDIN3± as the source for Logical Lane 0, the user must write LOGICAL_LANE0_SRC = 3.

# JESD204B SERIAL DATA INTERFACE

## JESD204B OVERVIEW

The AD9152 has four JESD204B data ports that receive data. The four JESD204B ports can be configured as part of a single JESD204B link. The AD9152 supports single link only.

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 37 shows the communication layers implemented in the AD9152 serial data interface to recover the clock and deserialize, descramble, and deframe the data before it is sent to the digital signal processing section of the device.

The physical layer establishes a reliable channel between the transmitter and the receiver, the data link layer unpacks the data into octets and descrambles the data, and 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, and Scrambling) define how the data is packed and instruct the device how to turn the serial data into samples. These parameters are defined in detail in the Transport Layer section. Only certain combinations of parameters are supported. Each supported combination is called a mode. In total, six single link modes are supported by the AD9152, as described in Table 33, which shows the associated clock rates when the lane rate is 10 Gbps.

Achieving and recovering synchronization of the lanes is very important. To simplify the interface to the transmitter, the AD9152 designates a master synchronization signal for the JESD204B link. $\overline{\text{SYNCOUT}}\pm$ is used as the master signal for all lanes. If any lane in the link loses synchronization, a resynchronization request is sent to the transmitter via the synchronization signal. The transmitter stops sending data and instead sends synchronization characters to all lanes in the link until resynchronization is achieved.

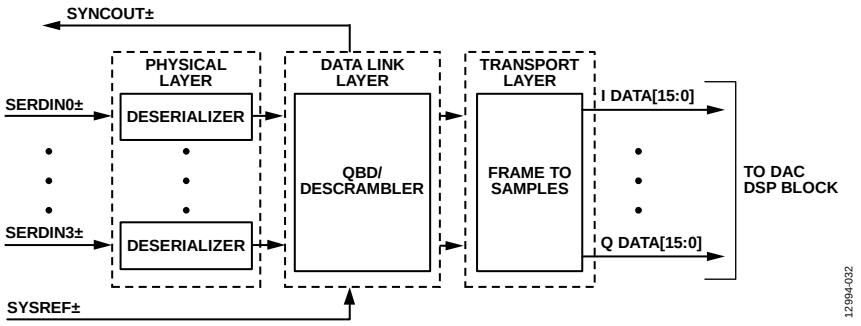


*Figure 37. Functional Block Diagram of Serial Link Receiver*

**Table 33. Single Link JESD204B Operating Modes**

| | Mode | | | | | |
|---|---|---|---|---|---|---|
| **Parameter** | **4** | **5** | **6** | **7** | **9** | **10** |
| M (Converter Counts) | 2 | 2 | 2 | 2 | 1 | 1 |
| L (Lane Counts) | 4 | 4 | 2 | 1 | 2 | 1 |
| S (Samples per Converter per Frame) | 1 | 2 | 1 | 1 | 1 | 1 |
| F (Octets per Frame per Lane) | 1 | 2 | 2 | 4 | 1 | 2 |
| Example Clocks for 10 Gbps Lane Rate | | | | | | |
| PClock (MHz) | 250 | 250 | 250 | 250 | 250 | 250 |
| Frame Clock (MHz) | 1000 | 500 | 500 | 250 | 1000 | 500 |
| Sample Clock (MHz) | 1000 | 1000 | 500 | 250 | 1000 | 500 |

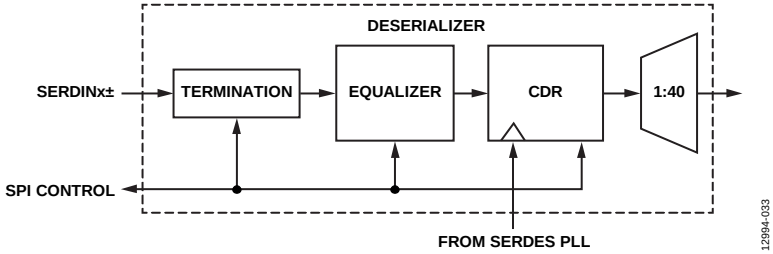


*Figure 38. Deserializer Block Diagram*

## PHYSICAL LAYER

The physical layer of the JESD204B interface, hereafter referred to as the deserializer, has four identical channels. Each channel consists of the terminators, an equalizer, a CDR circuit, and the 1:40 demux function (see Figure 38).

JESD204B data is input to the AD9152 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 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 AD9152 autocalibrates the input termination to 50 Ω. Before running the termination calibration, write to Register 0x2AA and Register 0x2AB as described in Table 34 to guarantee proper calibration. The termination calibration begins when Register 0x2A7, Bit 0 transitions from low to high.

The PHY termination autocalibration routine is as shown in Table 34.

**Table 34. PHY Termination Autocalibration Routine**

| Address | Value | Description |
|---|---|---|
| 0x2AA[1] | 0xB7 | JESD204B interface termination configuration |
| 0x2AB[1] | 0x87 | JESD204B interface termination configuration |
| 0x2A7 | 0x01 | Autotune PHY terminations |

[1] Register 0x2AA and Register 0x2A8 must be set to the recommend value in Table 34 and do not appear in the register map.

The input termination voltage of the DAC is sourced externally via the $V_{TT}$ pins (Pin 13 and Pin 30). Set $V_{TT}$ by connecting it to SVDD12. It is recommended to ac couple the JESD204B inputs to the JESD204B transmit device using 100 nF capacitors.

### *Receiver Eye Mask*

The AD9152 complies with the JESD204B specification regarding the receiver eye mask and is capable of capturing data that complies with this mask. Figure 39 shows the receiver eye mask normalized to the data rate interval with a $V_{TT}$ swing of 600 mV. See the JESD204B specification for more information regarding the eye mask and permitted receiver eye opening.

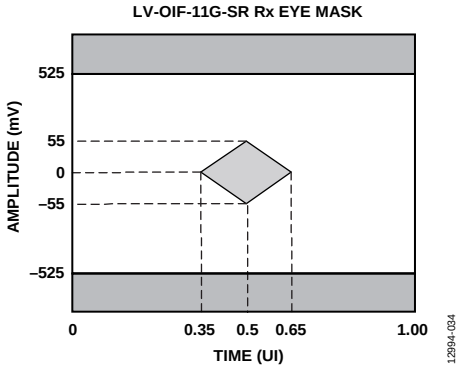


*Figure 39. Receiver Eye Mask*

### *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:

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

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

where:
*M* is the JESD204B parameter for converters per link.
*L* is the JESD204B parameter for lanes per link.

$$ByteRate = LaneRate/10$$

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

$$PClockRate = ByteRate/4$$

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

$$FrameRate = ByteRate/F$$

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

$$PClockFactor = FrameRate/PClockRate = 4/F$$

### *SERDES PLL*

#### Functional Overview of the SERDES PLL

The independent SERDES PLL uses integer-N techniques to achieve clock synthesis. The entire SERDES PLL is integrated on chip, including the VCO and the loop filter. The SERDES PLL VCO operates over the range of 5.75 GHz to 12.38 GHz.

In the SERDES PLL, a VCO divider block divides the VCO clock by 2 to generate a 2.88 GHz to 6.19 GHz quadrature clock for the deserializer cores. This clock is the input to the clock and data recovery block that is described in the Clock and Data Recovery section.

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

**Table 35. SERDES PLL Divider Settings**

| LaneRate (Gbps) | Divide by (DivFactor) | SERDES_PLL_DIV_MODE Register 0x289, Bits[1:0] |
|---|---|---|
| 1.44 to 3.09 | 1 | 2 |
| 2.88 to 6.19 | 2 | 1 |
| 5.75 to 12.38 | 4 | 0 |

Register 0x280 controls the synthesizer enable and recalibration.

To enable the SERDES PLL, first set the PLL divider register according to Table 35, and then enable the SERDES PLL by writing 1 to Register 0x280, Bit 0.

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

### SERDES PLL IRQ

The SERDES PLL lock and lost signals are available as IRQ events. Use Register 0x01F, Bits[4:3] to enable these signals, and then use Register 0x023, Bits[4:3] to read back their statuses and reset the IRQ signals. See the Interrupt Request Operation section for more information.

### SERDES PLL Fixed Register Writes

To optimize the PLL across all operating conditions, the SPI writes shown in Table 16 and Table 77 are required.

These writes properly set up the SERDES PLL, including the loop filter and the charge pump.

Table 36 is a lookup table for SPI register writes included in Table 16 and Table 17.

**Table 36. SERDES PLL VCO Control Lookup Table**

| SERDES PLL VCO Frequency | 0x296 | 0x291 | 0x28A |
|---|---|---|---|
| < 7.15 GHz | 0x02 | 0x49 | 0x7B |
| ≥ 7.15 GHz | 0x03 | 0x4C | 0x2B |

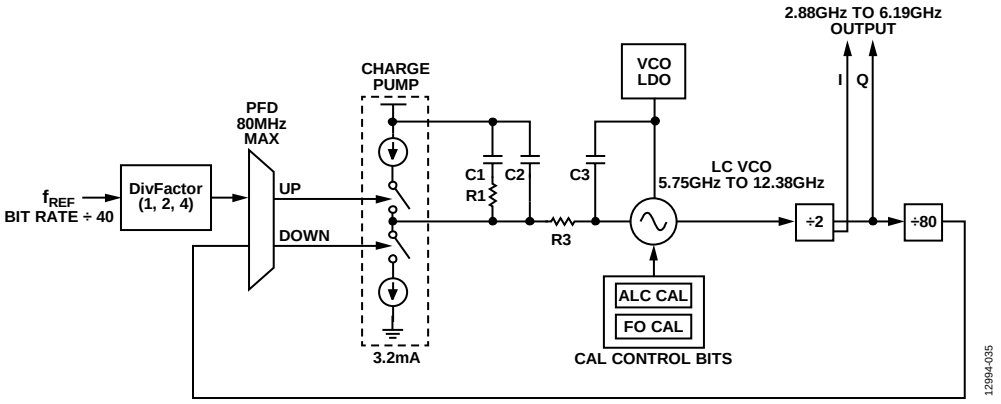


*Figure 40. SERDES PLL Synthesizer Block Diagram Including VCO Divider Block*

### Clock and Data Recovery

The deserializer is equipped with a CDR circuit. Instead of recovering the clock from the JESD204B serial lanes, the CDR acquires the clocks from the SERDES PLL. The 2.88 GHz to 6.19 GHz output from the SERDES PLL, shown in Figure 40, is the input to the CDR.

A CDR sampling mode must be selected to generate the lane rate clock inside the device. If the desired lane rate is greater than 6.19 GHz, half rate CDR operation must be used. If the desired lane rate is less than 6.19 GHz, disable half rate operation. If the lane rate is less than 3.09 GHz, disable half rate and enable 2× oversampling to recover the appropriate lane rate clock. Table 37 gives a breakdown of the CDR sampling settings that must be set dependent on the LaneRate.

**Table 37. CDR Operating Modes**

| LaneRate (Gbps) | ENHALFRATE, Register 0x230, Bit 5 | CDR_OVERSAMP, Register 0x230, Bit 1 |
|---|---|---|
| 1.44 to 3.09 | 0 | 1 |
| 2.88 to 6.19 | 0 | 0 |
| 5.75 to 12.38 | 1 | 0 |

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

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

### Power-Down Unused PHYs

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

### Equalization

To compensate for signal integrity distortions for each PHY channel due to PCB trace length and impedance, the AD9152 employs an easy to use, low power equalizer on each JESD204B channel. The AD9152 equalizers can compensate for insertion losses far greater than required by the JESD204B specification. The equalizers have two modes of operation determined by the EQ_POWER_MODE register setting in Register 0x268, Bits[7:6]. In low power mode (Register 0x268, Bits[7:6] = 2b'01) and operating at the maximum lane rate of 10 Gbps, the equalizer can compensate for up to 12 dB of insertion loss. In normal mode (Register 0x268, Bits[7:6] = 2b'00), the equalizer can compensate for up to 17.5 dB of insertion loss. This performance is shown in Figure 41 as an overlay to the JESD204B specification for insertion loss. Figure 41 shows the equalization performance at 10.0 Gbps, near the maximum baud rate for the AD9152.

Figure 42 and Figure 43 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 on FR4 materials. See the Board Level Hardware Considerations section for specific layout recommendations for the JESD204B channel.

Low power mode is recommended if the insertion loss of the JESD204B PCB channels is less than that of the most lossy supported channel for lower power mode (shown in Figure 41). If the insertion loss is greater than that, but still less than that of the most lossy supported channel for normal mode (shown in Figure 41), use normal mode. At 10 Gbps operation, the equalizer in normal mode consumes about 4 mW more power per lane used than in low power equalizer mode. Note that either mode can be used in conjunction with transmitter preemphasis to ensure functionality and/or optimize for power.

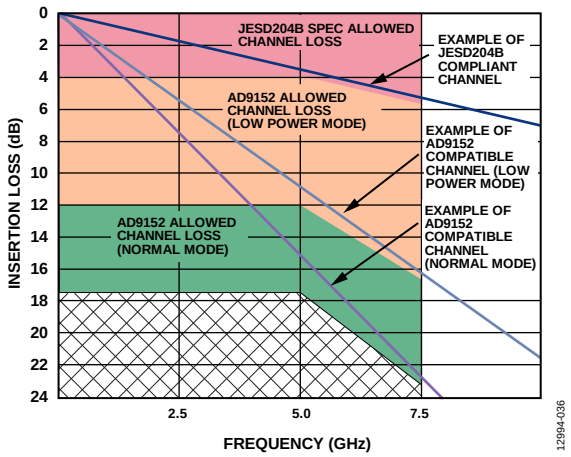


*Figure 41. Insertion Loss Allowed*

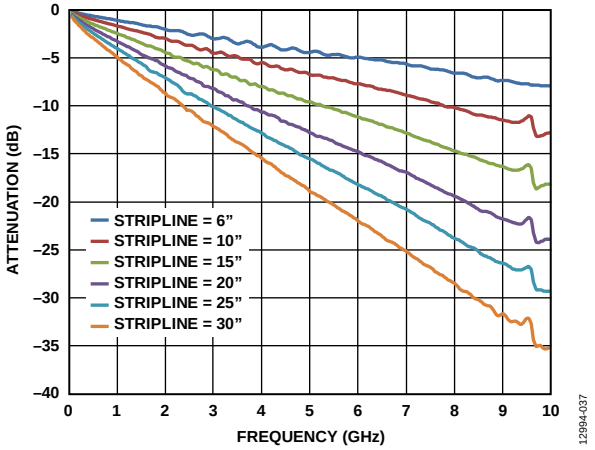


*Figure 42. Insertion Loss of 50 Ω Striplines on FR-4*

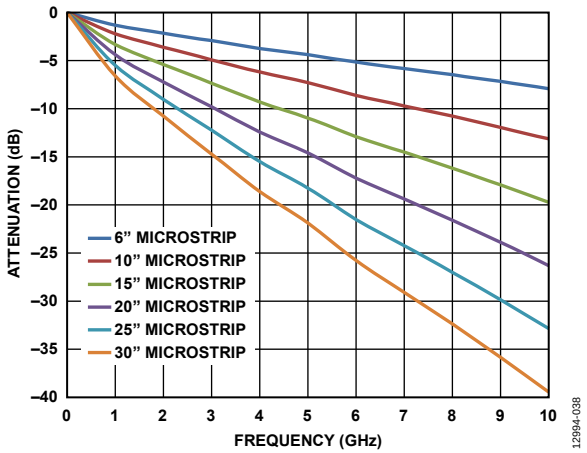


*Figure 43. Insertion Loss of 50 Ω Microstrips on FR-4*

## DATA LINK LAYER

The data link layer of the AD9152 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 44. It consists of a synchronization FIFO for each lane, a crossbar switch, a deframer, and descrambler.

The AD9152 can operate as a single link high speed JESD204B serial data interface. All four lanes of the JESD204B interface handle link layer communications such as code group synchronization, frame alignment, and frame synchronization.

The AD9152 decodes 8-bit/10-bit control characters, allowing marking of the start and end of the frame and alignment between serial lanes. The AD9152 serial interface link can issue a synchronization request by setting the SYNCOUT± signal low. The synchronization protocol follows Section 4.9 of the JESD204B standard. When a stream of four consecutive /K/ symbols is received, the AD9152 deactivates the synchronization request by setting the SYNCOUT± signal high at the next internal LMFC rising edge. Then, the AD9152 waits for the transmitter to issue an initial lane alignment sequence (ILAS). During the ILAS sequence, 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 45).

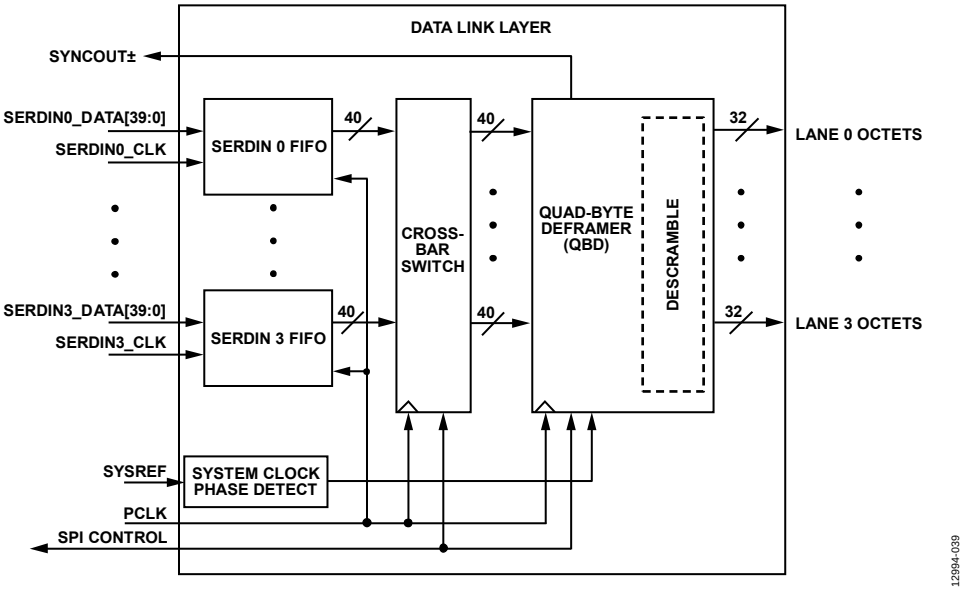


*Figure 44. Data Link Layer Block Diagram*

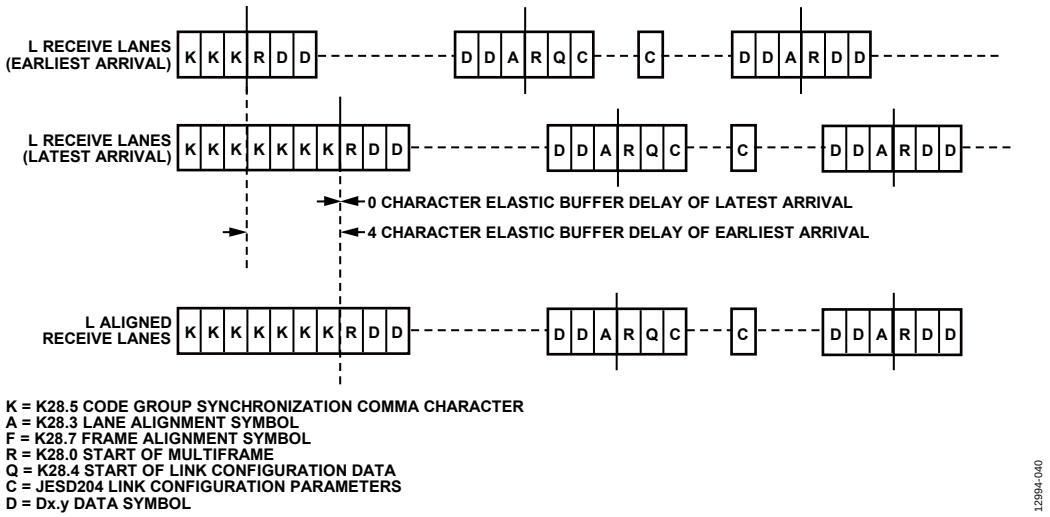


*Figure 45. Lane Alignment During ILAS*

### *JESD204B Serial Link Establishment*

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

**Step 1: Code Group Synchronization**

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

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

**Step 2: Initial Lane Alignment Sequence**

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

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

The ILAS consists of four or more multiframes. The last character of each multiframe is a multiframe alignment character, /A/. The first, third, and fourth multiframes are populated with predetermined data values. Note that Section 8.2 of the JESD204B specifications document describes the data ramp that is expected during ILAS. By default, the AD9152 does not require this ramp. Register 0x47E, Bit 0 can be set high to require the data ramp. The deframer uses the final /A/ of each lane to align the ends of the multiframes within the receiver. The second multiframe contains an R (K.28.0), Q (K.28.4), and then data corresponding to the link parameters. Additional multiframes can be added to the ILAS if needed by the receiver. By default, the AD9152 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.

**Step 3: Data Streaming**

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

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

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

- 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 a few ways (see the JESD204B Error Monitoring section for details).

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

Various test modes for verifying the link integrity can be found in the JESD204B Test Modes section.

### *Lane 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 PClock 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 0x01F, Bit 1 to enable the FIFO error bit, and then use Register 0x023, Bit 1 to read back its status and reset the IRQ signal. See the Interrupt Request Operation section for more information.

### *Crossbar Switch*

Register 0x308 and Register 0x309 allow arbitrary mapping of physical lanes (SERDINx±) to logical lanes used by the SERDES deframers.

**Table 38. Crossbar Registers**

| Address | Bits | Logical Lane |
|---|---|---|
| 0x308 | [2:0] | LOGICAL_LANE0_SRC |
| 0x308 | [5:3] | LOGICAL_LANE1_SRC |
| 0x309 | [2:0] | LOGICAL_LANE2_SRC |
| 0x309 | [5:3] | LOGICAL_LANE3_SRC |

Write each LOGICAL_LANEy_SRC with the number (x) of the desired physical lane (SERDINx±) from which to receive data. By default, all logical lanes use the corresponding physical lane as their data source. For example, by default LOGICAL_LANE0_SRC = 0, thus Logical Lane 0 receives data from Physical Lane 0 (SERDIN0±). If instead the user wants to use SERDIN3± as the source for Logical Lane 0, the user must write LOGICAL_LANE0_SRC = 3.

### *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 AD9152 consists of one quad byte deframer (QBD). The deframer takes in 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 (PClock) cycle.

The deframer uses the JESD204B parameters that the user has programmed into the register map to identify how the data has been packed and unpack it. The JESD204B parameters are discussed 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 AD9152 provides an optional descrambler block using a self synchronous descrambler with a polynomial: $1 + x^{14} + x^{15}$.

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

### *Syncing LMFC Signals*

The first step in guaranteeing synchronization across devices begins with syncing the LMFC signals. The I DAC and Q DAC share one LMFC signal. In Subclass 0, the LMFC signal is synchronized to an internal processing clock. In Subclass 1, all LMFC signals for all devices are synchronized to an external 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 an active high 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 a AD9516-1 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 AD9152 supports a single pulse or step, or 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 V to 2 V) or ac-coupled. When dc-coupled, a small amount of common-mode current (<500 μA) is drawn from the SYSREF± pins. See Figure 46 for the SYSREF± internal circuit.

To avoid this common-mode current draw, use a 50% duty-cycle periodic SYSREF± signal with ac coupling capacitors. If ac-coupled, the ac coupling capacitors combine with the resistors shown in Figure 46 to make a high-pass filter with an RC time constant of $\tau = RC$. Select C such that $\tau > 4/$ SYSREF± frequency.

In addition, the edge rate must be sufficiently fast—at least 6.3 V/ns is recommended per Table 5—to meet the SYSREF± vs. DAC clock keep out window (KOW) requirements.

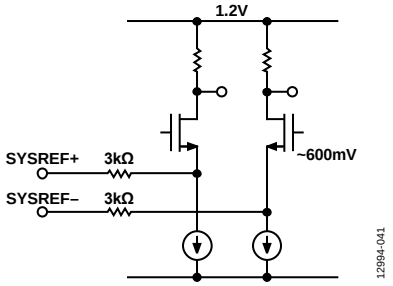


*Figure 46. SYSREF± Input Circuit*

### Sync Processing Modes Overview

The AD9152 supports various LMFC sync processing modes. These modes are one shot, continuous, windowed continuous, and monitor modes. All sync processing modes perform a phase check to see that the LMFC is phase aligned to an alignment edge. In Subclass 1, the SYSREF± pulse acts as the alignment edge; in Subclass 0, an internal processing clock acts as the alignment edge. If the signals are not in phase, a clock rotation occurs to align the signals. The sync modes are described in the following sections. See the Sync Procedure section for details on the procedure for syncing the LMFC signals.

### One Shot Sync Mode (SYNCMODE = 0x1)

In one shot sync mode, a phase check occurs on only the first alignment edge that is received after the sync machine is armed. If the phase error is larger than a specified window error tolerance, a phase adjustment occurs. Though an LMFC synchronization occurs only once, the SYSREF± signal can still be continuous.

### Continuous Sync Mode (SYNCMODE = 0x2)

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

Continuous mode differs from the one shot mode in two ways. First, no SPI cycle is required to arm the device; the alignment edge seen after continuous mode is enabled results in a phase check. Second, a phase check (and when necessary, clock rotation) occurs on every alignment edge in continuous mode. The one caveat to the previous statement is that when a phase rotation cycle is underway, subsequent alignment edges are ignored until the logic lane is ready again.

The maximum acceptable phase error (in DAC clock cycles) between the alignment edge and the LMFC edge is set in the error window tolerance register. If continuous sync mode is used with a nonzero error window tolerance, a phase check occurs on every SYSREF± pulse, but an alignment occurs only if the phase error is greater than the specified error window tolerance. If the jitter of the SYSREF signal violates the KOW specification given in Table 5 and therefore causes phase error uncertainty, the error tolerance can be increased to avoid constant clock rotations. Note that this means that the latency is less deterministic by the size of the window. If the error window tolerance must be set above 3, Subclass 0 with one shot sync is recommended.

For debug purposes, SYNCARM (Register 0x03A, Bit 6) can be used to inform the user that alignment edges are being received in continuous mode. Because the SYNCARM bit is self cleared after an alignment edge is received, the user can arm the sync (SYNCARM (Register 0x03A, Bit 6) = 1), and then read back SYNCARM. If SYNCARM = 0, the alignment edges are being received and phase checks are occurring. Arming the sync machine in this mode does not affect the operation of the device.

### One Shot Then Monitor Sync Mode (SYNCMODE = 0x9)

In one shot then monitor mode, the user can monitor the phase error in real time. Use this sync mode with a periodic SYSREF± signal. A phase check and alignment occurs on the first alignment edge received after the sync machine is armed. On all subsequent alignment edges, the phase is monitored and reported, but no clock phase adjustment occurs.

The phase error can be monitored on the SYNC_CURRERR register (Register 0x03C and Register 0x03D). Immediately after an alignment occurs, SYNC_CURRERR is forced to 0 to indicate that there is no difference between the alignment edge and the LMFC edge. On every subsequent alignment edge, the phase is checked. If the alignment is lost, the phase error is reported in the SYNC_CURRERR_L register in DAC clock cycles. If the phase error is beyond the selected window tolerance (Register 0x034, Bits[2:0]), Bit 6 or Bit 7 of Register 0x03D is set high depending on whether the phase error is on low or high side.

When an alignment occurs, snapshots of the last phase error (Register 0x03C) and the corresponding error flags (Register 0x03D, Bits[7:6]) are placed into readable registers for reference (Register 0x038 and Register 0x039, respectively).

### Sync Procedure

The procedure for enabling the sync is as follows:

1. Set the desired sync processing mode. The sync processing mode settings are listed in Table 39.
2. For Subclass 1, set the error window according to the uncertainty of the SYSREF± signal relative to the DAC clock and the tolerance of the application for deterministic latency uncertainty. The sync window tolerance settings are given in Table 40.
3. Enable sync by writing 1 to SYNCENABLE (Register 0x03A, Bit 7).
4. If in one shot mode, arm the sync machine by writing 1 to SYNCARM (Register 0x03A, Bit 6).
5. If in Subclass 1, ensure that at least one SYSREF± pulse is sent to the device.
6. Check the status by reading the following bit fields:
   a) SYNCBUSY (Register 0x03B, Bit 7) = 0 to indicate that the sync logic is no longer busy.
   b) SYNCLOCK (Register 0x03B, Bit 3) = 1 to indicate that the signals are aligned. This bit updates on every phase check.
   c) SYNCWLIM (Register 0x03B, Bit 1) = 0 to indicate that the phase error is not beyond the specified error window. This bit updates on every phase check.
   d) SYNCROTATE (Register 0x03B, Bit 2) = 1 if the phases were not aligned before the sync and a clock alignment occurred. This bit is sticky and can be cleared only by writing to SYNCCLRSTKY control bit (Register 0x03A, Bit 5).
   e) SYNCTRIP (Register 0x03B, Bit 0) = 1 to indicate that the alignment edge was received and the phase check occurred. This bit is sticky and can be cleared only by writing to SYNCCLRSTKY control bit (Register 0x03A, Bit 5).

**Table 39. Sync Processing Modes**

| Sync Processing Mode | SYNCMODE (Register 0x03A, Bits[3:0]) |
|---|---|
| One Shot | 0x01 |
| Continuous | 0x02 |
| One Shot Then Monitor | 0x09 |

**Table 40. Sync Window Tolerance**

| Sync Error Window Tolerance (DAC Clock Cycles) | ERRWINDOW (Register 0x034, Bits[2:0]) |
|---|---|
| ±½ | 0x00 |
| ±1 | 0x01 |
| ±2 | 0x02 |
| ±3 | 0x03 |
| ±4 | 0x04 |
| ±5 | 0x05 |
| ±6 | 0x06 |
| ±7 | 0x07 |

### LMFC Sync IRQ

The sync status bits (SYNCLOCK, SYNCROTATE, SYNCTRIP, and SYNCWLIM) are available as IRQ events.

Use Register 0x021, Bits[3:0] to enable the sync status bits, and then use Register 0x025, Bits[3:0] to read back their statuses and to reset the IRQ signals.

See the Interrupt Request Operation section for more information.

## *Deterministic Latency*

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

The AD9152 supports JESD204B Subclass 0 and Subclass 1 operation, but not Subclass 2. Write the subclass to Register 0x301, Bits[2:0] and once per link to Register 0x458, Bits[7:5].

### Subclass 0

Subclass 0 mode does not require any signal on the SYSREF± pins, which can be left disconnected.

Subclass 0 still requires that all lanes arrive within the same LMFC cycle.

### Minor Subclass 0 Caveats

Because the AD9152 requires an ILAS, the nonmultiple converter single lane (NMCDA-SL) case from the JESD204A specification is supported only when using the optional ILAS.

Error reporting using $\overline{\text{SYNCOUT}}$± is not supported when using Subclass 0 with F = 1.

### Subclass 1

Subclass 1 mode gives deterministic latency and allows links to be synced to within ½ a DAC clock period. It requires an external SYSREF± signal that is accurately phase aligned to the DAC clock.

### Deterministic Latency Requirements

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

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

### *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 49.

For proper functioning, all lanes on the 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 AD9152, this is not necessarily the case; instead, the AD9152 uses a local LMFC for the link ($LMFC_{Rx}$) that can be delayed from the SYSREF± aligned LMFC. Because the LMFC is periodic, this 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 AD9152 can achieve proper performance with a smaller total latency. Figure 47 and Figure 48 show a case where the link delay is larger than an LMFC period. Note that it can be accommodated by delaying $LMFC_{Rx}$.

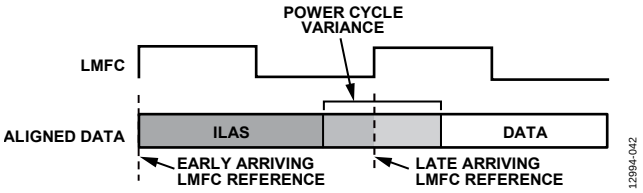


*Figure 47. Link Delay > LMFC Period Example*

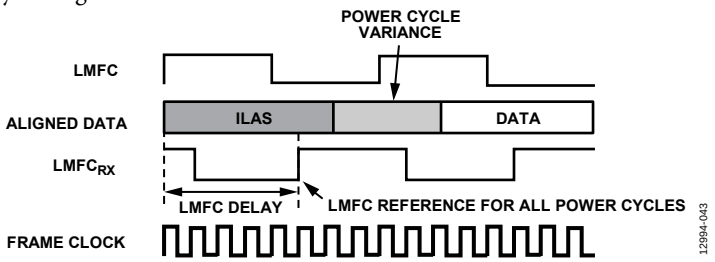


*Figure 48. LMFC_DELAY_x to Compensate for Link Delay > LMFC*

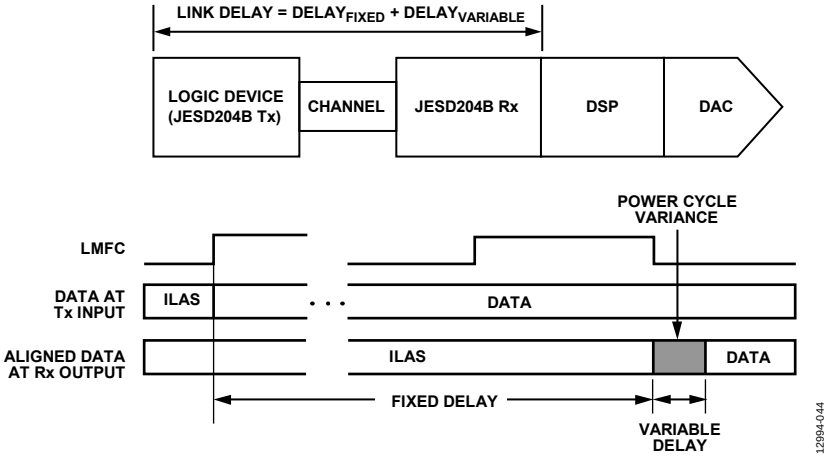


*Figure 49. JESD204B Link Delay = Fixed Delay + Variable Delay*

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

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

The RBD described in the JESD204B specification takes values from one frame clock cycle to K frame clock cycles, while the RBD of the AD9152 takes values from 0 PClock cycles to 10 PClock cycles. As a result, up to 10 PClock cycles of total delay variation can be absorbed. Because LMFCVar is in PClock cycles, and LMFCDel is in frame clock cycles, a conversion between these two units is needed. The PClockFactor, or number of frame clock cycles per PClock 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 both Register 0x306 for all devices in the system.

### Link Delay Setup Example, with Known Delays

All the known system delays can be used to calculate LMFCVar and LMFCDel as described in the Link Delay Setup section.

The example shown in Figure 50 is demonstrated in the following steps according to the procedure outlined in the Link Delay Setup section. Note that this example is in Subclass 1 to achieve deterministic latency, which has a PClockFactor (4/F) of two frame clock cycles per PClock cycle, and uses K = 32 (frames/multiframe). Because PCBFixed << PClockPeriod, PCBFixed is negligible in this example and not included in the calculations.

1. Find the receiver delays using Table 8.
   *RxFixed* = 17 PClock cycles
   *RxVar* = 2 PClock cycles
2. Find the transmitter delays. The equivalent table in the example JESD204B core (implemented on a GTH or GTX transceiver on a Virtex-6 FPGA) states that the delay is 56 ± 2 byte clock cycles.
   Because the PClockRate = ByteRate/4 as described in the Clock Relationships section, the transmitter delays in PClock cycles are as follows:
   *TxFixed* = 54/4 = 13.5 PClock cycles
   *TxVar* = 4/4 = 1 PClock cycle
3. Calculate MinDelayLane as follows:
   *MinDelayLane* = floor(*RxFixed* + *TxFixed* + *PCBFixed*)
   = floor(17 + 13.5 + 0)
   = floor(30.5)
   *MinDelayLane* = 30
4. Calculate MaxDelayLane as follows:
   *MaxDelayLane* = ceiling(*RxFixed* + *RxVar* + *TxFixed* + *TxVar* + *PCBFixed*))
   = ceiling(17 + 2 + 13.5 + 1 + 0)
   = ceiling(33.5)
   *MaxDelayLane* = 34
5. Calculate LMFCVar as follows:
   *LMFCVar* = (*MaxDelay* + 1) − (*MinDelay* − 1)
   = (34 + 1) − (30 − 1) = 35 − 29
   *LMFCVar* = 6 PClock cycles
6. Calculate LMFCDel as follows:
   *LMFCDel* = ((*MinDelay* − 1) × *PClockFactor*) % *K*
   = ((30 − 1) × 2) % 32 = (29 × 2) % 32
   = 58 % 32
   *LMFCDel* = 26 frame clock cycles
7. Write LMFCDel to Register 0x304 for all devices in the system. Write LMFCVar to Register 0x306 for all devices in the system.

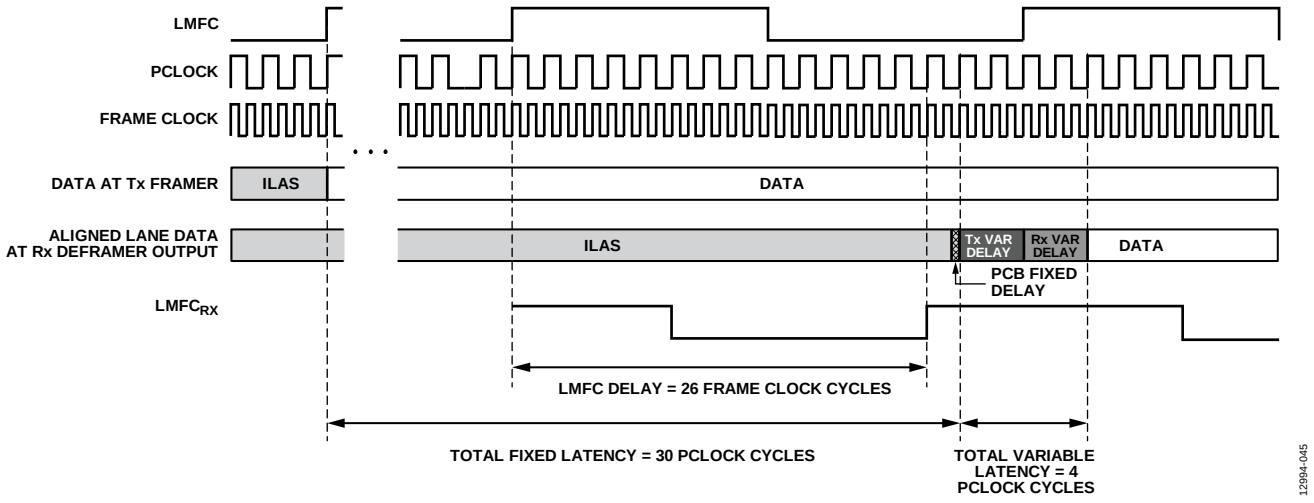


*Figure 50. LMFC Delay Calculation Example*

### Link Delay Setup Example, Without Known Delays

If the system delays are not known, the AD9152 can read back the link latency between $LMFC_{RX}$ for each link and the SYSREF± aligned LMFC. This information is then used to calculate LMFCVar and LMFCDel, as shown in the Without Known Delays section.

Figure 52 shows how DYN_LINK_LATENCY_0 (Register 0x302) provides a readback showing the delay (in PClock cycles) between $LMFC_{RX}$ and the transition from ILAS to the first data sample. By repeatedly power-cycling and taking this measurement, the minimum and maximum delays across power cycles can be determined and used to calculate LMFCVar and LMFCDel.

The example shown in Figure 52 is demonstrated in the following steps according to the procedure outlined in the Without Known Delays section. Note that this example is in Subclass 1 to achieve deterministic latency, which has a PClockFactor (frame clock rate/PClockRate) of 2 and uses K = 16; therefore PClocksPerMF = 8.

1. In Figure 52, for Link A, Link B, and Link C, the system containing the AD9152 (including the transmitter) is power cycled and configured 20 times. The AD9152 is configured as described in the Device Setup Guide. Because the point of this exercise is to determine LMFCDel and LMFCVar, the LMFCDel is programmed to 0 and DYN_LINK_LATENCY_0 is read from Register 0x302.
   The variation in the link latency over the 20 runs is shown in Figure 52 in gray.
   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 K/PClockFactor = 8. Add PClocksPerMF = 8 to the low set. 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.
2. Calculate the minimum of all delay measurements across all power cycles, links, and devices:
   *MinDelay* = min(all *Delay* values) = 4
3. Calculate the maximum of all delay measurements across all power cycles, links, and devices:
   *MaxDelay* = max(all *Delay* values) = 9
4. Calculate the total delay variation (with guard band) across all power cycles, links, and devices:
   $LMFCVar = (MaxDelay + 1) - (MinDelay - 1)$
   $= (9 + 1) - (4 - 1) = 10 - 3 = 7$ PClock cycles
5. Calculate the minimum delay in frame clock cycles (with guard band) across all power cycles, links, and devices:
   $LMFCDel = ((MinDelay - 1) \times PClockFactor) \% K$
   $= ((4 - 1) \times 2) \% 16 = (3 \times 2) \% 16$
   $= 6 \% 16 = 6$ frame clock cycles
6. Write LMFCDel to Register 0x304 for all devices in the system. Write LMFCVar to Register 0x306 for all devices in the system.

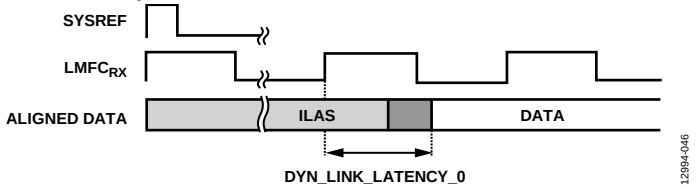


*Figure 51. DYN_LINK_LATENCY_0 Example*

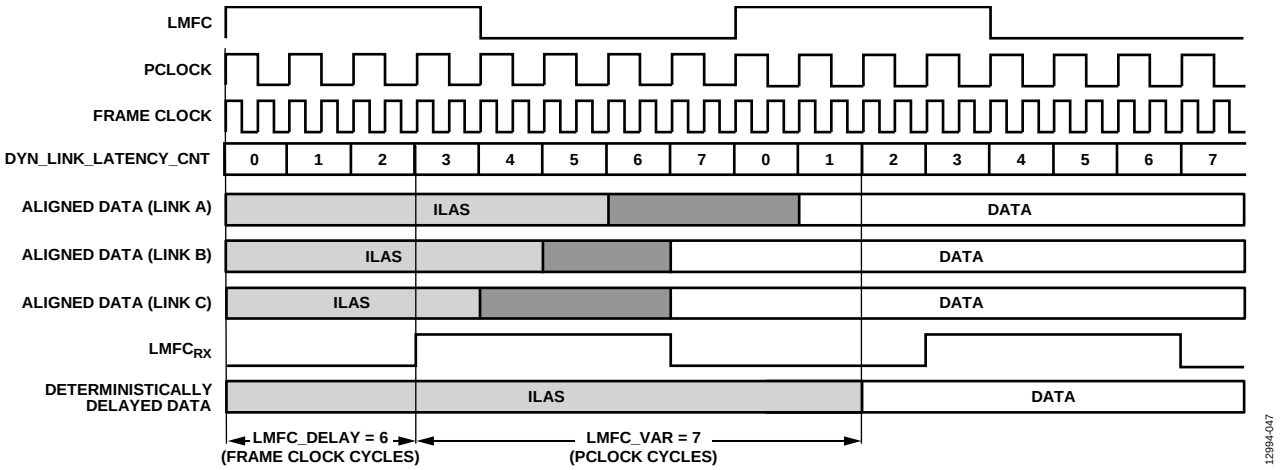


*Figure 52. Multilink Synchronization Settings, Derived Method Example*

## 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 41. A number of device parameters are defined in Table 42.

**Table 41. JESD204B Transport Layer Parameters**

| Parameter | Description |
|---|---|
| F | Number of octets per frame per lane: 1, 2, or 4. |
| K | Number of frames per multiframe.<br>K = 32 if F = 1, K = 16 or 32 otherwise. |
| L | Number of lanes per converter device (per link), as follows:<br>1, 2, or 4 (single link mode). |
| M | Number of converters per device (per link), as follows:<br>1 or 2 (single link mode). |
| S | Number of samples per converter, per frame: 1 or 2. |

**Table 42. JESD204B Device Parameters**

| Parameter | Description |
|---|---|
| CF | Number of control words per device clock per link. Not supported, must be 0. |
| CS | Number of control bits per conversion sample. Not supported, must be 0. |
| HD | High density user data format. Used when samples must be split across lanes.<br>Set to 1 when F = 1, otherwise 0. |
| N | Converter resolution = 16. |
| N' (or NP) | Total number of bits per sample = 16. |

Certain combinations of these parameters, called JESD204B operating modes, are supported by the AD9152. See Table 43 for a list of supported modes, along with their associated clock relationships.

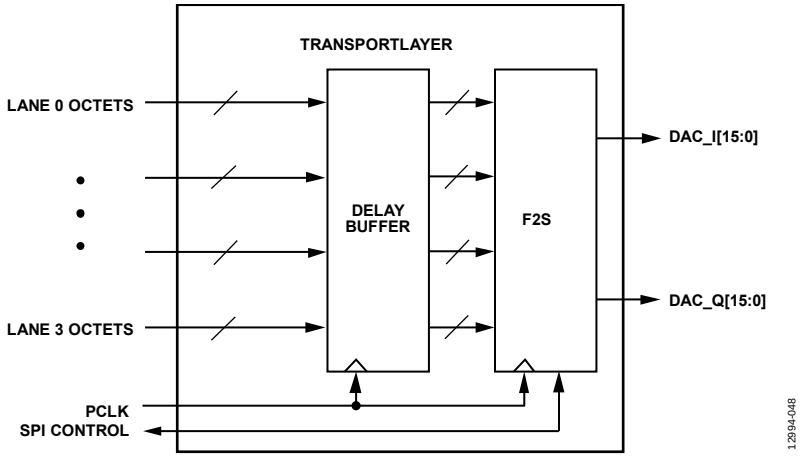


*Figure 53. Transport Layer Block Diagram*

**Table 43. Single Link JESD204B Operating Modes**

| | Mode | | | | | |
|---|---|---|---|---|---|---|
| **Parameter** | **4** | **5** | **6** | **7** | **9** | **10** |
| M (Converter Count) | 2 | 2 | 2 | 2 | 1 | 1 |
| L (Lane Count) | 4 | 4 | 2 | 1 | 2 | 1 |
| S (Samples per Converter per Frame) | 1 | 2 | 1 | 1 | 1 | 1 |
| F (Octets per Frame, per Lane) | 1 | 2 | 2 | 4 | 1 | 2 |
| K[1] (Frames per Multiframe) | 32 | 16/32 | 16/32 | 16/32 | 32 | 16/ 32 |
| HD (High Density) | 1 | 0 | 0 | 0 | 1 | 0 |
| N (Converter Resolution) | 16 | 16 | 16 | 16 | 16 | 16 |
| NP (Bits per Sample) | 16 | 16 | 16 | 16 | 16 | 16 |
| Example Clocks for 10 Gbps Lane Rate | | | | | | |
| PClock Rate (MHz) | 250 | 250 | 250 | 250 | 250 | 250 |
| FrameClock Rate (MHz) | 1000 | 500 | 500 | 250 | 1000 | 500 |
| Data Rate (MHz) | 1000 | 1000 | 500 | 250 | 1000 | 500 |

[1] K must be 32 in Mode 4 and Mode 9. It can be 16 or 32 in all other modes.

### Configuration Parameters

The AD9152 modes refer to the link configuration parameters for L, K, M, N, NP, S, and F. Table 44 provides the description and addresses for these settings.

**Table 44. Configuration Parameters**

| JESD204B Setting | Description | Address |
|---|---|---|
| L – 1 | Number of lanes – 1. | 0x453[4:0] |
| F – 1 | Number of ((octets per frame) per lane) – 1. | 0x454[7:0] |
| K – 1 | Number of frames per multiframe – 1. | 0x455[4:0] |
| M – 1 | Number of converters – 1. | 0x456[7:0] |
| N – 1 | Converter bit resolution – 1. | 0x457[4:0] |
| NP – 1 | Bit packing per sample – 1. | 0x458[4:0] |
| S – 1 | Number of ((samples per converter) per frame) – 1. | 0x459[4:0] |
| HD | High density format. Set to 1 if F = 1. Leave at 0 if F ≠ 1. | 0x45A[7] |
| F[1] | F parameter, in ((octets per frame) per lane). | 0x476[7:0] |
| DID | Device ID. Match the device ID sent by the transmitter. | 0x450[7:0] |
| BID | Bank ID. Match the bank ID sent by the transmitter. | 0x451[3:0] |
| LID0 | Lane ID for Lane 0. Match the lane ID sent by the transmitter on Logical Lane 0. | 0x452[4:0] |
| JESDV | JESD version. Match the version sent by the transmitter (0x0 = JESD204A, 0x1 = JESD204B). | 0x459[7:5] |

[1] The values that need to be written in Register 0x454 and Register 0x476 are different, F – 1 and F, respectively.

### 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. Figure 54 shows a detailed flow of the data through the various hardware blocks for Mode 4 (L = 4, M = 2, S = 1, F = 1). Simplified flow diagrams for all other modes are shown in Figure 54 through Figure 59.

### Single Link Configuration

The AD9152 uses the settings contained in Table 43. Mode 4 to Mode 10, except Mode 8, can be used for single link operation.

### Checking Proper Configuration

As a convenience, the AD9152 provides quick configuration checks. Register 0x030, Bit 5 is high if an illegal LMFC_DELAY_0 value is used. Register 0x030, Bit 3 is high if an unsupported combination of L, M, F, or S is used. Register 0x030, Bit 2 is high if an illegal K character is used. Register 0x030, Bit 1 is high if an illegal SUBCLASSV is used.

### Deskewing and Enabling Logical Lanes

After proper configuration, the logical lanes must be deskewed and enabled to capture data.

Set Bit x in Register 0x46C to 1 to deskew Logical Lane x and to 0 if that logical lane is not being used. Then, set Bit x in Register 0x47D to 1 to enable Logical Lane x and to 0 if that logical lane is not being used.

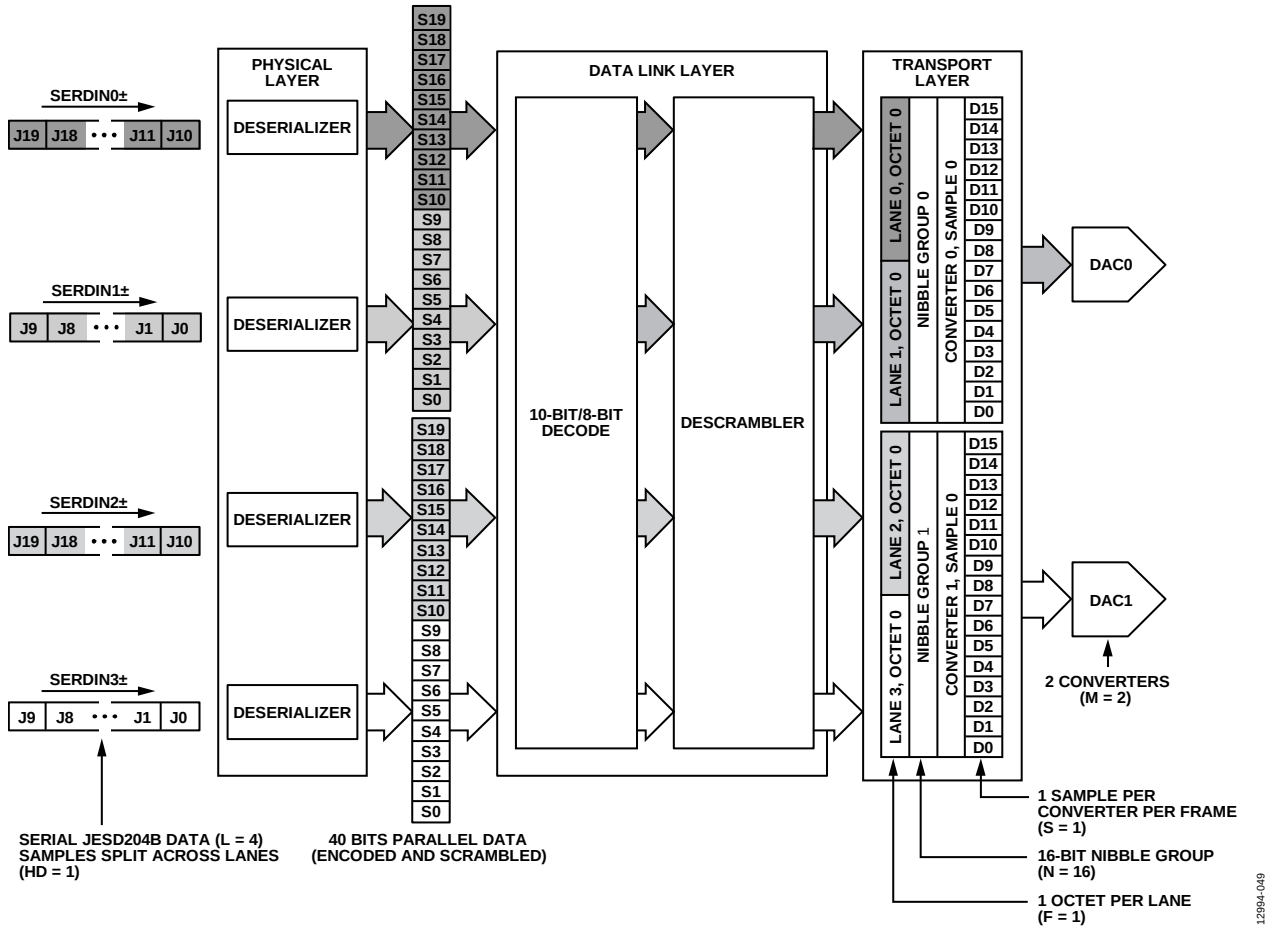


*Figure 54. JESD204B Mode 4 Data Deframing*

### *Mode Configuration Maps*

Table 45 to Table 50 contain the SPI configuration map for each mode shown in Figure 54 through Figure 59. Figure 54 through Figure 59 show the associated data flow through the deframing process of the JESD204B receiver for each of the modes. Mode 4 to Mode 10, except Mode 8, apply to single link operation.

Additional details regarding all the SPI registers can be found in the Register Map and Descriptions section.

**Table 45. SPI Configuration Map—Register Settings for JESD204B Parameters for Mode 4**

| Addr. | Setting | Description |
|---|---|---|
| 0x453 | 0x03 or 0x83 | Register 0x453, Bit 7 = 0 or 1: scrambling disabled or enabled; Register 0x453, Bits[4:0] = 0x3: L = 4 lanes per link |
| 0x454 | 0x00 | Register 0x454, Bits[7:0] = 0x00: F = 1 octet per frame per lane |
| 0x455 | 0x1F | Register 0x455, Bits[4:0] = 0x1F: K = 32 frames per multiframe |
| 0x456 | 0x01 | Register 0x456, Bits[7:0] = 0x01: M = 2 converters per link |
| 0x457 | 0x0F | Register 0x457, Bits[7:6] = 0x0: always set CS = 0; Register 0x457, Bits[4:0] = 0x0F: N = 16, always set to 16-bit resolution |
| 0x458 | 0x0F or 0x2F | Register 0x458, Bits[7:5] = 0x0 or 0x1: Subclass 0 or Subclass 1, Register 0x458, Bits[4:0] = 0xF: NP = 16 bits per sample |
| 0x459 | 0x20 | Register 0x459, Bits[7:5] = 0x1: set to JESD204B version, Register 0x459, Bits[4:0] = 0x0: S = 1 sample per converter per frame |
| 0x45A | 0x80 | Register 0x45A, Bit 7 = 1: HD = 1; Register 0x45A, Bits[4:0] = 0x00: always set CF = 0 |
| 0x46C | 0x0F | Register 0x46C, Bits[7:0] = 0x0F: deskew Lane 0 to Lane 3 |
| 0x476 | 0x01 | Register 0x476, Bits[7:0] = 0x01: F = 1 octet per frame |
| 0x47D | 0x0F | Register 0x47D, Bits[7:0] = 0x0F: enable Lane 0 to Lane 3 |

See Figure 54 for an illustration of the AD9152 JESD204B Mode 4 data deframing process.

**Table 46. SPI Configuration Map—Register Settings for JESD204B Parameters for Mode 5**

| Addr. | Setting | Description |
|---|---|---|
| 0x453 | 0x03 or 0x83 | Register 0x453, Bit 7 = 0 or 1: scrambling disabled or enabled; Register 0x453, Bits[4:0] = 0x3: L = 4 lanes per converter |
| 0x454 | 0x01 | Register 0x454, Bits[7:0] = 0x01: F = 2 octets per frame per lane |
| 0x455 | 0x0F or 0x1F | Register 0x455, Bits[4:0] = 0x0F or 0x1F: K = 16 or 32 frames per multiframe |
| 0x456 | 0x01 | Register 0x456, Bits[7:0] = 0x01: M = 2 converters per link |
| 0x457 | 0x0F | Register 0x457, Bits[7:6] = 0x0: always set CS = 0; Register 0x457, Bits[4:0] = 0x0F: N = 16, always set to 16-bit resolution |
| 0x458 | 0x0F or 0x2F | Register 0x458, Bits[7:5] = 0x0 or 0x1: Subclass 0 or Subclass 1, Register 0x458, Bits[4:0] = 0xF: NP = 16 bits per sample |
| 0x459 | 0x21 | Register 0x459, Bits[7:5] = 0x1: set to JESD204B version, Register 0x459, Bits[4:0] = 0x1: S = 2 samples per converter per frame |
| 0x45A | 0x00 | Register 0x45A, Bit 7 = 0: HD = 0; Register 0x45A, Bits[4:0] = 0x00: always set CF = 0 |
| 0x46C | 0x0F | Register0x46C[7:0] = 0x0F: deskew Lane 0 to Lane 3 |
| 0x476 | 0x02 | Register 0x476, Bits[7:0] = 0x02: F = 2 octets per frame |
| 0x47D | 0x0F | Register 0x47D, Bits[7:0] = 0x0F: 4 lanes enabled, set one bit per lane to enable |

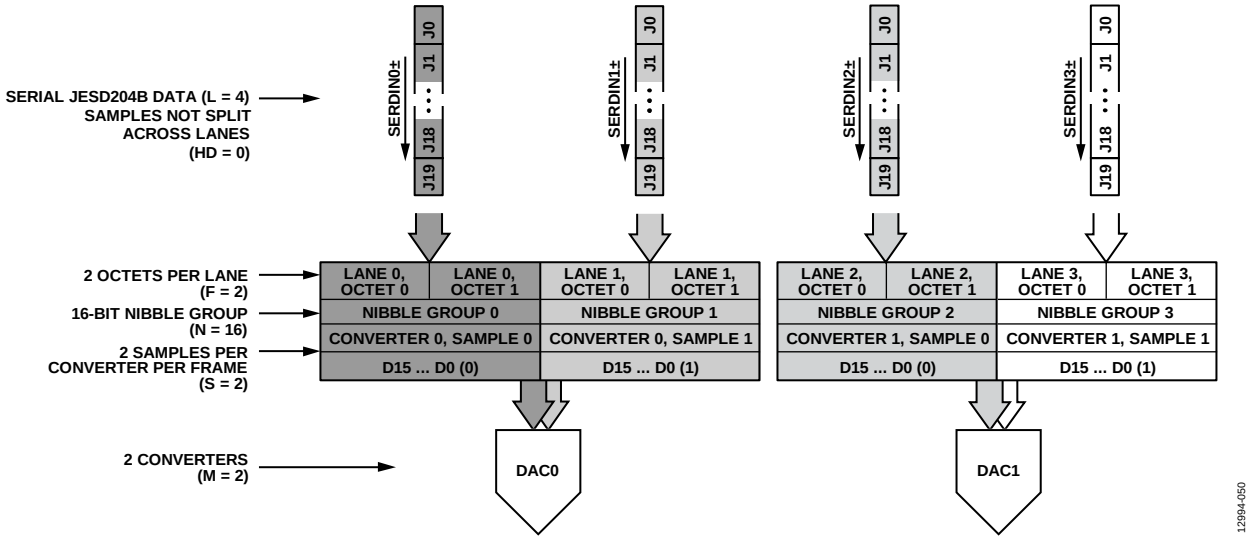


*Figure 55. JESD204B Mode 5 Data Deframing*

**Table 47. SPI Configuration Map—Register Settings for JESD204B Parameters for Mode 6**

| Address | Setting | Description |
|---|---|---|
| 0x453 | 0x01 or 0x81 | Register 0x453, Bit 7 = 0 or 1: scrambling disabled or enabled, Register 0x453, Bits[4:0] = 0x1: L = 2 lanes per link |
| 0x454 | 0x01 | Register 0x454, Bits[7:0] = 0x01: F = 2 octets per frame per lane |
| 0x455 | 0x0F or 0x1F | Register 0x455, Bits[4:0] = 0x0F or 0x1F: K = 16 or 32 frames per multiframe |
| 0x456 | 0x01 | Register 0x456, Bits[7:0] = 0x01: M = 2 converters per link |
| 0x457 | 0x0F | Register 0x457, Bits[7:6] = 0x0: always set CS = 0; Register 0x457, Bits[4:0] = 0x0F: N = 16, always set to 16-bit resolution |
| 0x458 | 0x0F or 0x2F | Register 0x458, Bits[7:5] = 0x0 or 0x1: Subclass 0 or Subclass 1, Register 0x458, Bits[4:0] = 0xF: NP = 16 bits per sample |
| 0x459 | 0x20 | Register 0x459, Bits[7:5] = 0x1: set to JESD204B version, Register 0x459, Bits[4:0] = 0x0: S = 1 sample per converter per frame |
| 0x45A | 0x00 | Register 0x45A, Bit 7 = 0: HD = 0; Register 0x45A, Bits[4:0] = 0x00: always set CF = 0 |
| 0x46C | 0x03 | Register 0x46C, Bits[7:0] = 0x03: deskew Lane 0 and Lane 1 |
| 0x476 | 0x02 | Register 0x476, Bits[7:0] = 0x02: F = 2 octets per frame |
| 0x47D | 0x03 | Register 0x47D, Bits[7:0] = 0x03: enable Lane 0 and Lane 1 |

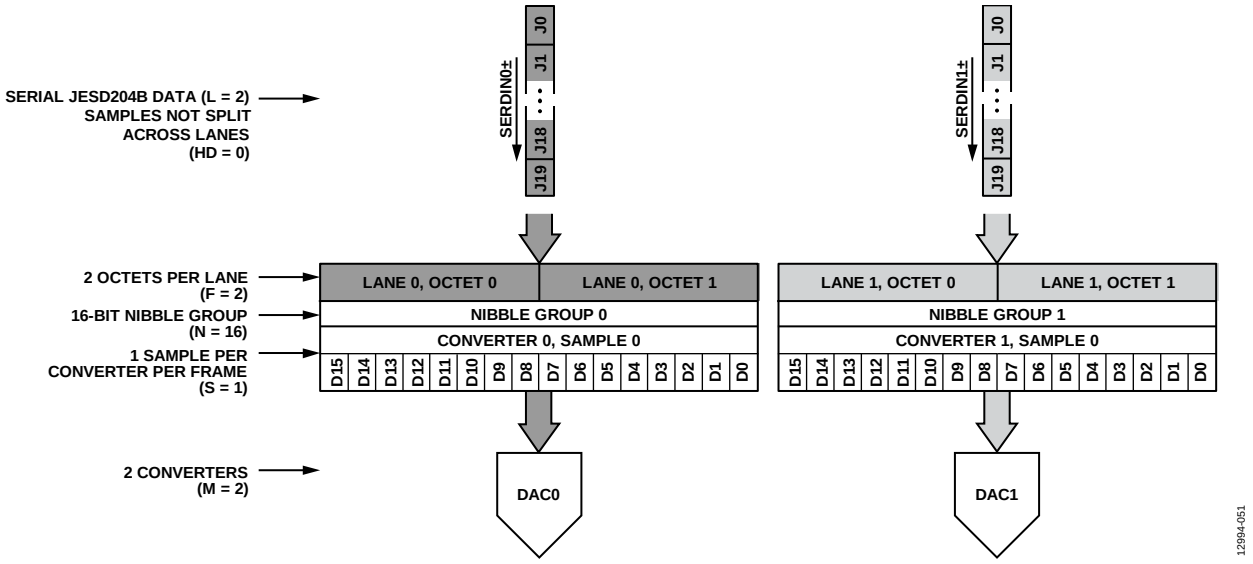


*Figure 56. JESD204B Mode 6 Data Deframing*

**Table 48. SPI Configuration Map—Register Settings for JESD204B Parameters for Mode 7**

| Address | Setting | Description |
|---|---|---|
| 0x453 | 0x00 or 0x80 | Register 0x453, Bit 7 = 0 or 1: scrambling disabled or enabled, Register 0x453, Bits[4:0] = 0x0: L = 1 lane per link |
| 0x454 | 0x03 | Register 0x454, Bits[7:0] = 0x03: F = 4 octets per frame per lane |
| 0x455 | 0x0F or 0x1F | Register 0x455, Bits[4:0] = 0x0F or 0x1F: K = 16 or 32 frames per multiframe |
| 0x456 | 0x01 | Register 0x456, Bits[7:0] = 0x01: M = 2 converters per link |
| 0x457 | 0x0F | Register 0x457, Bits[7:6] = 0x0: always set CS = 0; Register 0x457, Bits[4:0] = 0x0F: N = 16, always set to 16-bit resolution |
| 0x458 | 0x0F or 0x2F | Register 0x458, Bits[7:5] = 0x0 or 0x1: Subclass 0 or Subclass 1, Register 0x458, Bits[4:0] = 0xF: NP = 16 bits per sample |
| 0x459 | 0x20 | Register 0x459, Bits[7:5] = 0x1: set to JESD204B version, Register 0x459, Bits[4:0] = 0x0: S = 1 sample per converter per frame |
| 0x45A | 0x00 | Register 0x45A, Bit 7 = 0: HD = 0; Register 0x45A, Bits[4:0] = 0x00: always set CF = 0 |
| 0x46C | 0x01 | Register0x46C, Bits[7:0] = 0x01: deskew Lane 0 |
| 0x476 | 0x04 | Register 0x476, Bits[7:0] = 0x04: F = 4 octets per frame |
| 0x47D | 0x01 | Register 0x47D, Bits[7:0] = 0x01: enable Lane 0 |

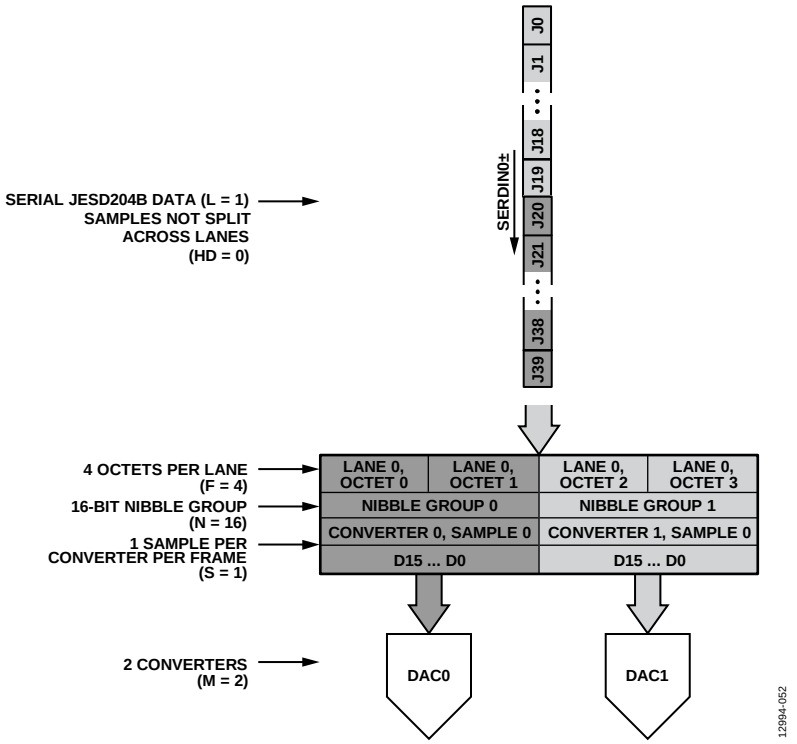


*Figure 57. JESD204B Mode 7 Data Deframing*

**Table 49. SPI Configuration Map—Register Settings for JESD204B Parameters for Mode 9**

| Address | Setting | Description |
|---|---|---|
| 0x453 | 0x01 or 0x81 | Register 0x453, Bit 7 = 0 or 1: scrambling disabled or enabled, Register 0x453, Bits[4:0] = 0x1: L = 2 lanes per link |
| 0x454 | 0x00 | Register 0x454, Bits[7:0] = 0x00: F = 1 octet per frame per lane |
| 0x455 | 0x1F | Register 0x455, Bits[4:0] = 0x1F: K = 32 frames per multiframe |
| 0x456 | 0x00 | Register 0x456, Bits[7:0] = 0x00: M = 1 converter per link |
| 0x457 | 0x0F | Register 0x457, Bits[7:6] = 0x0: always set CS = 0; Register 0x457, Bits[4:0] = 0x0F: N = 16, always set to 16-bit resolution |
| 0x458 | 0x0F or 0x2F | Register 0x458, Bits[7:5] = 0x0 or 0x1: Subclass 0 or Subclass 1, Register 0x458, Bits[4:0] = 0xF: NP = 16 bits per sample |
| 0x459 | 0x20 | Register 0x459, Bits[7:5] = 0x1: Set to JESD204B version, Register 0x459, Bits[4:0] = 0x0: S = 1 sample per converter per frame |
| 0x45A | 0x80 | Register 0x45A, Bit 7 = 1: HD = 1; Register 0x45A, Bits[4:0] = 0x00: always set CF = 0 |
| 0x46C | 0x03 | Register0x46C, Bits[7:0] = 0x03: deskew Lane 0 and Lane 1 |
| 0x476 | 0x01 | Register 0x476, Bits[7:0] = 0x01: F = 1 octet per frame |
| 0x47D | 0x03 | Register 0x47D, Bits[7:0] = 0x03: enable Lane 0 and Lane 1 |

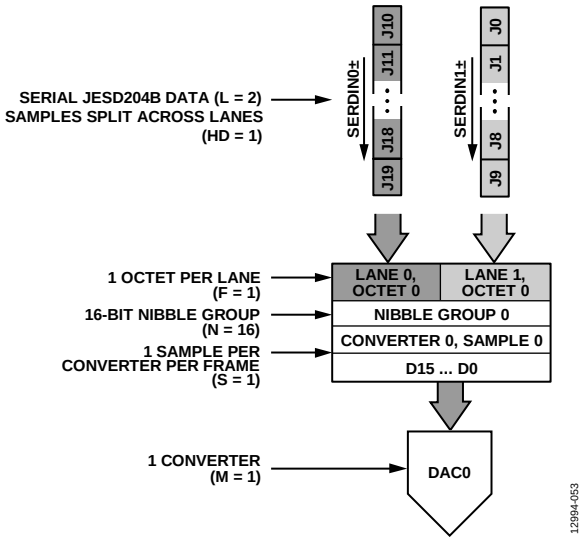


*Figure 58. JESD204B Mode 9 Data Deframing*

**Table 50. SPI Configuration Map—Register Settings for JESD204B Parameters for Mode 10**

| Address | Setting | Description |
|---|---|---|
| 0x453 | 0x00 or 0x80 | Register 0x453, Bit 7 = 0 or 1: scrambling disabled or enabled, Register 0x453, Bits[4:0] = 0x0: L = 1 lane per link |
| 0x454 | 0x01 | Register 0x454, Bits[7:0] = 0x01: F = 2 octets per frame per lane |
| 0x455 | 0x0F or 0x1F | Register 0x455, Bits[4:0] = 0x0F or 0x1F: K = 16 or 32 frames per multiframe |
| 0x456 | 0x00 | Register 0x456, Bits[7:0] = 0x00: M = 1 converter per link |
| 0x457 | 0x0F | Register 0x457, Bits[7:6] = 0x0: always set CS = 0; Register 0x457, Bits[4:0] = 0x0F: N = 16, always set to 16-bit resolution |
| 0x458 | 0x0F or 0x2F | Register 0x458, Bits[7:5] = 0x0 or 0x1: Subclass 0 or Subclass 1, Register 0x458, Bits[4:0] = 0xF: NP = 16 bits per sample |
| 0x459 | 0x20 | Register 0x459, Bits[7:5] = 0x1: set to JESD204B version, Register 0x459, Bits[4:0] = 0x0: S = 1 sample per converter per frame |
| 0x45A | 0x00 | Register 0x45A, Bit 7 = 0: HD = 0; Register 0x45A, Bits[4:0] = 0x00: always set CF = 0 |
| 0x46C | 0x01 | Register0x46C, Bits[7:0] = 0x01: deskew Lane 0 |
| 0x476 | 0x02 | Register 0x476, Bits[7:0] = 0x02: F = 2 octets per frame |
| 0x47D | 0x01 | Register 0x47D, Bits[7:0] = 0x01: enable Lane 0 |

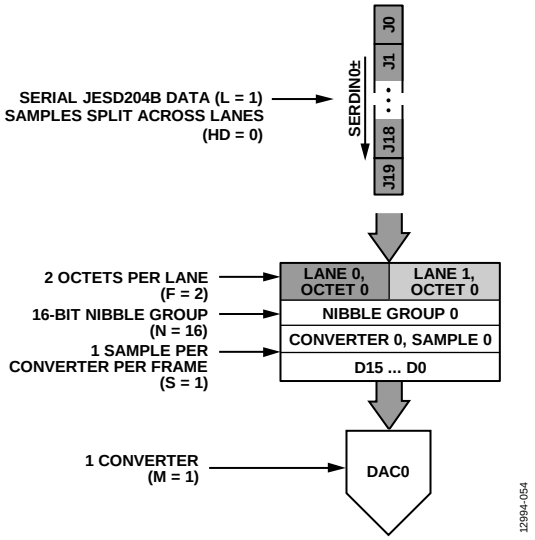


*Figure 59. JESD204B Mode 10 Data Deframing*

## JESD204B TEST MODES

### PHY PRBS Testing

The JESD204B receiver on the AD9152 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 AD9152 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 51.
3. Enable the PHY test for all lanes being tested by writing to PHY_TEST_EN (Register 0x315, Bits[3:0]). 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_ERROR_THRESHOLD (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.
7. Wait 500 ms.
8. Stop the test by writing 0 to PHY_TEST_START (Register 0x316, Bit 1).
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 3) in the PHY_SRC_ERR_CNT (Register 0x316, Bits[6:4]) and reading the PHY_PRBS_ERR_COUNT (Register 0x31C to Register 0x31A). The maximum error count is $2^{24-1}$. If all bits of Register 0x31C to Register 0x31A are high, the maximum error count on the selected lane has been exceeded.

**Table 51. 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 AD9152 supports the short transport layer (STPL) test as described in the JESD204B standard. This test can be used to verify the data mapping between the JESD204B transmitter and receiver. To perform this test, this function must be implemented in the logic device and enabled there. Before running the test on the receiver side, the link must be established and running without errors (see the Device Setup Guide).

The STPL test ensures that each sample from each converter is mapped appropriately according to the number of converters (M) and the number of samples per converter (S). As specified in the JESD204B standard, the converter manufacturer specifies what test samples are transmitted. Each sample must have a unique value. For example, if M = 2 and S = 2, four unique samples are transmitted repeatedly until the test is stopped. The expected sample must be programmed into the device and the expected sample is compared to the received sample one sample at a time until all have been tested. The process to perform this test on the AD9152 is described as follows:

1. Synchronize the JESD204B link.
2. Enable the STPL test at the JESD204B Tx.
3. Select Converter 0, Sample 0 for testing. Write 0 to SHORT_TPL_DAC_SEL (Register 0x32C, Bits[3:2]) and 0 to SHORT_TPL_SP_SEL (Register 0x32, Bits[5:4]).
4. Set the expected test sample for Converter 0, Sample 0. Program the expected 16-bit test sample into the SHORT_TPL_REF_SP_x registers (Register 0x32E and Register 0x32D).
5. Enable the STPL test. Write 1 to SHORT_TPL_TEST_EN (Register 0x32C, Bit 0).
6. Toggle the STPL reset, SHORT_TPL_TEST_RESET (Register 0x32C, Bit 1), from 0 to 1 then back to 0.
7. Check for failures. Read SHORT_TPL_FAIL (Register 0x32F, Bit 0), 0 = pass, 1 = fail.
8. Repeat Step 3 to Step 7 for each sample of each converter, Converter 0, Sample 0 through Converter M − 1, Sample S − 1.

### Repeated CGS and ILAS Test

Per Section 5.3.3.8.2 of the JESD204B specification, the AD9152 can verify that a constant stream of K28.5 characters is being received, or that a 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 AD9152 SERDES inputs. Next, set up the device and enable the links as described in the Device Setup Guide section. Ensure that the K28.5 characters are being received by verifying that $\overline{\text{SYNCOUT}}\pm$ has been deasserted and that the CGS has passed for all enabled lanes by reading Register 0x470.

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

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

## JESD204B ERROR MONITORING

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

Per Section 7.6 of the JESD204B specification, the AD9152 can detect disparity errors, not in table errors, and unexpected control character errors, and can optionally issue a sync request and reinitialize the link when errors occur.

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

#### Checking Error Counts

The error count can be checked for disparity errors, not in table errors, and unexpected control character errors. The error counts are on a per lane and per error type basis. Note that the lane select and counter select are programmed into Register 0x46B and the error count is read back from the same address. To check the error count, complete the following steps:

1. Select the desired lane and error type of the counter to view. Write these to Register 0x46B according to Table 52.
2. Read the error count from Register 0x46B. Note that the maximum error count is equal to the error threshold set in Register 0x47C.

**Table 52. Error Counters**

| Addr. | Bits | Variable | Description |
|---|---|---|---|
| 0x46B | [6:4] | LaneSel | LaneSel = x to monitor the error count of Lane x. |
| | [1:0] | CntrSel | CntrSel = 0b00 for a bad running disparity counter.<br>CntrSel = 0b01 for a not in table error counter.<br>CntrSel = 0b10 for an unexpected control character counter. |

#### Check for Error Count over Threshold

In addition to reading the error count per lane and error type as described in the Checking Error Counts section, the user can check a register to see if the error count for a given error type has reached a programmable threshold.

The same error threshold is used for the three error types (disparity, not in table, and unexpected control character).

The error counters are on a per error type basis. To use this feature, complete the following steps:

1. Program the desired error count threshold into ERRORTHRES (Register 0x47C).
2. Read back the error status for each error type to see if the error count has reached the error threshold.
   - Disparity errors are reported in Register 0x46D.
   - Not in table errors are reported in Register 0x46E.
   - Unexpected control character errors are reported in Register 0x46F.

#### Error Counter and IRQ Control

The user can write to Register 0x46D and Register 0x46F to reset or disable the error counts and to reset the IRQ for a given lane. Note that these are the same registers that report error count over threshold (see the Check for Error Count over Threshold section); thus, the readback is not the value that was written. For each error type,

1. Decide whether to reset the IRQ, disable the error count, and/or reset the error count for the given lane and error type.
2. Write the lane and desired reset or disable action to Register 0x46D to Register 0x46F according to Table 53.

**Table 53. Error Counter and IRQ Control: Disparity (Register 0x46D), Not in Table (Register 0x46E), and Unexpected Control Character (Register 0x46F)**

| Bits | Variable | Description |
|---|---|---|
| 7 | RstIRQ | RstIRQ = 1 to reset IRQ for the lane selected in Bits[2:0]. |
| 6 | Disable_ErrCnt | Disable_ErrCnt = 1 to disable the error count for the lane selected in Bits[2:0]. |
| 5 | RstErrCntr | RsteErrCntr = 1 to reset the error count for the lane selected in Bits[2:0]. |
| [2:0] | LaneAddr | LaneAddr = x to monitor the error count of Lane x. |

#### Monitoring Errors via SYNCOUT±

When one or more disparity, not in table, or unexpected control character error occurs, the error is reported on the SYNCOUT± pins per Section 7.6 of the JESD204B specification. The JESD204B specification states that the SYNCOUT± signal is asserted for exactly two frame periods when an error occurs. For the AD9152, the width of the SYNCOUT± pulse can be programmed to ½, 1, or 2 PClock cycles. The settings to achieve a SYNCOUT± pulse of two frame clock cycles are given in Table 54.

**Table 54. Setting SYNCOUT± Error Pulse Duration**

| JESD204B Mode IDs | PClockFactor (Frames/PClock) | SYNCB_ERR_DUR (Register 0x312, Bits[7:4]) Setting[1] |
|---|---|---|
| 4, 9 | 4 | 0 (default) |
| 5, 6, 10 | 2 | 1 |
| 7 | 1 | 2 |

[1] These register settings assert the SYNCOUT± signal for two frame clock cycle pulse widths.

### Disparity, Not in Table, and Unexpected Control Character IRQs

For disparity, not in table, and unexpected control character errors, error count over the threshold events are available as IRQ events. Enable these events by writing to Register 0x47A, Bits[7:5]. The IRQ event status can be read at the same address (Register 0x47A, 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 character errors are received, per Section 7.1 of the JESD204B specification. When a link reinitialization occurs, the resync request is five frames and nine octets long.

The user can optionally reinitialize the link when the error count for disparity errors, not in table errors, or unexpected control characters reaches a programmable error threshold. The process to enable the reinitialization feature for certain error types is as follows:

1. Set THRESHOLD_MASK_EN (Register 0x477, Bit 3) = 1. Note that when this bit is set, unmasked errors do not saturate at either threshold or maximum value.
2. Enable the sync assertion mask for each type of error by writing to the SYNCASSERTIONMASK register (Register 0x47B, Bits[7:5]) according to Table 55.
3. Program the desired error counter threshold into ERRORTHRES (Register 0x47C).
4. For each error type enabled in the SYNCASSERTION-MASK register, if the error counter on any lane reaches the programmed threshold, $\overline{\text{SYNCOUT}}$± falls, issuing a sync request. Note that all error counts are reset when a link reinitialization occurs. The IRQ does not reset and must be reset manually.

**Table 55. Sync Assertion Mask**

| Addr. | Bit No. | Bit Name | Description |
|---|---|---|---|
| 0x47B | 7 | BADDIS_S | Set to 1 to assert $\overline{\text{SYNCOUT}}$± if the disparity error count reaches the threshold |
| | 6 | NIT_S | Set to 1 to assert $\overline{\text{SYNCOUT}}$± if the not in table error count reaches the threshold |
| | 5 | UCC_S | Set to 1 to assert $\overline{\text{SYNCOUT}}$± if the unexpected control character 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 CODEGRPSYNCFLAG (Register 0x470) is high if Lane x received at least four K28.5 characters and passed code group synchronization.

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

Bit x of GOODCHKSUMFLG (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 Lane x. The parameters can be added either by summing the individual fields in the registers or summing the packed register. If Register 0x300, Bit 6 = 0 (default), the calculated checksums are the lower eight bits of the sum of the following fields: DID, BID, LID, SCR, L − 1, F − 1, K − 1, M − 1, N − 1, SUBCLASSV, NP − 1, JESDV, S − 1, and HD. If Register 0x300, Bit 6 = 1, the calculated checksums are the lower eight bits of the sum of Register 0x400 to Register 0x40C and LID (Register 0x412, Register 0x41A, and Register 0x422).

Bit x of INITIALLANESYNC (Register 0x473) is high if 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. To enable them, write to Register 0x47A, Bits[3:0]. The IRQ event status can be read at the same address (Register 0x47A, Bits[3:0]) after the IRQs are enabled. To reset the CGS IRQ, write 1 to Bit 7 of Register 0x470. To reset the frame sync IRQ, write 1 to Bit 7 of Register 0x471. To reset the checksum IRQ, write 1 to Bit 7 of Register 0x472. to reset the ILAS IRQ, write 1 to Bit 7 of Register 0x473. See the Interrupt Request Operation section for more information.

### Configuration Mismatch IRQ

The AD9152 has a configuration mismatch flag that is available as an IRQ event. Use Register 0x47B, Bit 3 to enable the mismatch flag (it is enabled by default), and then use Register 0x47B, Bit 4 to read back its status and reset the IRQ signal. See the Interrupt Request Operation section for more information.

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

Note that 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

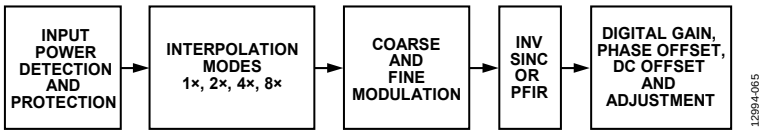


*Figure 60. Block Diagram of the Digital Datapath*

The block diagram in Figure 60 shows the functionality of the digital datapath (all blocks can be bypassed). The digital processing includes an input power detection block, three half-band interpolation filters, a quadrature modulator consisting of a fine resolution NCO and $f_{DAC}/4$ and $f_{DAC}/8$ coarse modulation block, an inverse sinc filter, and gain, phase, offset, and group delay adjustment blocks.

The interpolation filters take independent I and Q data streams. If using the modulation function, I and Q must be quadrature data to function properly.

Note that the pipeline delay changes when digital datapath functions are enabled/disabled. If fixed DAC pipeline latency is desired, do not reconfigure these functions after initial configuration.

## DATA FORMAT

BINARY_FORMAT (Register 0x110, Bit 7) controls the expected input data format. By default it is 0, which means the input data must be in twos complement. It can also be set to 1, which means input data is in offset binary (0x0000 is negative full scale and 0xFFFF is positive full scale).

## INTERPOLATION FILTERS

The transmit path contains three half-band interpolation filters, which each provide a 2× increase in output data rate and a low-pass function. The filters can be cascaded to provide a 4× or 8× interpolation ratio. Table 56 shows how to select each available interpolation mode, their usable bandwidths, and their maximum data rates. Note that $f_{DATA} = f_{DAC}/InterpolationFactor$. Register 0x030, Bit 0 is high if an unsupported interpolation mode is selected.

**Table 56. Interpolation Modes and Usable Bandwidth**

| Interpolation Mode | INTERP_MODE Reg. 0x112[2:0] | Usable Bandwidth | Maximum $f_{DATA}$ (MHz) |
|---|---|---|---|
| 1× (Bypass) | 0x00 | 0.5 × $f_{DATA}$ | 1238[1] |
| 2× | 0x01 | 0.4 × $f_{DATA}$ | 1125 |
| 4× | 0x02 | 0.4 × $f_{DATA}$ | 562.5 |
| 8× | 0x03 | 0.4 × $f_{DATA}$ | 281.25 |

[1] The maximum speed for 1× interpolation mode is limited by the JESD204B interface.

### *Filter Performance*

The interpolation filters interpolate between existing data in such a way that they minimize changes in the incoming data while suppressing the creation of interpolation images. This is shown for each filter in Figure 61.

The usable bandwidth (as shown in Table 56) 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.

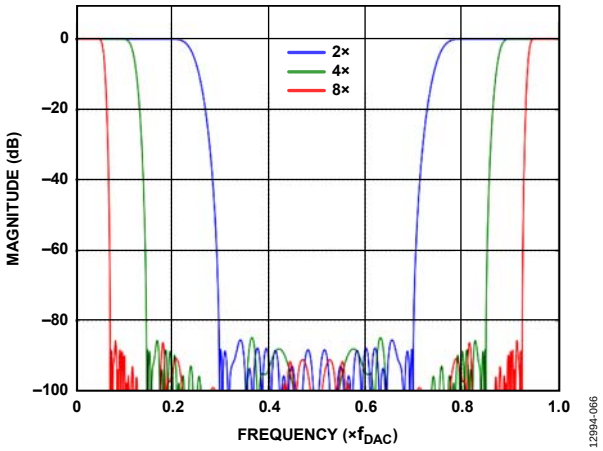


*Figure 61. All Band Responses of Interpolation Filters*

### Filter Performance Beyond Specified Bandwidth

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

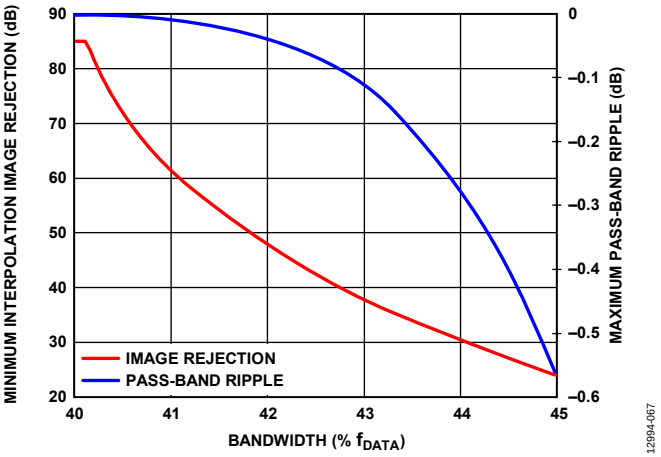


*Figure 62. Interpolation Filter Performance Beyond Specified Bandwidth*

Figure 62 shows the performance of the interpolation filters beyond 0.4 × $f_{DATA}$. Note that the ripple increases much slower than the image rejection decreases. This means that if the application can tolerate degraded image rejection from the interpolation filters, more bandwidth can be used.

## DIGITAL MODULATION

The AD9152 provides two modes to modulate the baseband quadrature signal to the desired DAC output frequency.

- $f_{DAC}/4$ and $f_{DAC}/8$ coarse modulation
- NCO fine modulation

The coarse modulation modes ($f_{DAC}/4$ and $f_{DAC}/8$) allow modulation by those particular frequencies. The NCO fine modulation mode allows modulating by a programmable frequency at the cost of higher power consumption, depending on the DAC rate. Modulation mode is selected as shown in Table 57.

**Table 57. Modulation Mode Selection**

| Modulation Mode | MODULATION_TYPE, Register 0x111, Bits[3:2] |
|---|---|
| None | 0b00 |
| NCO Fine Modulation | 0b01 |
| Coarse, $f_{DAC}/4$ | 0b10 |
| Coarse, $f_{DAC}/8$ | 0b11 |

Based on the difference of direct digital synthesis (DDS) accumulator, NCO modulation has the following two modes:

- Typical accumulator-based DDS (see the NCO Fine Modulation section)
- Programmable modulus DDS (see the Programmable Modulus DDS section)

### *$f_{DAC}/4$ and $f_{DAC}/8$ Modulation*

The $f_{DAC}/4$ and $f_{DAC}/8$ modulation are common modulation modes to translate the input baseband frequency to a fixed $f_{DAC}/4$ or $f_{DAC}/8$ IF frequency. These coarse modulation are selected by setting Bits[3:2] in Register 0x111. These modes provide lower power modulation frequencies of 1/4 or 1/8 of the DAC sampling rate. When modulation frequencies other than this frequency are required, the NCO modulation mode must be used.

### *NCO Fine Modulation*

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 63. This allows output signals to be placed anywhere in the output spectrum with very fine frequency resolution.

The NCO produces a quadrature carrier to translate the input signal to a new center frequency. A quadrature carrier is a pair of sinusoidal waveforms of the same frequency, offset 90° from each other. The frequency of the quadrature carrier is set via an 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 63.

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

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

The FTW is set as shown in Table 58.

**Table 58. NCO FTW Registers**

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

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

SEL_SIDEBAND (Register 0x111, Bit 1) is a convenience bit that can be set to use the negative modulation result. This is equivalent to flipping the sign of FTW. SEL_SIDEBAND also applies to fs/4 and fs/8 modulation.

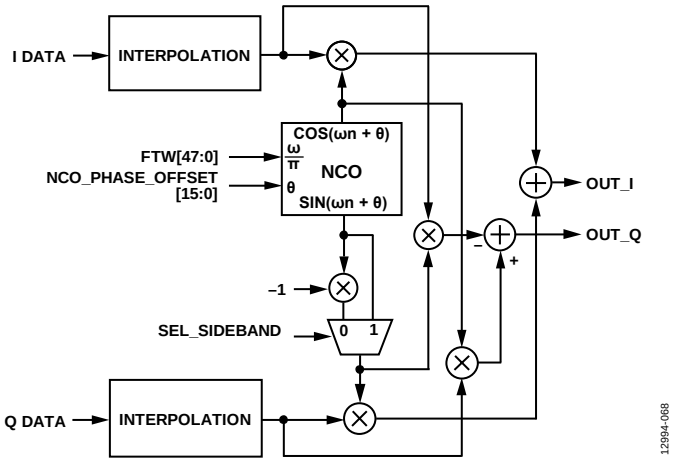


*Figure 63. NCO Fine Modulator Block Diagram*

### NCO Phase Offset

The phase offset feature allows rotation of the I and Q phases. Unlike phase adjust, this feature moves the phases of both I and Q channels together. Phase offset can be used only when using NCO fine modulation.

$$-180° \le DegreesOffset < +180°$$
$$PhaseOffset = (DegreesOffset/180°) \times 2^{15}$$

where *PhaseOffset* is a 16-bit, twos complement number.

The NCO phase offset is set as shown in Table 59. Because this function is part of the fine modulation block, phase offset is not updated immediately upon writing. Instead, it updates on the rising edge of FTW_UPDATE_REQ (Register 0x113, Bit 0), along with the FTW.

**Table 59. NCO Phase Offset Registers**

| Address | Value |
|---|---|
| 0x11A | NCO_PHASE_OFFSET[7:0] |
| 0x11B | NCO_PHASE_OFFSET[15:8] |

### Programmable Modulus DDS

The programmable modulus is a modification of the typical accumulator-based DDS architecture (NCO). 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, it extends the use of DDS 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 AD9152 is such that the fraction, M/N, is expressible per the equation below. 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 0x158 to Register 0x15D.
*B* is programmed in Register 0x152 to Register 0x157.

### Programmable Modulus Example

Consider the case in which $f_{DAC}$ = 250 MHz and the desired value of $f_{CARRIER}$ is 25 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 (25,000,000/250,000,000) to its lowest terms. That is,

$$M/N = 25{,}000{,}000/250{,}000{,}000 = 1/10$$

Therefore, M = 1 and N = 10. After calculation, X = 429,496,729; A = 3; and B = 5. Programming these values into the registers causes the modulus DDS to produce an output frequency of exactly 25 MHz, given a 250 MHz sampling clock. For more details, see the AN-953 Application Note.

## NCO ALIGNMENT

The NCO alignment block phase aligns the NCO output from multiple converters. Two NCO alignment modes are supported by the AD9152. The first is a SYSREF± alignment mode that phase aligns the NCO outputs to the rising edge of a SYSREF± pulse. The second alignment mode is a data key alignment; when this mode is enabled, the AD9152 aligns the NCO outputs when a user specified data pattern arrives at the DAC input.

### SYSREF± NCO Alignment

As with the LMFC alignment, in Subclass 1, a SYSREF± pulse phase aligns the NCO outputs of multiple devices in a system and multiple channels on the same device. Note that in Subclass 0, this alignment mode can be used to align the NCO outputs within a device to an internal processing clock edge. No SYSREF± edge is needed in Subclass 0, but multichip alignment cannot be achieved. The steps to achieve a SYSREF± NCO alignment are as follows:

1. Set NCO_ALIGN_MODE (Register 0x050, Bits[1:0] = 0b01) for SYSREF± NCO alignment mode.
2. Set NCO_ALIGN_ARM (Register 0x050, Bit 7) to 1.
3. Perform an LMFC alignment to force the NCO phase align (see the Syncing LMFC Signals section). The phase alignment occurs on the next SYSREF± edge.
   Note that if in one shot sync mode, the LMFC alignment block must be armed by setting Register 0x03A, Bit 6 = 1. If in continuous mode or one shot then monitor mode, the LMFC align block does not need to be armed; the NCO align automatically trips on the next SYSREF± edge.
4. Check the alignment status. If NCO phase alignment was successful, NCO_ALIGN_PASS (Register 0x050, Bit 4) = 1. If phase alignment failed, NCO_ALIGN_FAIL (Register 0x050, Bit 3) = 1.

### Data Key NCO Alignment

In addition to supporting the SYSREF± alignment mode, the AD9152 supports a mode in which the NCO phase alignment occurs when a user-specified pattern is seen at the DAC input. The steps to achieve a data key NCO alignment are as follows:

1. Set NCO_ALIGN_MODE (Register 0x050, Bits[1:0]) to 0b10.
2. Write the expected 16-bit data key for the I and Q datapath into NCOKEYI[15:0] (Register 0x051 to Register 0x052) and NCOKEYQ[15:0] (Register 0x053 to Register 0x054), respectively.
3. Set NCO_ALIGN_ARM (Register 0x050, Bit 7) to 1.
4. Send the expected 16-bit I and Q data keys to the device to achieve NCO alignment.
5. Check the alignment status. If the expected data key was seen at the DAC input, NCO_ALIGN_MTCH (Register 0x050, Bit 5) = 1. If NCO phase alignment was successful, NCO_ALIGN_PASS (Register 0x050, Bit 4) = 1. If phase alignment failed, NCO_ALIGN_FAIL (Register 0x050, Bit 3) = 1.

Multiple device NCO alignment can be achieved with the data key alignment mode. To achieve multichip NCO alignment, program the same expected data key on all devices, arm all devices, and then send the data key to all devices/channels at the same time.

### NCO Alignment IRQ

An IRQ event showing whether the NCO align was tripped is available. Use Register 0x021, Bit 4 to enable the IRQ and then use Register 0x025, Bit 4 to read back its status and reset the IRQ signal. See the Interrupt Request Operation section for details.

## INVERSE SINC

The AD9152 provides a digital inverse sinc filter to compensate the DAC roll-off over frequency. The filter is enabled by setting the INVSINC_ENABLE bit (Register 0x111, Bit 7) and is disabled by default.

The inverse sinc ($sinc^{-1}$) filter is a seven-tap FIR filter. Figure 64 shows the frequency response of sin(x)/x roll-off, the inverse sinc filter, and the composite response. The composite response has less than ±0.05 dB pass-band ripple up to a frequency of $0.4 \times f_{DAC}$. To provide the necessary peaking at the upper end of the pass band, the inverse sinc filter shown has an intrinsic insertion loss of approximately 3.8 dB; in many cases, this can be partially compensated as described in the Digital Gain section.

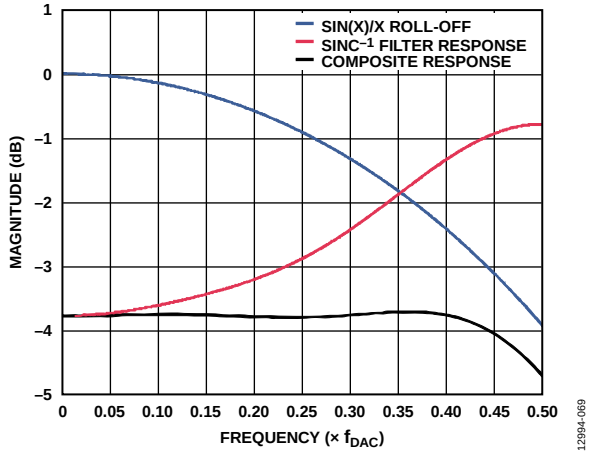


*Figure 64. Responses of sin(x)/x Roll-Off, the Sinc$^{-1}$ Filter, and the Composite of the Two Input Signal Power Detection and Protection*

## PROGRAMMABLE FIR FILTER (PFIR)

The PFIR is a seven-tap FIR filter, which can be programmed through the registers to compensate the gain nonflatness of RF signal chain.

The PFIR is in parallel with INVSINC and is a superset of INVSINC. Do not enable the PFIR and INVSINC at the same time. A detailed specification of the PFIR filter follows:

- The coefficients are in 1.8 format, one sign bit, 8 resolution bits, and range from −1 to +1. Set the coefficients in Register 0x17A to Register 0x181.
- Symmetry coefficients.
- The PFIR works with programmable coefficients with 0 dB to 6 dB gain at the filter output. To avoid signal overflow, set the filter to less than 6 dB gain. A gain of −6 dB is applied at the PFIR output to bring it back to avoid overflow on the following blocks. However, the gain loss here can be easily compensated at the final digital gain stages. See the Digital Gain section.
- The PFIR can be turned off to save power.

## DIGITAL GAIN, PHASE ADJUST, DC OFFSET, AND COARSE GROUP DELAY

Digital gain, phase adjust, and dc offset (as described in the Digital Gain section, Phase Adjust section, and DC Offset section) allow compensation of imbalances in the I and Q paths due to analog mismatches between I/Q DAC outputs, quadrature modulator I/Q baseband inputs, and DAC/modulator interface I/Q paths. These imbalances can cause the two following issues:

- An unwanted sideband signal to appear at the quadrature modulator output with significant energy. This can be tuned out using digital gain and phase adjust. Tuning the quadrature gain and phase adjust values can optimize complex image rejection in single sideband radios or can optimize the error vector magnitude (EVM) in zero IF (ZIF) architectures.
- The dc offset can cause LO leakage through a modulator, which can be compensated with the offset feature in the DAC.

Coarse group delay allows adjustment of the delay through the DAC, which can be used to adjust digital predistortion (DPD) loop delay.

### *Digital Gain*

Digital gain can be used to independently adjust the digital signal magnitude being fed into each DAC. This is useful to balance the gain between I and Q channels of a dual or to cancel out the insertion loss of the inverse sinc filter. Digital gain must be enabled when using the blanking state machine (see the Downstream Protection section). If digital gain is disabled, TXENx must be tied high.

Digital gain is enabled by setting the DIG_GAIN_ENABLE bit (Register 0x111, Bit 5). In addition to enabling the function, the amount of digital gain (GainCode) desired must be programmed. By default, digital gain is enabled and GainCode = 0x800, which means 0 dB gain.

$$0 \le Gain \le 4095/2048$$
$$-\infty \text{ dB} \le dBGain \le 6.018 \text{ dB}$$
$$Gain = GainCode \times (1/2048)$$
$$dBGain = 20 \times \log10(Gain)$$
$$GainCode = 2048 \times Gain = 2048 \times 10^{dBGain/20}$$

where *GainCode* is a 12-bit, unsigned binary number.

The I/Q digital gain is set as shown in Table 60.

**Table 60. Digital Gain Registers**

| Addr. | Value | Description |
|---|---|---|
| 0x111[5] | DIG_GAIN_ENABLE | Set to 1 to enable digital gain |
| 0x13C | IDAC_DIG_GAIN[7:0] | I DAC LSB gain code |
| 0x13D | IDAC_DIG_GAIN[11:8] | I DAC MSB gain code |
| 0x13E | QDAC_DIG_GAIN[7:0] | Q DAC LSB gain code |
| 0x13F | QDAC_DIG_GAIN[11:8] | Q DAC MSB gain code |

### Phase Adjust

Ordinarily, the I and Q channels of each DAC pair have an angle of 90° between them. The phase adjust feature changes the angle between the I and Q channels, which can help balance the phase into a modulator.

$$-14 \le DegreesAdjust < 14$$
$$IQPhaseAdj = (DegreesAdjust/14) \times 2^{12}$$

where *IQPhaseAdj* is a 13-bit, twos complement number.

The phase adjust is set as shown in Table 61.

**Table 61. I/Q Phase Adjustment Registers**

| Addr. | Value | Description |
|---|---|---|
| 0x111[4] | PHASE_ADJ_ENABLE | Set to 1 to enable phase adjust |
| 0x11C | PHASE_ADJ[7:0] | LSB phase adjust code |
| 0x11D | PHASE_ADJ[12:8] | MSB phase adjust code |

### DC Offset

The dc offset feature individually offsets the data into the I or Q DACs. This can be used to cancel LO leakage.

The offset is programmed individually for I and Q as a 16-bit twos complement number in LSBs, plus a 5-bit, twos complement number in sixteenths of an LSB, as shown in Table 62.

$$-2^{15} \le LSBsOffset < 2^{15} - 1$$
$$-16/16 \text{ LSB} \le SixteenthsOffset \le 15/16 \text{ LSB}$$

**Table 62. DC Offset Registers**

| Addr. | Value | Description |
|---|---|---|
| 0x135[0] | DC_OFFSET_ON | Set to 1 to enable dc offset |
| 0x136 | LSB_OFFSET_I[7:0] | I DAC LSB dc offset code |
| 0x137 | LSB_OFFSET_I[15:8] | I DAC MSB dc offset code |
| 0x138 | LSB_OFFSET_Q[7:0] | Q DAC LSB dc offset code |
| 0x139 | LSB_OFFSET_Q[15:8] | Q DAC MSB dc offset code |
| 0x13A | SIXTEENTH_OFFSET_I | I DAC sub-LSB dc offset code |
| 0x13B | SIXTEENTH_OFFSET_Q | Q DAC sub-LSB dc offset code |

Figure 65 shows how the DAC output currents vary as a function of the LSBsOffset value. With the digital inputs fixed at midscale (0x0000, twos complement data format), Figure 65 shows the nominal current of the positive node of the DAC output, $I_{IOUT+}/I_{QOUT+}$, as the DAC offset value is swept from 0 to 65,535. Because IOUT+/QOUT+ and IOUT−/QOUT− are complementary current outputs, the sum of $I_{IOUT+}$ and $I_{IOUT-}$ or $I_{QOUT+}$ and $I_{QOUT-}$ is always 20 mA.

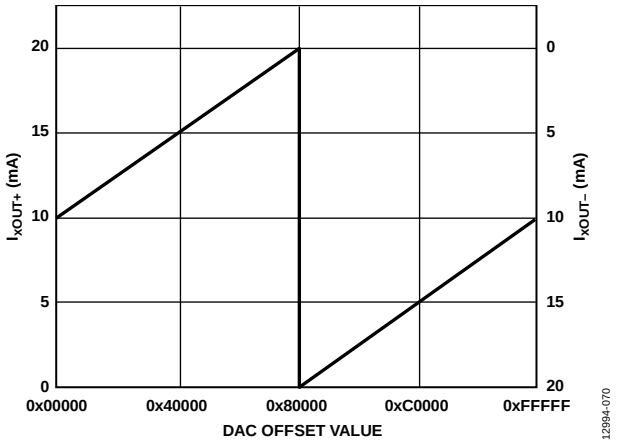


*Figure 65. DAC Output Currents vs. DAC Offset Value*

### Coarse Group Delay Adjustment

Coarse group delay is programmed in Register 0x047. The range of the delay is −4 DAC clock periods to +3 DAC clock periods and the resolution is 1/2 DAC clock period. Coarse group delay can be used for DPD loop delay adjustment.

## DOWNSTREAM PROTECTION

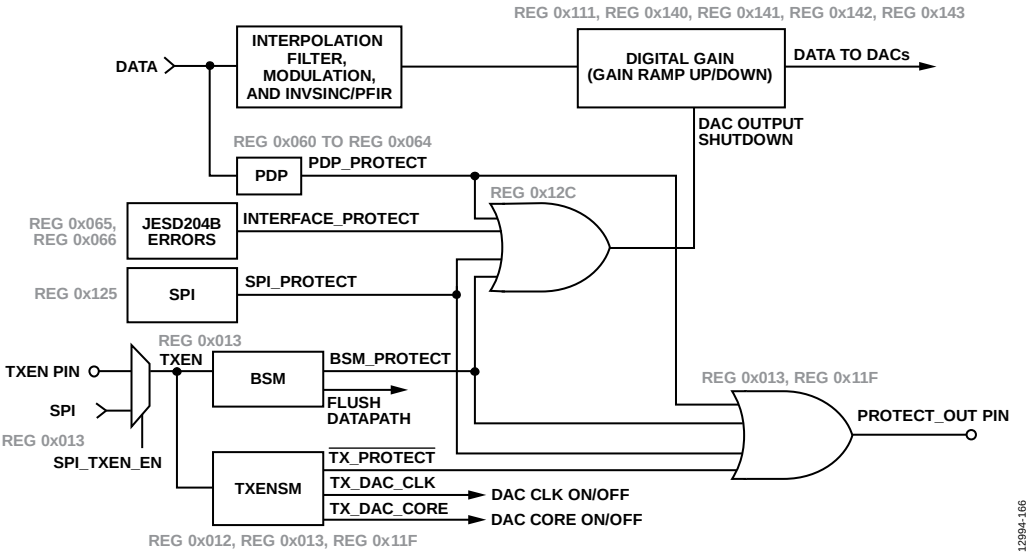


*Figure 66. Downstream Protection Block Diagram*

The AD9152 has several blocks designed to protect the power amplifier (PA) of the system, as well as other downstream blocks. First, the DAC output can be shut down by tuning the digital gain that is automatically triggered by the following signals: PDP_PROTECT, INTERFACE_PROTECT, SPI_PROTECT, and BSM_PROTECT. Second, an external pin (PROTECT_OUT) can be used to shut down external components. The PROTECT_OUT pin is triggered by the following signals: PDP_PROTECT, INTERFACE_PROTECT, SPI_PROTECT, and TX_PROTECT.

The downstream protection function largely consists of a power detection and protection (PDP) block, a blanking state machine (BSM), and a transmit enable state machine (TXENSM).

The PDP block can be used to monitor incoming data. If a moving average of the data power goes above a threshold, the PDP block provides a signal (PDP_PROTECT) that can be routed externally and turn off the DAC output gradually by tuning the digital gain.

The TXENSM block controls the delay between TXEN and the Tx_PROTECT signals. At the same time, the Tx_PROTECT signal can optionally be routed externally to the PROTECT_OUT pin, and the TXENSM can also generate the TX_DAC_CLK signal and TX_DAC_CORE signal to power down the DAC clock and the DAC core.

The BSM block flushes the datapath and turns the DAC output on or off by tuning the digital gain that is triggered by BSM_PROTECT, which can also be routed to the external PROTECT_OUT pin. BSM_PROTECT follows the status of TXEN.

TXEN can come from the external TXEN pin or the SPI, which is selected by SPI_TXEN_EN (Register 0x013, Bit 1).

### *Power Detection and Protection*

The input signal 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) that can be routed externally to shut down a PA and shut down the DAC output.

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 whose output is the average of the input signal power in a certain number of samples.

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 a signal on the PROTECT_OUT pin and turn off the DAC output through digital gain.

The choice of PDP_AVG_TIME (Register 0x062) and PDP_THRESHOLD (Register 0x060 to Register 0x061) for effective protection are application dependent. Experiment with real-world vectors to ensure proper configuration. The PDP_POWER readback (Register 0x063 to Register 0x064) can help by storing the maximum power when a set threshold is passed. The PDP block is configured as shown in Table 63.

**Table 63. PDP Registers**

| Addr. | Bit No. | Value | Description |
|---|---|---|---|
| 0x060 | [7:0] | PDP_THRESHOLD[7:0] | Power that triggers PDP_PROTECT. 8 LSBs. |
| 0x061 | [4:0] | PDP_THRESHOLD[12:8] | 5 MSBs. |
| 0x062 | 7 | PDP_ENABLE | Set to 1 to enable PDP. |
| | [3:0] | PDP_AVG_TIME | Can be set from 0 to 10. Averages across $2^{(9 + PDP\_AVG\_TIME)}$, IQ sample pairs. |
| 0x063 | [7:0] | PDP_POWER[7:0] | If PDP_THRESHOLD is crossed, this reads back the maximum power seen. If not, this reads back the instantaneous power. 8 LSBs. |
| 0x064 | [4:0] | PDP_POWER[12:8] | 5 MSBs. |
| 0x12C | 7 | PROTECT_MODE | If this bit is high, the DAC is in protect mode, and is shut down automatically when some errors occur. |
| | 0 | DACOFF_AVG_PW | If this bit is high, Bit 7 is high, and the input average power is greater than the given threshold (see Register 0x060 and Register 0x061) within a given time window, the DAC output shuts down automatically. |
| 0x013 | 6 | PDP_PROTECT_OUT | 1: PDP_PROTECT triggers PROTECT_OUT. |

### Power Detection and Protection IRQ

The PDP_PROTECT signal is available as an IRQ event.

Use Register 0x021, Bit 7 to enable PDP_PROTECT and then use Register 0x025, Bit 7 to read back its status and reset the IRQ signal.

See the Interrupt Request Operation section for more information.

### *Transmit Enable State Machine*

The TXENSM block controls the delay between TXEN and the $\overline{\text{Tx\_PROTECT}}$ signals. At the same time, the $\overline{\text{Tx\_PROTECT}}$ signal can optionally be routed externally to the PROTECT_OUT pin and the TXENSM can generate the TX_DAC_CLK signal and TX_DAC_CORE signal to power down the DAC clock and DAC core.

If DACA_MASK (Register 0x012, Bit 6) = 1, a falling edge of TXEN causes the DAC core I DAC and Q DAC to power down; a rising edge of TXEN causes the DAC core I DAC and Q DAC to power up.

If CLKA_MASK (Register 0x012, Bit 4 ) = 1, a falling edge of TXEN causes the DAC clock to power down; a rising edge of TXEN causes the DAC clock to power up.

The TXENSM is configured as shown in Table 64.

**Table 64. TXENSM Registers**

| Addr. | Bit No. | Value | Description |
|---|---|---|---|
| 0x012 | 6 | DACA_MASK | DAC core power-down mask for TXEN. |
| | 4 | CLKA_MASK | Datapath power-down mask for TXEN. |
| 0x013 | 5 | TX_PROTECT_OUT | 1: TX_PROTECT triggers PROTECT_OUT. |
| 0x11F | 0 | TXEN_SM_EN | If high, enable TXEN state machine. |

### TXENSM Startup Sequence

To ensure that the TXENSM functions properly, the following sequence must be used. Excepting Register 0x012 and Register 0x013, other registers must follow the recommend values in Table 65.

**Table 65. TXENSM Startup Sequence[1]**

| Addr. | Value | Description |
|---|---|---|
| 0x012 | 0x00 | Enable the DAC core and datapath power-down mask |
| 0x013 | 0x20 | TX_PROTECT triggers PROTECT_OUT |
| 0x140 | 0x04 | Gain ramp up step |
| 0x142 | 0x09 | Gain ramp down step |
| 0x11F | 0x83 | Enable the TXEN state machine |

[1] Perform these writes in the order they are listed in this table.

After applying the sequence in Table 65, the function of the TXENSM timing is shown in Figure 67.

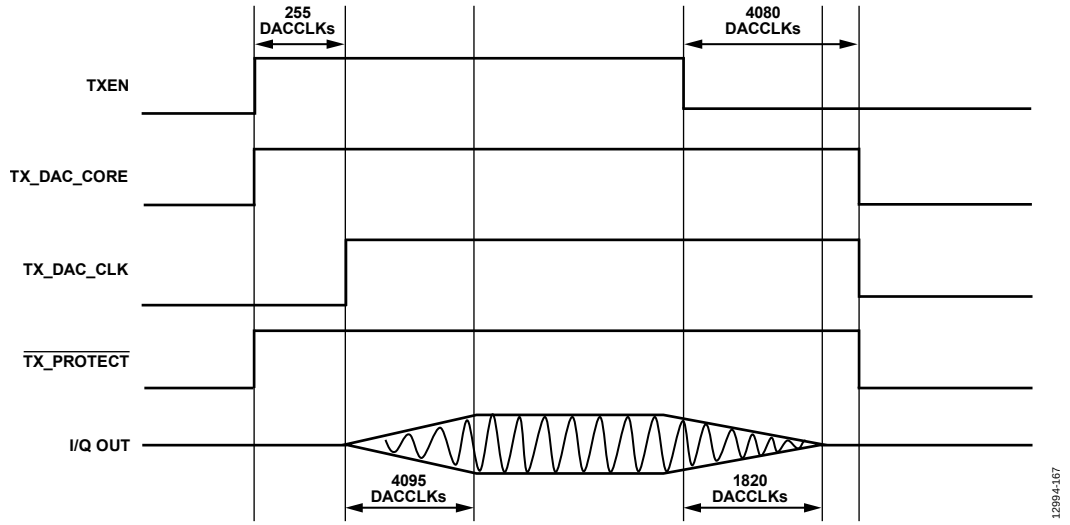


*Figure 67. TXENSM Timing*

### *Blanking State Machine (BSM)*

The BSM block flushes the datapath and turns the DAC output on or off through the digital gain and is triggered by BSM_PROTEST, which can also be routed to the external PROTECT_OUT pin. BSM_PROTECT follows the status of TXEN. The BSM is configured as shown in Table 66.

**Table 66. TXENSM Registers**

| Addr. | Bit No. | Value | Description |
|---|---|---|---|
| 0x013 | 4 | BSM_PROTECT_OUT | 1: BSM triggers PROTECT_OUT |

### *Shutdown DAC Output*

The DAC output can be shut down gradually by tuning the digital gain that is automatically triggered by the following signals: PDP_PROTECT, INTERFACE_PROTECT, SPI_PROTECT, and BSM_PROTECT. For proper ramping, digital gain must be enabled. The step size to use when ramping the gain to 0 or its assigned value can be controlled via the GAIN_RAMP_DOWN_STEP registers (Register 0x142 and Register 0x143) and the GAIN_RAMP_UP_STEP registers (Register 0x140 and Register 0x141).

Besides the PDP block and the BSM block, certain JESD204B and SPI write errors can also be configured to shut down the DAC output when they occur using Register 0x065, Register 0x066, and Register 0x125.

### *PROTECT_OUT Generation*

Register 0x013 controls which signals are ORed into the external PROTECT_OUT signal (see Table 67). Register 0x11F, Bit 2 can be used to invert the PROTECT_OUT signal. By default, PROTECT_OUT is high when the output is valid.

**Table 67. PROTECT_OUT Registers**

| Addr. | Bit No. | Value | Description |
|---|---|---|---|
| 0x013 | 6 | PDP_PROTECT_OUT | 1: PDP_PROTECT triggers PROTECT_OUT |
| | 5 | TX_PROTECT_OUT | 1: TX_PROTECT triggers PROTECT_OUT |
| | 4 | BSM_PROTECT_OUT | 1: BSM_PROTECT triggers PROTECT_OUT |
| | 3 | SPI_PROTECT_OUT_EN | 1: SPI_PROTECT triggers PROTECT_OUT |
| | 2 | SPI_PROTECT_OUT_CTRL | 1: PROTECT_OUT is low |
| | 1 | SPI_TXEN_EN | 1: TXEN is controlled by the SPI |
| | 0 | SPI_TXEN_CTRL | 1: TXEN is high |

## DATAPATH PRBS

The datapath PRBS verifies that the AD9152 datapath is receiving and correctly decoding data. The datapath PRBS verifies that the JESD204B parameters of the transmitter and receiver match, the lanes of the receiver are mapped appropriately, lanes have been appropriately inverted, if necessary, and in general that the start-up routine has been implemented correctly. Note that the datapath PRBS function applies only to 2×, 4×, and 8× interpolation.

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

1. Set up the device in the desired operating mode. See the Device Setup Guide section for details on setting up the device.
2. Send the PRBS7 or PRBS15 data.
3. Write 0 to Register 0x14B, Bit 2 for PRBS7 or write 1 for PRBS15.
4. Write 0b11 to Register 0x14B, Bits[1:0] to enable and reset the PRBS test.
5. Write 0b01 to Register 0x14B, Bits[1:0] to enable the PRBS test and release reset.
6. Wait 500 ms.
7. Check the status by checking the IRQ for the I DAC and the Q DAC PRBS as described in the Datapath PRBS IRQ section.
8. Read Register 0x14B, Bits[7:6]. Bit 6 is 0 if the I DAC of the selected dual has any errors. Bit 7 is 0 if the Q DAC of the selected dual has any errors. This must match the IRQ.
9. Read Register 0x14C to read the error count for the I DAC. Read Register 0x14D to read the error count for the Q DAC.

Note that the PRBS processes 32 bits at a time, and compares the 32 new bits to the previous set of 32 bits. It detects (and reports) only 1 error in every group of 32 bits; therefore, the error count partly depends on when the errors are seen.

For example,

- Bits: 32 good, 31 good, 1 bad; 32 good (2 errors)
- Bits: 32 good, 22 good, 10 bad; 32 good (2 errors)
- Bits: 32 good, 31 good, 1 bad; 31 good, 1 bad; 32 good (3 errors)

### *Datapath PRBS IRQ*

The PRBS fail signals for each DAC are available as IRQ events. Use Register 0x020, Bits[1:0] to enable the fail signals, and then use Register 0x02s[1:0] to read back their statuses and reset the IRQ signals. See the Interrupt Request Operation section for more information.

## DC TEST MODE

As a convenience, the AD9152 provides a dc test mode, which is enabled by setting Register 0x0F7, Bit 1. When this mode is enabled, the datapath is given 0 (midscale) for its data.

In conjunction with dc offset, this test mode can provide desired dc data to the DACs. This test mode can also provide sinusoidal data to the DACs by combining digital modulation (to set frequency) and dc offset (to set amplitude). See the DC Offset section.

# INTERRUPT REQUEST OPERATION

The AD9152 provides an interrupt request output signal on Pin 10 ($\overline{IRQ}$) that can be used to notify an external host processor of significant device events. On assertion of the interrupt, query the device to determine the precise event that occurred. The $\overline{IRQ}$ pin is an open-drain, active low output. Pull the $\overline{IRQ}$ pin high external to the device. This pin can be tied to the interrupt pins of other devices with open-drain outputs to wire; OR these pins together.

Figure 68 shows a simplified block diagram of how the IRQ blocks works. If IRQ_EN is low, the INTERRUPT_SOURCE signal is set to 0. If IRQ_EN is high, any rising edge of an event causes the INTERRUPT_SOURCE signal to be set high. If any INTERRUPT_SOURCE signal is high, the $\overline{IRQ}$ pin is pulled low. INTERRUPT_SOURCE can be reset to 0 by either an IRQ reset signal or a device reset.

Depending on STATUS_MODE, the EVENT_STATUS bit reads back an event or INTERRUPT_SOURCE. The AD9152 has several IRQ register blocks that can monitor up to 48 events (depending on device configuration). Certain details vary by IRQ register block as described in Table 68. Table 69 shows which registers the IRQ_EN, IRQ_RESET, and STATUS_MODE signals in Figure 68 originate from, as well as the address where EVENT_STATUS is read back.

**Table 68. IRQ Register Block Details**

| Register Block | Event Reported | EVENT_STATUS |
|---|---|---|
| 0x01F to 0x025 | Per chip | INTERRUPT_SOURCE if IRQ is enabled, if not, it is an event |
| 0x46D to 0x46F; 0x470 to 0x473; 0x47A | Per link and lane | INTERRUPT_SOURCE if IRQ is enabled, if not, 0 |
| 0x47B, Bit 4 | Per link | INTERRUPT_SOURCE if IRQ is enabled, if not, 0 |

## INTERRUPT SERVICE ROUTINE

Interrupt request management begins 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. For Register 0x01F to Register 0x026, EVENT_STATUS has a live readback. For other events, see their registers.
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. Enable the interrupt by writing 1 to IRQ_EN.
7. Clear the interrupt by writing 1 to IRQ_RESET.

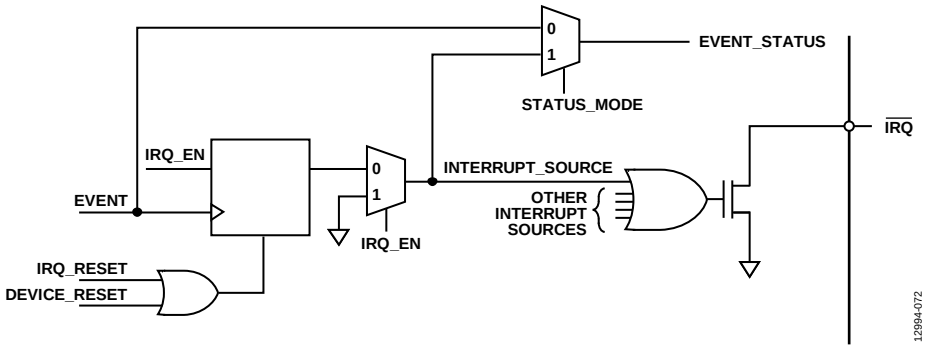


*Figure 68. Simplified Schematic of $\overline{IRQ}$ Circuitry*

**Table 69. IRQ Register Block Address of IRQ Signal Details**

| | Address of IRQ Signals | | | |
|---|---|---|---|---|
| Register Block | IRQ_EN | IRQ_RESET | STATUS_MODE | EVENT_STATUS |
| 0x01F to 0x025 | 0x01F to 0x021; R/W per chip | 0x023 to 0x025; W per chip | STATUS_MODE = IRQ_EN | 0x023 to 0x25; R per chip |
| 0x46D to 0x46F | 0x47A; W per link | 0x46D to 0x46F; W per link and lane | Not applicable, STATUS_MODE = 1 | 0x47A; R per link |
| 0x470 to 0x473 | 0x47A; W per link | 0x470 to 0x473; W per link | Not applicable, STATUS_MODE = 1 | 0x47A; R per link |
| 0x47B, Bit 4 | 0x47B, Bit 3; R/W per link; 1 by default | 0x47B, Bit 4; W per link | Not applicable, STATUS_MODE = 1 | 0x47B, Bit 4; R per link |

# DAC INPUT CLOCK CONFIGURATIONS

The AD9152 DAC sample clock (DACCLK) can be sourced directly through DACCLK± (Pin 48 and Pin 49) or by clock multiplication through the REFCLK± differential input (Pin 3 and Pin 4). Clock multiplying employs the on-chip PLL that accepts a reference clock operating at a submultiple of the desired DACCLK rate. The PLL then multiplies the reference clock up to the desired DACCLK frequency, which generates all the internal clocks required by the DAC. The clock multiplier provides a high quality clock that meets the performance requirements of most applications. Using the on-chip clock multiplier removes the burden of generating and distributing the high speed DACCLK.

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

## DRIVING THE DACCLK± AND REFCLK± INPUTS

The DACCLK± and REFCLK± differential inputs share similar clock receiver input circuitry, shown in Figure 69. The on-chip clock receiver has a differential input impedance of 10 kΩ. It is self biased to a common-mode voltage of approximately 600 mV. The inputs can be driven by differential PECL or LVDS drivers with ac coupling between the clock source and the receiver.

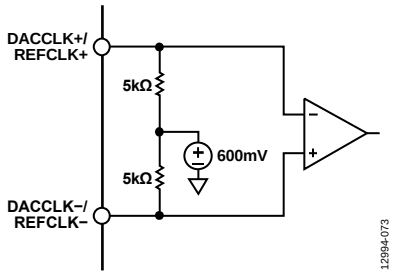


*Figure 69. Simplified Equivalent Circuit of the Clock Receiver Input*

The minimum input drive level to the differential clock input is 400 mV p-p differential. The optimal performance is achieved when the clock input signal is between 600 mV p-p differential and 800 mV p-p differential. Whether using the on-chip clock multiplier or sourcing the DACCLK directly, the input clock signal to the device must have low jitter and fast edge rates to optimize the DAC noise performance. Direct clocking with a low noise clock produces the lowest noise spectral density at the DAC outputs.

The clocks and clock receiver are powered down by default. The clocks must be enabled by writing to Register 0x011. To enable all clocks on the device, write 0x00 to Register 0x011.

## CONDITION SPECIFIC REGISTER WRITES

### *Clock Multiplication Relationships*

The on-chip PLL clock multiplier circuit generates the DAC sample rate clock from a lower frequency reference clock. The PLL is integrated on chip, including the VCO and the loop filter. The VCO operates over the frequency range of 6 GHz to 12 GHz.

The PLL configuration parameters must be programmed before the PLL is enabled. Step by step instructions on how to program the PLL can be found in the Temperature Tracking section. The functional block diagram of the clock multiplier is shown in Figure 70.

The clock multiplication circuit generates the DAC sampling clock from the REFCLK± input, which is fed in on the REFCLK± differential pins (Pin 3 and Pin 4). The frequency of the REFCLK± input is referred to as $f_{REF}$.

The REFCLK± input is divided by the variable RefDivFactor. Select the RefDivFactor variable to ensure that the frequency into the phase frequency detector (PFD) block is between 35 MHz and 80 MHz. The valid values for RefDivFactor are 2, 4, 8, 16, or 32. Each RefDivFactor maps to the appropriate REF_DIV_MODE register control according to Table 70. The REF_DIV_MODE register is programmed through Register 0x08C, Bits[2:0].

**Table 70. Mapping of RefDivFactor to REF_DIV_MODE**

| DAC Reference Frequency Range (MHz) | Divide by Factor (RefDivFactor) | REF_DIV_MODE, Reg. 0x08C, Bits[2:0] |
|---|---|---|
| 70 to 160 | 2 | 1 |
| 160 to 320 | 4 | 2 |
| 320 to 640 | 8 | 3 |
| 640 to 1000 | 16 | 4 |

The range of $f_{REF}$ is 80 MHz to 1 GHz, and the output frequency of the PLL is 420 MHz to 2.25 GHz. Use the following equations to determine the RefDivFactor:

$$35\ \text{MHz} < \frac{f_{REF}}{RefDivFactor} < 80\ \text{MHz} \quad (1)$$

where:
$f_{REF}$ is the reference frequency on the REFCLK± input pins.
*RefDivFactor* is the reference divider division ratio.

The BCount value is the divide ratio of the loop divider. It is set to divide the $f_{DAC}$ to frequency match the $f_{REF}$/RefDivFactor. Select BCount so that the following equation is true:

$$\frac{f_{DAC}}{2 \times BCount} = \frac{f_{REF}}{RefDivFactor} \quad (2)$$

where:
$f_{DAC}$ is the DAC sample clock.
*BCount* is the feedback loop divider ratio.

The BCount value is programmed with Bits[7:0] of Register 0x085. It is programmable from 6 to 127.

The PFD compares $f_{REF}$/RefDivRate to $f_{DAC}$/(2 × BCount) and pulses the charge pump up or down to control the frequency of

the VCO. A low noise VCO is tunable over an octave with an oscillation range of 6 GHz to 12 GHz.

The clock multiplication circuit operates such that the VCO outputs a frequency, $f_{VCO}$.

$$f_{VCO} = f_{DAC} \times LODivFactor \quad (3)$$

From Equation 2, the DAC sample clock frequency, $f_{DAC}$, equals

$$f_{DAC} = 2 \times BCount \times \frac{f_{REF}}{RefDivFactor} \quad (4)$$

The LODivFactor is chosen to keep $f_{VCO}$ in the operating range between 6 GHz and 12 GHz. The valid values for LODivFactor are 4, 8, and 16. Each LODivFactor maps to a LO_DIV_MODE value. The LO_DIV_MODE (Register 0x08B, Bits[1:0]) is programmed as described in Table 71.

**Table 71. DAC VCO Divider Selection**

| DAC Frequency Range (MHz) | Divide by Factor (LODivFactor) | LO_DIV_MODE, Register 0x08B, Bits[1:0] |
|---|---|---|
| >1500 | 4 | 1 |
| 750 to 1500 | 8 | 2 |
| 420 to 750 | 16 | 3 |

Table 72 lists some common frequency examples for the RefDivFactor, LODivFactor, and BCount values that are needed to configure the PLL properly.

**Table 72. Common Frequency Examples**

| Freq. (MHz) | $f_{DAC}$ (MHz) | $f_{VCO}$ (MHz) | RefDiv-Factor | LODiv-Factor | BCount |
|---|---|---|---|---|---|
| 368.64 | 1474.56 | 11796.48 | 8 | 8 | 16 |
| 184.32 | 1474.56 | 11796.48 | 4 | 8 | 16 |
| 307.2 | 1228.88 | 9831.04 | 8 | 8 | 16 |
| 122.88 | 983.04 | 7864.35 | 2 | 8 | 8 |
| 61.44 | 983.04 | 7864.35 | 1 | 8 | 8 |
| 491.52 | 1966.08 | 7864.35 | 8 | 4 | 16 |
| 245.76 | 1966.08 | 7864.35 | 4 | 4 | 16 |

### *Temperature Tracking*

When properly configured, the device automatically selects one of the 512 VCO bands. The PLL settings selected by the device ensure that the PLL remains locked over the full −40°C to +85°C operating temperature range of the device without further adjustment. The PLL remains locked over the full temperature range even if the temperature during initialization is at one of the temperature extremes.

To properly configure temperature tracking, See Figure 17 and Figure 18 and the $f_{VCO}$ dependent SPI writes shown in Table 73.

**Table 73. DAC PLL VCO Control Lookup Table**

| VCO Frequency Range (GHz) | Register 0x1B6 Setting | Register 0x1BB Setting | Register 0x1B5 Setting | Register 0x1C5 Setting |
|---|---|---|---|---|
| $f_{VCO}$ < 6.84 | 0x49 | 0x1A | 0xC7 | 0x08 |
| 6.84 ≤ $f_{VCO}$ < 8.69 | 0x49 | 0x12 | 0xC9 | 0x06 |
| 8.69 ≤ $f_{VCO}$ < 10.54 | 0x4D | 0x04 | 0xC9 | 0x06 |
| $f_{VCO}$ ≥ 10.54 | 0x4D | 0x04 | 0xC9 | 0x06 |

## STARTING THE PLL

The programming sequence for the DAC PLL is as follows:

1. Use the equations in the Clock Multiplication Relationships section to find $f_{VCO}$, $f_{REF}$, BCount, RefDivMode, and LODivMode.
2. Program the registers in Figure 17 and Figure 18
3. Program the value of LODivMode into Register 0x08B, Bits[1:0].
4. Program the value of BCount into Register 0x085, Bits[7:0].
5. Program the value of RefDivMode into Register 0x08C, Bits[2:0].
6. Based on the $f_{VCO}$ found in Step 1, write the temperature tracking registers as shown in Table 73.
7. Enable the DAC PLL synthesizer by setting Register 0x083, Bit 4 to 1.

Register 0x084, Bit 5 notifies the user that the DAC PLL calibration is completed and is valid.

Register 0x084, Bit 1 notifies the user that the PLL has locked.

Register 0x084, Bit 7 and Register 0x084, Bit 6 notify the user that the DAC PLL has reached the upper or lower edge of its operating band, respectively. If either of these bits are high, recalibrate the DAC PLL by setting Register 0x083, Bit 7 to 0 and then 1.

### DAC PLL IRQ

The DAC PLL lock and lost signals are available as IRQ events. Use Register 0x01F, Bits[7:6] to enable these signals, and then use Register 0x023, Bits[7:6] to read back their statuses and reset the IRQ signals. See the Interrupt Request Operation section for more information.

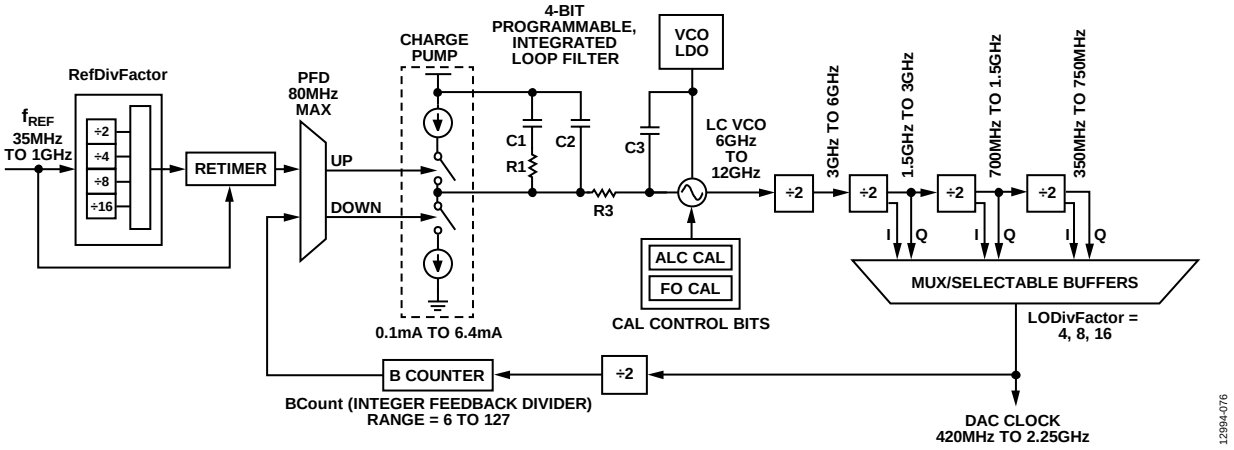


*Figure 70. Device Clock PLL Block Diagram*

# ANALOG OUTPUTS

## TRANSMIT DAC OPERATION

Figure 71 shows a simplified block diagram of the transmit path DACs. The DAC core consists of a current source array, a switch core, digital control logic, and full-scale output current control. The DAC full-scale output current ($I_{OUTFS}$) is nominally 20.22 mA. The output currents from the IOUT±/QOUT± pins are complementary, meaning that the sum of the two currents always equals the full-scale current of the DAC. The digital input code to the DAC determines the effective differential current delivered to the load.

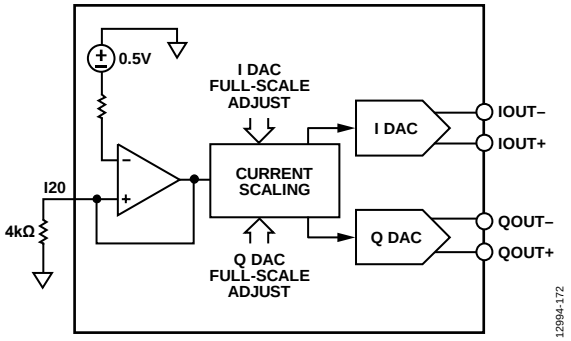


*Figure 71. Simplified Block Diagram of the DAC Core*

The DAC has a 0.5 V band gap reference with an output impedance of 5 kΩ. A 4 kΩ external resistor, $R_{SET}$, must be connected from the I120 pin to the ground plane. This resistor, along with the reference control amplifier, sets up the correct internal bias currents for the DAC. Because the full-scale current is inversely proportional to this resistor, the tolerance of $R_{SET}$ is reflected in the full-scale output amplitude.

The full-scale current equation, where the DAC gain is set individually for the I and Q DACs in Register 0x040 through Register 0x043, respectively, is as follows:

$$I_{FS} = \frac{V_{REF}}{R_{SET}} \times \left(13.33 + \left(\frac{1}{19.19} \times DAC\ gain\right)\right)$$

For nominal values of $V_{REF}$ (1.2 V), $R_{SET}$ (4 kΩ), and DAC gain (1023), the full-scale current of the DAC is typically 20 mA. The DAC full-scale current can be adjusted from 4 mA to 20 mA by programming the values in Register 0x040 through Register 0x043, as shown in Table 74 and Figure 72.

**Table 74. DAC Full-Scale Current Registers**

| Address | Value | Description |
|---|---|---|
| 0x040[1:0] | DACFSC_I[9:8] | I DAC MSB gain code |
| 0x041[7:0] | DACFSC_I[7:0] | I DAC LSB gain code |
| 0x042[1:0] | DACFSC_Q[9:8] | Q DAC MSB gain code |
| 0x043[7:0] | DACFSC_Q[7:0] | Q DAC LSB gain code |

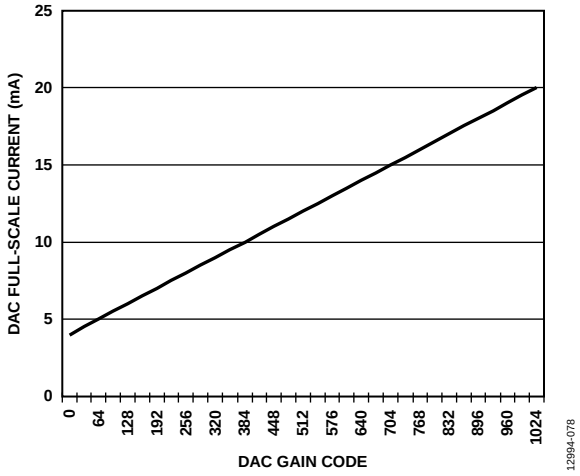


*Figure 72. DAC Full-Scale Current vs. DAC Gain Code*

### *Transmit DAC Transfer Function*

The output currents from the IOUT+/QOUT+ and IOUT−/QOUT− pins are complementary, meaning that the sum of the positive and negative currents always equals the full-scale current of the DAC. The digital input code to the DAC determines the effective differential current delivered to the load. IOUT± and QOUT± provide the maximum output current when all bits are high for binary data. The output currents vs. DACCODE for the DAC outputs using binary format are expressed as

$$I_{OUTP} = \frac{DACCODE_{BIN}}{2^N} \times I_{OUTFS} \quad (5)$$

$$I_{OUTN} = I_{OUTFS} - I_{OUTP} \quad (6)$$

where $DACCODE_{BIN}$ is the 16-bit input to the DAC in unsigned binary. $DACCODE_{BIN}$ has a range of 0 to $2^N - 1$.

If the data format is twos complement, the output currents are expressed as

$$I_{OUTP} = \frac{DACCODE_{TWOS} + 2^{N-1}}{2^N} \times I_{OUTFS} \quad (7)$$

$$I_{OUTN} = I_{OUTFS} - I_{OUTP}$$

# TEMPERATURE SENSOR

The AD9152 has a band gap temperature sensor for monitoring the temperature changes of the AD9152. The temperature must be calibrated against a known temperature to remove the device to device variation on the band gap circuit used to sense the temperature.

To monitor temperature change, take a reading at a known ambient temperature for a single-point calibration of each AD9152 device.

$$Tx = T_{REF} + 7.16 \times (CODE\_x - CODE\_REF)/1000$$

where:
*CODE_x* is the readback code at the unknown temperature, *Tx*.
*CODE_REF* is the readback code at the known calibrated temperature, $T_{REF}$.

To use the temperature sensor, it must be enabled by setting Register 0x12F, Bit 0 to 1. The user must write 0 and then 1 to Register 0x134, Bit 0 before reading back the die temperature from Register 0x132 and Register 0x133.

# EXAMPLE START-UP SEQUENCE

Table 75 through Table 83 show the register writes needed to set up the AD9152 with $f_{DAC}$ = 1474.56 MHz, 2× interpolation, and the DAC PLL enabled with a 368.64 MHz reference clock. The JESD204B interface is configured in Mode 4, single link mode, Subclass 1, and scrambling is enabled with all four SERDES lanes running at 7.3728 Gbps, inputting twos complement formatted data. No remapping of lanes with the crossbar is used in this example.

The sequence of steps to properly start up the AD9152 are as follows:

1. Set up the SPI interface, power up necessary circuit blocks, make required writes to the configuration register, and set up the DAC clocks (see Step 1: Start Up the DAC).

Set the digital features of the AD9152 (see [1] 0x denotes a register value that the user must fill in. See the Variable and Description columns for information on selecting the appropriate register value.

2. Step 2: Digital Datapath).
3. Set up the JESD204B links (see Step 3: Transport Layer).
4. Set up the physical layer of the SERDES interface (see Step 4: Physical Layer).
5. Set up the data link layer of the SERDES interface. This procedure is for quick startup or debug only and does not guarantee deterministic latency (see Step 5: Data Link Layer).
6. Check for errors on the link (see Step 6: Optional Error Monitoring).

These steps are outlined in detail in the following sections in tables that list the required register write and read commands.

## STEP 1: START UP THE DAC

### *Power-Up and DAC Initialization*

**Table 75. Power-Up and DAC Initialization**

| Command | Addr. | Value | Description |
|---|---|---|---|
| W | 0x000 | 0xBD | Soft reset |
| W | 0x000 | 0x3C | Deassert reset, set 4-wire SPI |
| W | 0x011 | 0x00 | Enable the reference, DAC channels, and clocks |
| W | 0x080 | 0x04 | Enable duty cycle correction |
| W | 0x081 | 0x04 | Power up the SYSREF± receiver, disable hysteresis |
| W | 0x1CD | 0xD8 | Band gap configuration |

### *Required Device Configurations*

**Table 76. Required SERDES PLL Configuration**

| Command | Addr. | Value[1] | Description |
|---|---|---|---|
| W | 0x284 | 0x62 | SERDES PLL configuration |
| W | 0x285 | 0xC9 | SERDES PLL configuration |
| W | 0x286 | 0xE | SERDES PLL configuration |
| W | 0x287 | 0x12 | SERDES PLL configuration |
| W | 0x28A | 0x | See Table 36 |
| W | 0x28B | 0x0 | SERDES PLL configuration |
| W | 0x290 | 0x89 | SERDES PLL configuration |
| W | 0x291 | 0x | See Table 36 |
| W | 0x294 | 0x24 | SERDES PLL configuration |
| W | 0x296 | 0x | See Table 36 |
| W | 0x297 | 0xD | SERDES PLL configuration |
| W | 0x299 | 0x2 | SERDES PLL configuration |
| W | 0x29A | 0x8E | SERDES PLL configuration |
| W | 0x29C | 0x2A | SERDES PLL configuration |
| W | 0x29F | 0x7E | SERDES PLL configuration |
| W | 0x2A0 | 0x6 | SERDES PLL configuration |

[1] 0x denotes a register value that the user must fill in. See the Variable and Description columns for information on selecting the appropriate register value.

### *Configure the DAC PLL*

**Table 77. Required DAC PLL Configuration**

| Command | Addr. | Value | Description |
|---|---|---|---|
| W | 0x08D | 0x7B | DAC PLL configuration |
| W | 0x1B0 | 0x0 | DAC PLL configuration |
| W | 0x1B9 | 0x24 | DAC PLL configuration |
| W | 0x1BC | 0xD | DAC PLL configuration |
| W | 0x1BE | 0x2 | DAC PLL configuration |
| W | 0x1BF | 0x8E | DAC PLL configuration |
| W | 0x1C0 | 0x2A | DAC PLL configuration |
| W | 0x1C4 | 0x7E | DAC PLL configuration |
| W | 0x1C1 | 0x2C | DAC PLL configuration |

**Table 78. Configure the DAC PLL**

| Command | Addr. | Value | Description |
|---|---|---|---|
| W | 0x08B | 0x02 | Set the VCO LO divider to 8 such that 6 GHz ≤ $f_{VCO} = f_{DAC} \times 2^{(LODivMode+1)} \leq 12$ GHz. |
| W | 0x08C | 0x03 | Set the reference clock divider to 8 so that the reference clock into the PLL is less than 80 MHz. |
| W | 0x085 | 0x10 | Set the B counter to 16 to divide the DAC clock down to 2× the reference clock. |
| W | 0x1B6 | 0x | See Table 73 |
| W | 0x1B5 | 0x | See Table 73 |
| W | 0x1BB | 0x | See Table 73 |
| W | 0x1B4 | 0x78 | Optimal DAC PLL VCO settings. |
| W | 0x1C5 | 0x | See Table 73 |
| W | 0x08A | 0x12 | Optimal DAC PLL VCO settings. |
| W | 0x087 | 0x62 | Optimal DAC PLL loop filter settings. |
| W | 0x088 | 0xC9 | Optimal DAC PLL loop filter settings. |
| W | 0x089 | 0x0E | Optimal DAC PLL loop filter settings. |
| W | 0x083 | 0x10 | Enable the DAC PLL. |
| R | 0x084 | 0x01 | Verify that Bit 1 reads back high for PLL locked. |

[1] 0x denotes a register value that the user must fill in. See the Variable and Description columns for information on selecting the appropriate register value.

## STEP 2: DIGITAL DATAPATH

**Table 79. Digital Datapath**

| Command | Addr. | Value | Description |
|---|---|---|---|
| W | 0x112 | 0x01 | Set the interpolation to 2×. |
| W | 0x110 | 0x00 | Set twos complement data format. |

## STEP 3: TRANSPORT LAYER

**Table 80. Link Transport Layer**

| Command | Addr. | Value | Description |
|---|---|---|---|
| W | 0x200 | 0x00 | Power up the interface |
| W | 0x201 | 0x00 | Enable all lanes |
| W | 0x300 | 0x00 | First power down JESD204B digital (by default) |
| W | 0x450 | 0x00 | Set the device ID to match Tx (0x00 in this example) |
| W | 0x451 | 0x00 | Set the bank ID to match Tx (0x00 in this example) |
| W | 0x452 | 0x00 | Set the lane ID to match Tx (0x00 in this example) |
| W | 0x453 | 0x83 | Set descrambling and L = 4 (in n – 1 notation) |
| W | 0x454 | 0x00 | Set F = 1 (in n – 1 notation) |
| W | 0x455 | 0x1F | Set K = 32 (in n – 1 notation) |
| W | 0x456 | 0x01 | Set M = 2 (in n – 1 notation) |
| W | 0x457 | 0x0F | Set N = 16 (in n – 1 notation) |
| W | 0x458 | 0x2F | Set Subclass 1 and NP = 16 (in n – 1 notation) |
| W | 0x459 | 0x20 | Set JESD204B Version and S = 1 (in n – 1 notation) |
| W | 0x45A | 0x80 | Set HD = 1 |
| W | 0x45D | 0x45 | Set checksum for Lane 0 |
| W | 0x46C | 0x0F | Deskew Lane 0 to Lane 3 |
| W | 0x476 | 0x01 | Set F (not in n – 1 notation) |
| W | 0x47D | 0x0F | Enable Lane 0 to Lane 3 |

## STEP 4: PHYSICAL LAYER

**Table 81. Physical Layer**

| Command | Addr. | Value | Description |
|---|---|---|---|
| W | 0x2A7 | 0x01 | Autotune PHY setting |
| W | 0x314 | 0x01 | SERDES SPI configuration |
| W | 0x230 | 0x29 | Configure CDRs in half rate mode and set the SYNCOUT± swing VOD to 350 mV |
| W | 0x206 | 0x00 | Resets CDR logic |
| W | 0x206 | 0x01 | Release CDR logic reset |
| W | 0x289 | 0x04 | Configure the PLL divider to 1 along with PLL required configuration |
| W | 0x280 | 0x01 | Enable the SERDES PLL |
| R | 0x281 | 0x01 | Verify that Bit 0 reads back high for the SERDES PLL lock |

## STEP 5: DATA LINK LAYER

Note that this procedure does not guarantee deterministic latency.

**Table 82. Data Link Layer—Does Not Guarantee Deterministic Latency**

| Command | Addr. | Value | Description |
|---|---|---|---|
| W | 0x301 | 0x01 | Set the subclass = 1 |
| W | 0x304 | 0x00 | Set the LMFC delay setting to 0 |
| W | 0x306 | 0x0A | Set the LMFC receive buffer delay to 10 |
| W | 0x03A | 0x01 | Set sync mode = one shot sync |
| W | 0x03A | 0x81 | Enable the sync machine |
| W | 0x03A | 0xC1 | Arm the sync machine |
| SYSREF± | | | Ensure that at least one SYSREF± edge is sent to the device |
| W | 0x300 | 0x01 | Bit 0 = 1 to enable Link 0. |

## STEP 6: OPTIONAL ERROR MONITORING

### *Link Checks*

Confirm that the registers in Table 83 read back as noted and system tasks are completed as described.

**Table 83. Link Checks**

| Command | Addr. | Value | Description |
|---|---|---|---|
| R | 0x470 | 0x0F | Acknowledge that four consecutive K28.5 characters have been detected on Lane 0 to Lane 3. Confirm that SYNCOUT± is high. |
| SERDINx± | | | Apply ILAS and data to SERDES input pins. |
| R | 0x471 | 0x0F | Check for frame sync on all lanes. |
| R | 0x472 | 0x0F | Check for good checksum. |
| R | 0x473 | 0x0F | Check for ILAS. |

# BOARD LEVEL HARDWARE CONSIDERATIONS

## POWER SUPPLY RECOMMENDATIONS

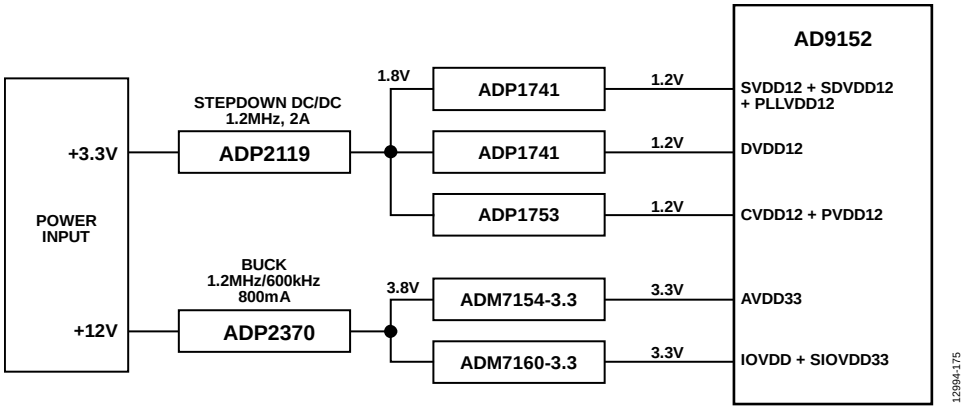


*Figure 73. Power Supply Connections*

**Table 84. Power Supplies**

| Power Supply Domain | Voltage (V) | Circuitry |
|---|---|---|
| DVDD12[1] | 1.2 | Digital core |
| PVDD12[2] | 1.2 | DAC PLL |
| CVDD12[1] | 1.2 | DAC clocking |
| SVDD12[3] | 1.2 | JESD204B analog |
| SDVDD12[3] | 1.2 | JESD204B digital |
| PLLVDD12[3] | 1.2 | SERDES PLL |
| $V_{TT}$[4] | 1.2 | $V_{TT}$ |
| AVDD33 | 3.3 | DAC |
| IOVDD | 3.3 | SPI interface and IOs |
| SIOVDD33 | 3.3 | Sync LVDS transmit |

[1] This supply requires a 1.3 V supply when operating at maximum DAC sample rates. See Table 3 for details.
[2] This supply may be combined with CVDD12 on the same regulator with a separate supply filter network and sufficient bypass capacitors near the pins.
[3] This supply requires a 1.3 V supply when operating at maximum interface rates. See Table 4 for details.
[4] This supply is connected to SVDD12 and does not need separate circuitry.

The power supply domains are described in Table 84. The power supplies can be grouped into separate PCB domains as show in Figure 73. All the AD9152 supply domains must remain as noise free as possible. Optimal DAC output NSD and DAC output phase noise performance can be achieved using linear regulators that provide excellent power supply rejection. AVDD33, PVDD12, and CVDD12 are particularly sensitive to supply noise.

## JESD204B SERIAL INTERFACE INPUTS (SERDIN0± TO SERDIN3±)

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 41). The AD9152 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 AD9152 as near 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.
- Use a PCB material with a low dielectric constant (<4) to minimize loss, if possible.

When choosing between stripline and microstrip techniques, consider the following: stripline has less loss (see Figure 42) and emits less EMI, but requires the use of vias that can add complexity to the task of controlling the impedance, whereas microstrip (see Figure 43) 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 micro vias to minimize via inductance.
- If using standard vias, use the maximum via length to minimize the stub size. For example, on an 8-layer board, use Layer 7 for the stripline pair (see Figure 74).
- For each via pair, place a pair of ground vias adjacent to them to minimize the impedance discontinuity (see Figure 74).

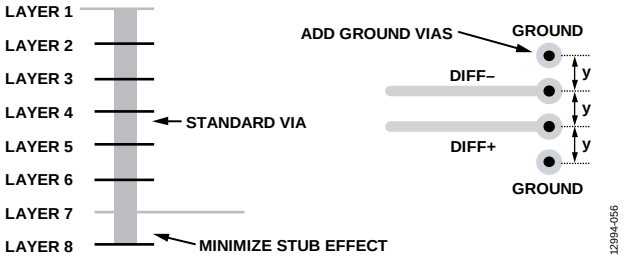


*Figure 74. Minimizing Stub Effect and Adding Ground Vias for Differential Stripline Traces*

### Return Loss

The JESD204B specification limits the amount of return loss allowed in a converter device and a logic device, but does not specify return loss for the channel. However, every effort must be made to maintain a continuous impedance on the transmission line between the transmitting logic device and the AD9152. As mentioned in the Insertion Loss section, minimizing the use of vias, or eliminating them all together, reduces one of the primary sources for impedance mismatches on a transmission line. 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 AD9152 handles this internally with a calibrated termination scheme for the receiving end of the line. See the Interface Power-Up and Input Termination section for details on this circuit and the calibration routine.

### Signal Skew

There are many sources for signal skew, but the two sources to consider when laying out a PCB are interconnect skew within a single JESD204B link and skew between multiple JESD204B links. In each case, keeping the channel lengths matched to within 15 mm is adequate for operating the JESD204B link at speeds of up to 12.38 Gbps. Managing the interconnect skew within a single link is fairly straightforward. Managing multiple links across multiple devices is more complex. However, follow the 15 mm guideline for length matching.

### 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 does not offer more noise immunity than the broadside routing method (traces routed on adjacent layers), it is easier to route and manufacture so that the impedance continuity is maintained. An illustration of the broadside technique vs. the coplanar technique is shown in Figure 75.

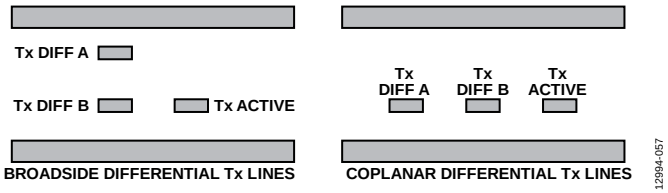


*Figure 75. Broadside vs. Coplanar Differential Stripline Routing Techniques*

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 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 76.

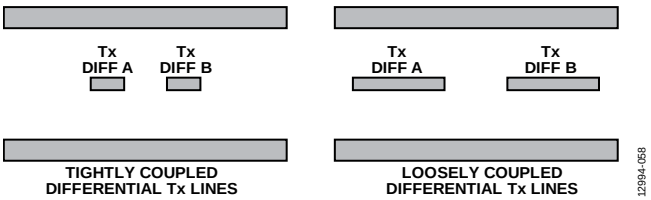


*Figure 76. Tightly Coupled vs. Loosely Coupled Differential Traces*

### AC Coupling Capacitors

The AD9152 requires that the JESD204B input signals be ac-coupled to the source. These capacitors must be 100 nF and placed as close as possible to the transmitting logic device. To minimize the impedance mismatch at the pads, select the

package size of the capacitor so that the pad size on the PCB matches the trace width as closely as possible.

### *SYNCOUT±, SYSREF±, and DACCLK±/REFCLK± Signals*

The SYNCOUT± and SYSREF± signals on the AD9152 are low speed LVDS differential signals. Use controlled impedance traces routed with 100 Ω differential impedance and 50 Ω to ground when routing these signals. As with the SERDIN0± to SERDIN3± 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 SYNCOUT± signal from other noisy signals, because noise on the SYNCOUT± might be interpreted as a request for K characters.

It is important to keep similar trace lengths for the DACCLK±/REFCLK± and SYSREF± signals from the clock source to each of the devices on either end of the JESD204B links, see Figure 77. If using a clock chip that can tightly control the phase of DACCLK±/REFCLK± and SYSREF±, the trace length matching requirements are greatly reduced.

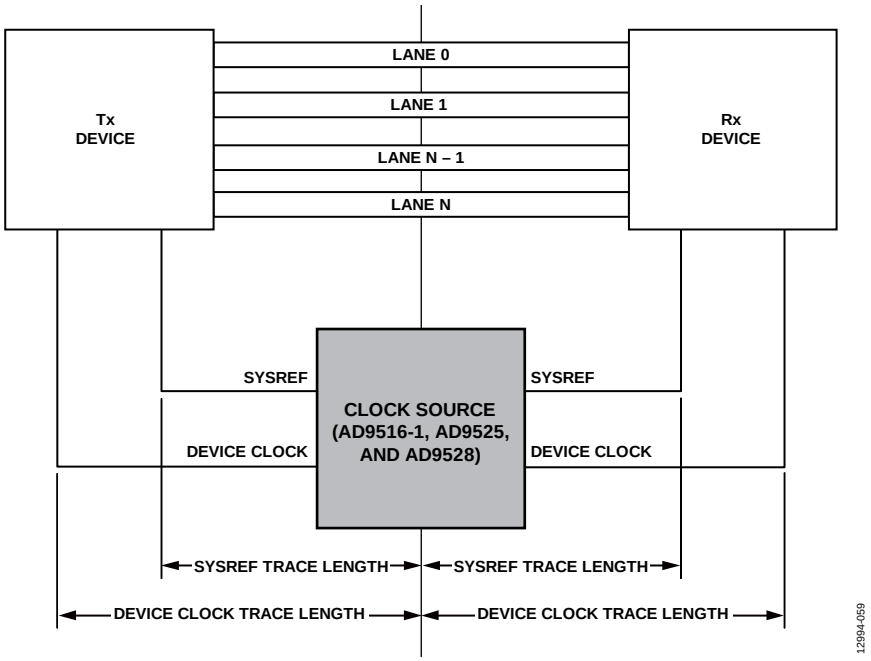


*Figure 77. SYSREF± Signal and Device Clock Trace Length*

# REGISTER MAP AND DESCRIPTIONS

In the following tables, register addresses (Reg. column) and reset values (Reset column) are hexadecimal. In the read/write (R/W) column, R means read only, W means write only, R/W means read/write, and N/A means not applicable.

## DEVICE CONFIGURATION REGISTER MAP

**Table 85. Device Configuration Register Map**

| 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 |
| 0x003 | CHIPTYPE | CHIPTYPE | | | | | | | | 0x04 | R |
| 0x004 | PRODIDL | PRODIDL | | | | | | | | 0x52 | R |
| 0x005 | PRODIDH | PRODIDH | | | | | | | | 0x91 | R |
| 0x006 | CHIPGRADE | RESERVED | | | DEV_REVISION | | | | | 0x08 | R |
| 0x011 | PWRCNTRL0 | PD_BG | PD_DACI | PD_DACQ | PD_DIGCLK | PD_ICLK | PD_QCLK | PD_PCLK | PD_CLKRCVR | 0x6C | R/W |
| 0x012 | TXENMASK | RESERVED | DACA_MASK | RESERVED | CLKA_MASK | RESERVED | | | | 0x00 | R/W |
| 0x013 | PWRCNTRL3 | RESERVED | PDP_PROTECT_OUT | TX_PROTECT_OUT | BSM_PROTECT_OUT | SPI_PROTECT_OUT_EN | SPI_PROTECT_OUT_CTRL | SPI_TXEN_EN | SPI_TXEN_CTRL | 0x20 | R/W |
| 0x014 | PWRCNTRL1 | RESERVED | | POWER_DN_I | POWER_DN_Q | RESERVED | | POWER_UP_I | POWER_UP_Q | 0x00 | R |
| 0x01F | IRQ_ENABLE0 | IRQEN_DACPLLLOST | IRQEN_DACPLLLOCK | RESERVED | IRQEN_SERPLLLOST | IRQEN_SERPLLLOCK | RESERVED | IRQEN_LANEFIFOERR | IRQEN_DRDLFIFOERR | 0x00 | R/W |
| 0x020 | IRQ_ENABLE1 | IRQEN_PARMBAD | IRQEN_LANEFIFO | IRQEN_DLYBUF | IRQEN_DATAREADY | IRQEN_OVERFLOW | RESERVED | IRQEN_PRBSQ | IRQEN_PRBSI | 0x00 | R/W |
| 0x021 | IRQ_ENABLE2 | IRQEN_PDPERR | RESERVED | | IRQEN_NCOALIGN | IRQEN_SYNCLOCK | IRQEN_SYNCROTATE | IRQEN_SYNCWLIM | IRQEN_SYNCTRIP | 0x00 | R/W |
| 0x023 | IRQ_STATUS0 | DACPLLLOST | DACPLLLOCK | RESERVED | SERPLLLOST | SERPLLLOCK | RESERVED | LANEFIFOERR | DRDLFIFOERR | 0x00 | R |
| 0x024 | IRQ_STATUS1 | PARMBAD | LANEFIFO | DLYBUF | DATAREADY | OVERFLOW | RESERVED | PRBSQ | PRBSI | 0x00 | R |
| 0x025 | IRQ_STATUS2 | PAERR | RESERVED | RESERVED | NCOALIGN | SYNCLOCK | SYNCROTATE | SYNCWLIM | SYNCTRIP | 0x00 | R |
| 0x026 | OVERFLOW_STATUS0 | PFIR_OVERFLOW | INT1_OVERFLOW | INT2_OVERFLOW | INT3_OVERFLOW | COARSE_MOD_BY8_OVERFLOW | FINE_MOD_OVERFLOW | PHASE_ADJ_OVERFLOW | GAIN_ADJ_OVERFLOW | 0x00 | R |
| 0x027 | OVERFLOW_STATUS1 | RESERVED | | | | | | | DC_OFFSET_OVERFLOW | 0x00 | R |
| 0x030 | JESD_CHECKS | RESERVED | | ERR_DLYOVER | ERR_WINLIMIT | ERR_JESDBAD | ERR_KUNSUPP | ERR_SUBCLASS | ERR_INTSUPP | 0x00 | R |
| 0x032 | SYNC_DACDELAY_L | DAC_DELAY_L | | | | | | | | 0x00 | R/W |
| 0x033 | SYNC_DACDELAY_H | RESERVED | | | | | | | DAC_DELAY_H | 0x00 | R/W |
| 0x034 | SYNC_ERRWINDOW | RESERVED | | | | | ERRWINDOW | | | 0x00 | R/W |
| 0x035 | SYNC_DLYCOUNT | DLYCOUNT | | | | | | | | 0x00 | R/W |
| 0x036 | SYNC_REFCOUNT | REFCOUNT | | | | | | | | 0x00 | R/W |
| 0x038 | SYNC_LASTERR_L | LASTERROR_L | | | | | | | | 0x00 | R |
| 0x039 | SYNC_LASTERR_H | LASTUNDER | LASTOVER | RESERVED | | | | | LASTERROR_H | 0x00 | R |
| 0x03A | SYNC_CONTROL | SYNCENABLE | SYNCARM | SYNCCLRSTKY | SYNCCLRLAST | SYNCMODE | | | | 0x00 | R/W |
| 0x03B | SYNC_STATUS | SYNCBUSY | RESERVED | | | SYNCLOCK | SYNCROTATE | SYNCWLIM | SYNCTRIP | 0x00 | R |
| 0x03C | SYNC_CURRERR_L | CURRERROR_L | | | | | | | | 0x00 | R |
| 0x03D | SYNC_CURRERR_H | CURRUNDER | CURROVER | RESERVED | | | | | CURRERROR_H | 0x00 | R |
| 0x03E | ERROR_THERM | THRMOLD | RESERVED | | THRMOVER | THRMPOS | THRMZERO | THRMNEG | THRMUNDER | 0x00 | R |
| 0x040 | DAC_GAIN1_I | RESERVED | | | | | | DACFSC_I[9:8] | | 0x03 | R/W |
| 0x041 | DAC_GAIN0_I | DACFSC_I[7:0] | | | | | | | | 0xFF | R/W |
| 0x042 | DAC_GAIN1_Q | RESERVED | | | | | | DACFSC_Q[9:8] | | 0x03 | R/W |
| 0x043 | DAC_GAIN0_Q | DACFSC_Q[7:0] | | | | | | | | 0xFF | R/W |
| 0x047 | COARSE_GROUP_DLY | COARSE_GROUP_DLY_I | | | | COARSE_GROUP_DLY_Q | | | | 0x88 | R/W |
| 0x050 | NCOALIGN_MODE | NCO_ALIGN_ARM | RESERVED | NCO_ALIGN_MTCH | NCO_ALIGN_PASS | NCO_ALIGN_FAIL | RESERVED | NCO_ALIGN_MODE | | 0x00 | R/W |
| 0x051 | NCOKEY_ILSB | NCOKEYI[7:0] | | | | | | | | 0x00 | R/W |

| Reg. | Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Reset | RW |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x052 | NCOKEY_IMSB | NCOKEYI[15:8] | | | | | | | | 0x00 | R/W |
| 0x053 | NCOKEY_QLSB | NCOKEYQ[7:0] | | | | | | | | 0x00 | R/W |
| 0x054 | NCOKEY_QMSB | NCOKEYQ[15:8] | | | | | | | | 0x00 | R/W |
| 0x060 | PDP_THRES0 | PDP_THRESHOLD[7:0] | | | | | | | | 0x00 | R/W |
| 0x061 | PDP_THRES1 | RESERVED | | | PDP_THRESHOLD[12:8] | | | | | 0x00 | R/W |
| 0x062 | PDP_AVG_TIME | PDP_ENABLE | RESERVED | | | PDP_AVG_TIME | | | | 0x00 | R/W |
| 0x063 | PDP_POWER0 | PDP_POWER[7:0] | | | | | | | | 0x00 | R |
| 0x064 | PDP_POWER1 | RESERVED | | | PDP_POWER[12:8] | | | | | 0x00 | R |
| 0x065 | PA_OFFGAIN0 | EN_UKCIRQ-OFFGAIN | RESERVED | | | | | EN_DELAYBUFF-EROFFGAIN | EN_LANEFIFO-OFFGAIN | 0x00 | R/W |
| 0x066 | PA_OFFGAIN1 | EN_CMMIRQ-OFFGAIN | EN_CGSIRQ-GAIN | EN_FSIRQOFF-GAIN | EN_GCSIRQ-OFFGAIN | EN_ILSIRQ-OFFGAIN | EN_ILDIRQOFF-GAIN | EN_DISIRQOFF-GAIN | EN_NITIRQOFF-GAIN | 0x00 | R/W |
| 0x080 | CLKCFG0 | RESERVED | | | | | DUTY_EN | RESERVED | | 0x04 | R/W |
| 0x081 | SYSREF_ACTRL0 | RESERVED | | | PD_SYSREF | HYS_ON | SYSREF_RISE | HYS_CNTRL1 | | 0x10 | R/W |
| 0x082 | SYSREF_ACTRL1 | HYS_CNTRL0 | | | | | | | | 0x00 | R/W |
| 0x083 | DACPLLCNTRL | RECAL_ DACPLL | RESERVED | | ENABLE_ DACPLL | RESERVED | | | | 0x00 | R/W |
| 0x084 | DACPLLSTATUS | CP_OVER-RANGE_H | CP_OVER-RANGE_L | CP_CAL_VALID | VCO_CAL_ PROGRESS | CURRENTS_ READY | RESERVED | DAC_PLL_LOCK | RESERVED | 0x00 | R/W |
| 0x085 | DACINTEGERWORD0 | B_COUNT | | | | | | | | 0x06 | R/W |
| 0x087 | DACLOOPFILT1 | LF_C2_WORD | | | | LF_C1_WORD | | | | 0x88 | R/W |
| 0x088 | DACLOOPFILT2 | LF_R1_WORD | | | | LF_C3_WORD | | | | 0x88 | R/W |
| 0x089 | DACLOOPFILT3 | LF_BYPASS_ R3 | LF_BYPASS_ R1 | LF_BYPASS_C2 | LF_BYPASS_C1 | LF_R3_WORD | | | | 0x08 | R/W |
| 0x08A | DACCPCNTRL | RESERVED | | CP_CURRENT | | | | | | 0x20 | R/W |
| 0x08B | DACLOGENCNTRL | RESERVED | | | | | | LO_DIV_MODE | | 0x00 | R/W |
| 0x08C | DACLDOCNTRL1 | RESERVED | | | | | REF_DIV_MODE | | | 0x00 | R/W |
| 0x08E | CLK_DETECT | RESERVED | | | PD_DAC_ ONDIFF | PD_DAC_ ONDET | CLK_ON | IS_DIFF | CLK_DET_EN | 0x00 | R/W |
| 0x0F7 | DIG_TEST0 | RESERVED | | | | | | DC_TEST_MOD | DIG_CLK_PD | 0x1C | R/W |
| 0x0F8 | DC_TEST_VALUEI0 | DC_TEST_VALUEI[7:0] | | | | | | | | 0x00 | R/W |
| 0x0F9 | DC_TEST_VALUEI1 | DC_TEST_VALUEI[15:8] | | | | | | | | 0x00 | R/W |
| 0x0FA | DC_TEST_VALUEQ0 | DC_TEST_VALUEQ[7:0] | | | | | | | | 0x00 | R/W |
| 0x0FB | DC_TEST_VALUEQ1 | DC_TEST_VALUEQ[15:8] | | | | | | | | 0x00 | R/W |
| 0x110 | DATA_FORMAT | BINARY_ FORMAT | RESERVED | | | | | | | 0x00 | R/W |
| 0x111 | DATAPATH_CTRL | INVSINC_ ENABLE | RESERVED | DIG_GAIN_ ENABLE | PHASE_ADJ_ ENABLE | MODULATION_TYPE | | SEL_SIDEBAND | RESERVED | 0x20 | R/W |
| 0x112 | INTERP_MODE | SINGLE_ DAC_EN | RESERVED | | | | | INTERP_MODE | | 0x01 | R/W |
| 0x113 | NCO_FTW_UPDATE | RESERVED | | | | | | FTW_UPDATE_ ACK | FTW_UPDATE_ REQ | 0x00 | R/W |
| 0x114 | FTW0 | FTW[7:0] | | | | | | | | 0x00 | R/W |
| 0x115 | FTW1 | FTW[15:8] | | | | | | | | 0x00 | R/W |
| 0x116 | FTW2 | FTW[23:16] | | | | | | | | 0x00 | R/W |
| 0x117 | FTW3 | FTW[31:24] | | | | | | | | 0x00 | R/W |
| 0x118 | FTW4 | FTW[39:32] | | | | | | | | 0x00 | R/W |
| 0x119 | FTW5 | FTW[47:40] | | | | | | | | 0x10 | R/W |
| 0x11A | NCO_PHASE_ OFFSET0 | NCO_PHASE_OFFSET[7:0] | | | | | | | | 0x00 | R/W |
| 0x11B | NCO_PHASE_ OFFSET1 | NCO_PHASE_OFFSET[15:8] | | | | | | | | 0x00 | R/W |
| 0x11C | IQ_PHASE_ADJ0 | PHASE_ADJ[7:0] | | | | | | | | 0x00 | R/W |
| 0x11D | IQ_PHASE_ADJ1 | RESERVED | | | PHASE_ADJ[12:8] | | | | | 0x00 | R/W |

| Reg. | Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Reset | RW |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x11F | TXEN_SM_0 | FALL_COUNTERS | | RISE_COUNTERS | | RESERVED | PROTECT_OUT_INVERT | RESERVED | TXEN_SM_EN | 0x83 | R/W |
| 0x125 | DACOUT_ON_DOWN | RESERVED | | | | | | DACOUT_SHUTDOWN | DACOUT_ON_TRIGGER | 0x00 | R/W |
| 0x12C | DACOFF | PROTECT_MODE | RESERVED | | | | | | DACOFF_AVG_PW | 0x81 | R/W |
| 0x12F | DIE_TEMP_CTRL0 | RESERVED | FS_CURRENT | | | RESERVED | | | TEMP_SENSOR_ENABLE | 0x20 | R/W |
| 0x132 | DIE_TEMP0 | DIE_TEMP[7:0] | | | | | | | | 0x00 | R |
| 0x133 | DIE_TEMP1 | DIE_TEMP[15:8] | | | | | | | | 0x00 | R |
| 0x134 | DIE_TEMP_UPDATE | RESERVED | | | | | | | DIE_TEMP_UPDATE | 0x00 | R/W |
| 0x135 | DC_OFFSET_CTRL | RESERVED | | | | | | | DC_OFFSET_ON | 0x00 | R/W |
| 0x136 | IPATH_DC_OFFSET_1PART0 | LSB_OFFSET_I[7:0] | | | | | | | | 0x00 | R/W |
| 0x137 | IPATH_DC_OFFSET_1PART1 | LSB_OFFSET_I[15:8] | | | | | | | | 0x00 | R/W |
| 0x138 | QPATH_DC_OFFSET_1PART0 | LSB_OFFSET_Q[7:0] | | | | | | | | 0x00 | R/W |
| 0x139 | QPATH_DC_OFFSET_1PART1 | LSB_OFFSET_Q[15:8] | | | | | | | | 0x00 | R/W |
| 0x13A | IPATH_DC_OFFSET_2PART | RESERVED | | | SIXTEENTH_OFFSET_I | | | | | 0x00 | R/W |
| 0x13B | QPATH_DC_OFFSET_2PART | RESERVED | | | SIXTEENTH_OFFSET_Q | | | | | 0x00 | R/W |
| 0x13C | IDAC_DIG_GAIN0 | IDAC_DIG_GAIN[7:0] | | | | | | | | 0x00 | R/W |
| 0x13D | IDAC_DIG_GAIN1 | RESERVED | | | | IDAC_DIG_GAIN[11:8] | | | | 0x08 | R/W |
| 0x13E | QDAC_DIG_GAIN0 | QDAC_DIG_GAIN[7:0] | | | | | | | | 0x00 | R/W |
| 0x13F | QDAC_DIG_GAIN1 | RESERVED | | | | QDAC_DIG_GAIN[11:8] | | | | 0x08 | R/W |
| 0x140 | GAIN_RAMP_UP_STEP0 | GAIN_RAMP_UP_STEP[7:0] | | | | | | | | 0x04 | R/W |
| 0x141 | GAIN_RAMP_UP_STEP1 | RESERVED | | | | GAIN_RAMP_UP_STEP[11:8] | | | | 0x00 | R/W |
| 0x142 | GAIN_RAMP_DOWN_STEP0 | GAIN_RAMP_DOWN_STEP[7:0] | | | | | | | | 0x09 | R/W |
| 0x143 | GAIN_RAMP_DOWN_STEP1 | RESERVED | | | | GAIN_RAMP_DOWN_STEP[7:0] | | | | 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 |
| 0x151 | DATAPATH_CTRL2 | RESERVED | | PFIR_DEMOD4_ENABLE | PFIR_ENABLE | RESERVED | | NEG_DDS_FREQ | MODULUS_ENABLE | 0x00 | R/W |
| 0x152 | ACC_MODULUS0 | ACC_MODULUS[7:0] | | | | | | | | 0x00 | R/W |
| 0x153 | ACC_MODULUS1 | ACC_MODULUS[15:8] | | | | | | | | 0x00 | R/W |
| 0x154 | ACC_MODULUS2 | ACC_MODULUS[23:16] | | | | | | | | 0x00 | R/W |
| 0x155 | ACC_MODULUS3 | ACC_MODULUS[31:24] | | | | | | | | 0x00 | R/W |
| 0x156 | ACC_MODULUS4 | ACC_MODULUS[39:32] | | | | | | | | 0x00 | R/W |
| 0x157 | ACC_MODULUS5 | ACC_MODULUS[47:40] | | | | | | | | 0x00 | R/W |
| 0x158 | ACC_DELTA0 | ACC_DELTA[7:0] | | | | | | | | 0x00 | R/W |
| 0x159 | ACC_DELTA1 | ACC_DELTA[15:8] | | | | | | | | 0x00 | R/W |
| 0x15A | ACC_DELTA2 | ACC_DELTA[23:16] | | | | | | | | 0x00 | R/W |
| 0x15B | ACC_DELTA3 | ACC_DELTA[31:24] | | | | | | | | 0x00 | R/W |
| 0x15C | ACC_DELTA4 | ACC_DELTA[39:32] | | | | | | | | 0x00 | R/W |
| 0x15D | ACC_DELTA5 | ACC_DELTA[47:40] | | | | | | | | 0x00 | R/W |
| 0x17A | PFIR_COEFF0_L | PFIR_COEFF0[7:0] | | | | | | | | 0x00 | R/W |
| 0x17B | PFIR_COEFF0_H | RESERVED | | | | | | | PFIR_COEFF0[8] | 0x00 | R/W |

| Reg. | Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Reset | RW |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x17C | PFIR_COEFF1_L | PFIR_COEFF1[7:0] | | | | | | | | 0x00 | R/W |
| 0x17D | PFIR_COEFF1_H | RESERVED | | | | | | | PFIR_COEFF1[8] | 0x00 | R/W |
| 0x17E | PFIR_COEFF2_L | PFIR_COEFF2[7:0] | | | | | | | | 0x00 | R/W |
| 0x17F | PFIR_COEFF2_H | RESERVED | | | | | | | PFIR_COEFF2[8] | 0x00 | R/W |
| 0x180 | PFIR_COEFF3_L | PFIR_COEFF3[7:0] | | | | | | | | 0x00 | R/W |
| 0x181 | PFIR_COEFF3_H | RESERVED | | | | | | | PFIR_COEFF3[8] | 0x00 | R/W |
| 0x182 | PFIR_COEFF_ UPDATE | RESERVED | | | | | | | PFIR_COEFF_ UPDATE | 0x00 | R/W |
| 0x1B4 | DACPLLT4 | BYP_LOAD_ DELAY | VCO_CAL_OFFSET | | | | RESERVED | EXT_BAND_EN | EXT_BAND2 | 0x78 | R/W |
| 0x1B5 | DACPLLT5 | INIT_ALC_VALUE | | | | VCO_VAR | | | | 0x83 | R/W |
| 0x1B6 | DACPLLT6 | RESERVED | PORESETB_ VCO | EXT_VCO_BITSEL | | VCO_LVL_OUT | | | | 0x4A | R/W |
| 0x1BB | DACPLLTB | RESERVED | | | VCO_BIAS_TCF | | VCO_BIAS_REF | | | 0x0C | R/W |
| 0x1C5 | DACPLLT18 | RESERVED | | | | VCO_VAR_REF | | | | 0x08 | R/W |
| 0X1FE | TEST_MODE | RESERVED | | | | | TSTWINDOW | | | 0x00 | R/W |
| 0x200 | MASTER_PD | RESERVED | | | | | | | SPI_PD_MASTER | 0x01 | R/W |
| 0x201 | PHY_PD | RESERVED | | | | SPI_PD_PHY | | | | 0x00 | R/W |
| 0x203 | GENERIC_PD | RESERVED | | | | | | SPI_SYNC_PD | RESERVED | 0x00 | R/W |
| 0x206 | CDR_RESET | RESERVED | | | | | | | SPI_CDR_RESETN | 0x01 | R/W |
| 0x230 | CDR_OPERATING_ MODE_REG_0 | RESERVED | | ENHALFRATE | RESERVED | | | CDR_OVERSAMP | SYNCOUTB_ SWING | 0x28 | R/W |
| 0x268 | EQ_BIAS_REG | EQ_POWER_MODE | | RESERVED | | | | | | 0x62 | R/W |
| 0x280 | SERDESPLL_ ENABLE_CNTRL | RESERVED | | | | | RECAL_ SERDESPLL | RESERVED | ENABLE_ SERDESPLL | 0x00 | R/W |
| 0x281 | SERDES_PLL_ STATUS | RESERVED | | SERDES_CP_ OVER_RANGE_H | SERDES_CP_ OVER_RANGE_L | SERDES_PLL_ CAL_VALID | SERDES_VCO_ CAL_PROGRESS | SERDES_PLL_ CURRENTS_ READY | SERDES_PLL_ LOCK | 0x00 | R |
| 0x289 | REF_CLK_DIVIDER_ LDO | RESERVED | | | | | | SERDES_PLL_DIV_MODE | | 0x04 | R/W |
| 0x2A7 | TERM_BLK1_ CTRLREG0 | RESERVED | | | | | | | SPI_I_TUNE_R_ CAL_TERMBLK1 | 0x00 | R/W |
| 0x300 | GENERAL_JRX_ CTRL_0 | RESERVED | CHECKSUM_ MODE | RESERVED | | | | | LINK_EN | 0x00 | R/W |
| 0x301 | GENERAL_JRX_ CTRL_1 | RESERVED | | | | | SUBCLASSV_LOCAL | | | 0x01 | R/W |
| 0x302 | DYN_LINK_ LATENCY_0 | RESERVED | | | DYN_LINK_LATENCY_0 | | | | | 0x00 | R/W |
| 0x304 | LMFC_DELAY_0 | RESERVED | | | LMFC_DELAY_0 | | | | | 0x00 | R/W |
| 0x306 | LMFC_VAR_0 | RESERVED | | | LMFC_VAR_0 | | | | | 0x06 | 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 |
| 0x30C | FIFO_STATUS_REG_0 | RESERVED | | | | LANE_FIFO_FULL | | | | 0x00 | R |
| 0x30D | FIFO_STATUS_REG_1 | RESERVED | | | | LANE_FIFO_EMPTY | | | | 0x00 | R |
| 0x311 | SYNCB_GEN_0 | RESERVED | | | | RESERVED | EOMF_MASK_0 | RESERVED | EOF_MASK_0 | 0x00 | R/W |
| 0x312 | SYNCB_GEN_1 | SYNCB_ERR_DUR | | | | SYNCB_SYNCREQ_DUR | | | | 0x00 | R/W |
| 0x313 | SYNCB_GEN_3 | LMFC_PERIOD | | | | | | | | 0x00 | R |
| 0x314 | SERDES_SPI_REG | SERDES_SPI_CONFIG | | | | | | | | 0x00 | R/W |
| 0x315 | PHY_PRBS_TEST_EN | RESERVED | | | | 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[7:0] | | | | | | | | 0x00 | R/W |
| 0x318 | PHY_PRBS_TEST_ THRESHOLD_ MIDBITS | PHY_PRBS_THRESHOLD[15:8] | | | | | | | | 0x00 | R/W |

| Reg. | Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Reset | RW |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x319 | PHY_PRBS_TEST_THRESHOLD_HIBITS | PHY_PRBS_THRESHOLD[23:16] | | | | | | | | 0x00 | R/W |
| 0x31A | PHY_PRBS_TEST_ERRCNT_LOBITS | PHY_PRBS_ERR_CNT[7:0] | | | | | | | | 0x00 | R |
| 0x31B | PHY_PRBS_TEST_ERRCNT_MIDBITS | PHY_PRBS_ERR_CNT[15:8] | | | | | | | | 0x00 | R |
| 0x31C | PHY_PRBS_TEST_ERRCNT_HIBITS | PHY_PRBS_ERR_CNT[23:16] | | | | | | | | 0x00 | R |
| 0x31D | PHY_PRBS_TEST_STATUS | RESERVED | | | | PHY_PRBS_PASS | | | | 0x0F | R |
| 0x32C | SHORT_TPL_TEST_0 | RESERVED | | SHORT_TPL_SP_SEL | | SHORT_TPL_DAC_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_FAIL | 0x00 | R |
| 0x334 | JESD_BIT_INVERSE_CTRL | RESERVED | | | | JESD_BIT_INVERSE | | | | 0x00 | R/W |
| 0x400 | DID_REG | DID_RD | | | | | | | | 0x00 | R |
| 0x401 | BID_REG | ADJCNT_RD | | | | BID_RD | | | | 0x00 | R |
| 0x402 | LID0_REG | RESERVED | ADJDIR_RD | PHADJ_RD | LID0_RD | | | | | 0x00 | R |
| 0x403 | SCR_L_REG | SCR_RD | RESERVED | | L-1_RD | | | | | 0x00 | R |
| 0x404 | F_REG | F-1_RD | | | | | | | | 0x00 | R |
| 0x405 | K_REG | RESERVED | | | K-1_RD | | | | | 0x00 | R |
| 0x406 | M_REG | M-1_RD | | | | | | | | 0x00 | R |
| 0x407 | CS_N_REG | CS_RD | | RESERVED | N-1_RD | | | | | 0x00 | R |
| 0x408 | NP_REG | SUBCLASSV_RD | | | NP-1_RD | | | | | 0x00 | R |
| 0x409 | S_REG | JESDV_RD | | | S-1_RD | | | | | 0x00 | R |
| 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 | CHECKSUM_REG | FCHK0_RD | | | | | | | | 0x00 | R |
| 0x40E | COMPSUM0_REG | FCMP0_RD | | | | | | | | 0x00 | R |
| 0x412 | LID1_REG | RESERVED | | | LID1_RD | | | | | 0x00 | R |
| 0x415 | CHECKSUM1_REG | FCHK1_RD | | | | | | | | 0x00 | R |
| 0x416 | COMPSUM1_REG | FCMP1_RD | | | | | | | | 0x00 | R |
| 0x41A | LID2_REG | RESERVED | | | LID2_RD | | | | | 0x00 | R |
| 0x41D | CHECKSUM2_REG | FCHK2_RD | | | | | | | | 0x00 | R |
| 0x41E | COMPSUM2_REG | FCMP2_RD | | | | | | | | 0x00 | R |
| 0x422 | LID3_REG | RESERVED | | | LID3_RD | | | | | 0x00 | R |
| 0x425 | CHECKSUM3_REG | FCHK3_RD | | | | | | | | 0x00 | R |
| 0x426 | COMPSUM3_REG | FCMP3_RD | | | | | | | | 0x00 | R |
| 0x450 | ILS_DID | DID | | | | | | | | 0x00 | R/W |
| 0x451 | ILS_BID | ADJCNT | | | | BID | | | | 0x00 | R/W |
| 0x452 | ILS_LID0 | RESERVED | ADJDIR | PHADJ | LID0 | | | | | 0x00 | R/W |
| 0x453 | ILS_SCR_L | SCR | RESERVED | | L-1 | | | | | 0x83 | 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 |
| 0x458 | ILS_NP | SUBCLASSV | | | NP-1 | | | | | 0x2F | R/W |
| 0x459 | ILS_S | JESDV | | | S-1 | | | | | 0x20 | R/W |
| 0x45A | ILS_HD_CF | HD | RESERVED | | CF | | | | | 0x80 | R/W |
| 0x45B | ILS_RES1 | RES1 | | | | | | | | 0x00 | R/W |

| Reg. | Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Reset | RW |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x45C | ILS_RES2 | RES2 | | | | | | | | 0x00 | R/W |
| 0x45D | ILS_CHECKSUM | FCHK0 | | | | | | | | 0x45 | R/W |
| 0x46B | ERRCNTRMON_RB | READERRORCNTR | | | | | | | | 0x00 | R |
| 0x46B | ERRCNTRMON | RESERVED | LANESEL | | | RESERVED | | CNTRSEL | | 0x00 | W |
| 0x46C | LANEDESKEW | LANEDESKEW | | | | | | | | 0x0F | R/W |
| 0x46D | BADDISPARITY_RB | BADDIS | | | | | | | | 0x00 | R |
| 0x46D | BADDISPARITY | RST_IRQ_DIS | DISABLE_ERR_CNTR_DIS | RST_ERR_CNTR_DIS | RESERVED | | LANE_ADDR_DIS | | | 0x00 | W |
| 0x46E | NIT_RB | NIT | | | | | | | | 0x00 | R |
| 0x46E | NIT_W | RST_IRQ_NIT | DISABLE_ERR_CNTR_NIT | RST_ERR_CNTR_NIT | RESERVED | | LANE_ADDR_NIT | | | 0x00 | W |
| 0x46F | UNEXPECTED_CONTROL_RB | UCC | | | | | | | | 0x00 | R |
| 0x46F | UNEXPECTED_CONTROL_W | RST_IRQ_UCC | DISABLE_ERR_CNTR_UCC | RST_ERR_CNTR_UCC | RESERVED | | LANE_ADDR_UCC | | | 0x00 | W |
| 0x470 | CODEGRPSYNCFLG | CODEGRPSYNC | | | | | | | | 0x00 | R/W |
| 0x471 | FRAMESYNCFLG | FRAMESYNC | | | | | | | | 0x00 | R/W |
| 0x472 | GOODCHKSUMFLG | GOODCHECKSUM | | | | | | | | 0x00 | R/W |
| 0x473 | INITLANESYNCFLG | INITIALLANESYNC | | | | | | | | 0x00 | R/W |
| 0x476 | CTRLREG1 | F | | | | | | | | 0x01 | R/W |
| 0x477 | CTRLREG2 | ILAS_MODE | RESERVED | | | THRESHOLD_MASK_EN | RESERVED | | | 0x00 | R/W |
| 0x478 | KVAL | KSYNC | | | | | | | | 0x01 | R/W |
| 0x47A | IRQVECTOR_FLAG | BADDIS_FLAG | NIT_FLAG | UCC_FLAG | RESERVED | INITIALLANE-SYNC_FLAG | BADCHECKSUM_FLAG | FRAMESYNC_FLAG | CODEGRPSYNC_FLAG | 0x00 | R |
| 0x47A | IRQVECTOR_MASK | BADDIS_MASK | NIT_MASK | UCC_MASK | RESERVED | INITIALLANE-SYNC_MASK | BADCHECKSUM_MASK | FRAMESYNC_MASK | CODEGRPSYNC_MASK | 0x00 | W |
| 0x47B | SYNCASSERTION-MASK | BADDIS_S | NIT_S | UCC_S | CMM | CMM_ENABLE | RESERVED | | | 0x08 | R/W |
| 0x47C | ERRORTHRES | ETH | | | | | | | | 0xFF | R/W |
| 0x47D | LANEENABLE | RESERVED | | | | LANE_ENA | | | | 0x0F | R/W |
| 0x47E | RAMP_ENA | RESERVED | | | | | | | ENA_RAMP_CHECK | 0x00 | R/W |

## DEVICE CONFIGURATION REGISTER DESCRIPTIONS

**Table 86. Device Configuration Register Descriptions**

| Addr. | Name | Bits | Bit Name | Settings | Description | Reset | Access |
|---|---|---|---|---|---|---|---|
| 0x000 | SPI_INTFCONFA | 7 | SOFTRESET_M | | Soft reset (mirror). | 0x0 | R |
| | | 6 | LSBFIRST_M | | LSB first (mirror). | 0x0 | R |
| | | 5 | ADDRINC_M | | Address increment (mirror). | 0x0 | R |
| | | 4 | SDOACTIVE_M | | SDO active (mirror). | 0x0 | R |
| | | 3 | SDOACTIVE | | SDO active. | 0x0 | R/W |
| | | 2 | ADDRINC | | Address increment. When set, causes incrementing streaming addresses; otherwise descending addresses are generated. | 0x0 | R/W |
| | | | | 1 | During streaming bytes mode (multibyte), the addresses are incremented. | | |
| | | | | 0 | During streaming bytes mode (multibyte), the addresses are decremented. | | |
| | | 1 | LSBFIRST | | LSB first. When set, causes input and output data to be oriented as LSB first. If this bit is clear, data is oriented as MSB first. | 0x0 | R/W |
| | | | | 1 | Shift LSB in first. | | |
| | | | | 0 | Shift MSB in first. | | |
| | | 0 | SOFTRESET | | Soft reset. Setting this bit initiates a reset. This bit is auto-clearing after the soft reset is complete. | 0x0 | R/W |
| | | | | 1 | Assert soft reset. | | |
| 0x003 | CHIPTYPE | [7:0] | CHIPTYPE | | The product type is high speed DAC, which is | 0x4 | R |

| Addr. | Name | Bits | Bit Name | Settings | Description | Reset | Access |
|---|---|---|---|---|---|---|---|
| | | | | | represented by a code of 0x04. | | |
| 0x004 | PRODIDL | [7:0] | PRODIDL | | Product ID low. | 0x52 | R |
| 0x005 | PRODIDH | [7:0] | PRODIDH | | Product ID high. | 0x91 | R |
| 0x006 | CHIPGRADE | [7:5] | RESERVED | | Reserved. | 0x0 | R |
| | | [4:0] | DEV_REVISION | | Device revision. | 0x8 | R |
| 0x011 | PWRCNTRL0 | 7 | PD_BG | | Reference power-down. Powers down the band gap reference for the entire chip. Circuits are not provided with bias currents. | 0x0 | R/W |
| | | | | 0 | Reference on. | | |
| | | | | 1 | Reference powered down (overrides TXEN masked bit). | | |
| | | 6 | PD_DAC_I | | Power down DAC bias for I DAC. | 0x1 | R/W |
| | | | | 1 | Power down I DAC. | | |
| | | 5 | PD_DAC_Q | | Power down DAC bias for Q DAC. | 0x1 | R/W |
| | | | | 1 | Power down Q DAC. | | |
| | | 4 | PD_DIGCLK | | Power down digital clock. | 0x0 | R/W |
| | | 3 | PD_ICLK | | Power down DAC clock For I DAC. | 0x1 | R/W |
| | | 2 | PD_QCLK | | Power down DAC clock For Q DAC. | 0x1 | R/W |
| | | 1 | PD_PCLK | | Power down PCLK. | 0x0 | R/W |
| | | 0 | PD_CLKRCVR | | Power down clock receiver. | 0x0 | R/W |
| 0x012 | TXENMASK | 7 | RESERVED | | Reserved. | 0x0 | R/W |
| | | 6 | DACA_MASK | | DAC power-down mask for TXEN. | 0x0 | R/W |
| | | | | 1 | If TXEN is low, DAC I and DAC Q are powered down. | | |
| | | 5 | RESERVED | | Reserved. | 0x0 | R/W |
| | | 4 | CLKA_MASK | | DAC clock power-down mask from TXEN. | 0x0 | R/W |
| | | | | 1 | If TXEN is low digital clocks are powered down. | | |
| | | [3:0] | RESERVED | | Reserved. | 0x0 | R/W |
| 0x013 | PWRCNTRL3 | 7 | RESERVED | | Reserved. | 0x0 | R/W |
| | | 6 | PDP_PROTECT_OUT | 1 | PDP_PROTECT triggers PROTECT_OUT. | 0x0 | R/W |
| | | 5 | TX_PROTECT_OUT | 1 | TX_PROTECT triggers PROTECT_OUT. | 0x1 | R/W |
| | | 4 | BSM_PROTECT_OUT | 1 | BSM_PROTEXT triggers PROTECT_OUT. | 0x0 | R/W |
| | | 3 | SPI_PROTECT_OUT_EN | 1 | SPI_PROTECT triggers PROTECT_OUT. | 0x0 | R/W |
| | | 2 | SPI_PROTECT_OUT_CTRL | 1 | PROTECT_OUT is low. | 0x0 | R/W |
| | | 1 | SPI_TXEN_EN | 1 | TXEN is controlled by SPI. | 0x0 | R/W |
| | | 0 | SPI_TXEN_CTRL | 1 | TXEN SPI is high. | 0x0 | R/W |
| 0x014 | PWRCNTRL1 | [7:6] | RESERVED | | Reserved. | 0x0 | R |
| | | 5 | POWER_DN_I | | I DAC power-down status. | 0x0 | R |
| | | | | 1 | DAC is fully powered down. | | |
| | | 4 | POWER_DN_Q | | Q DAC power-down status. | 0x0 | R |
| | | | | 1 | DAC is fully powered down. | | |
| | | [3:2] | RESERVED | | Reserved. | 0x0 | R |
| | | 1 | POWER_UP_I | | I DAC power-up status. | 0x0 | R |
| | | | | 1 | DAC is fully powered up. | | |
| | | 0 | POWER_UP_Q | | Q DAC power-up status. | 0x0 | R |
| | | | | 1 | DAC is fully powered up. | | |
| 0x01F | IRQ_ENABLE0 | 7 | IRQEN_DACPLLLOST | | Enable the IRQ of DAC PLL lost detection. | 0x0 | R/W |
| | | | | 1 | If IRQEN_DACPLLLOST goes high, it latches and pulls $\overline{IRQ}$ low. | | |
| | | | | 0 | IRQEN_DACPLLLOST shows current status. | | |
| | | 6 | IRQEN_DACPLLLOCK | | Enable the IRQ of DAC PLL lock detection. | 0x0 | R/W |
| | | | | 1 | If IRQEN_DACPLLLOCK goes high, it latches and pulls $\overline{IRQ}$ low. | | |
| | | | | 0 | IRQEN_DACPLLLOCK shows current status. | | |
| | | 5 | RESERVED | | Reserved. | 0x0 | R/W |
| | | 4 | IRQEN_SERPLLLOST | | Enable the IRQ of SERDES PLL lost detection. | 0x0 | R/W |
| | | | | 1 | If IRQEN_SERPLLLOST goes high, it latches and | | |

| Addr. | Name | Bits | Bit Name | Settings | Description | Reset | Access |
|---|---|---|---|---|---|---|---|
| | | | | | pulls IRQ low. | | |
| | | | | 0 | IRQEN_SERPLLLOST shows current status. | | |
| | | 3 | IRQEN_SERPLLLOCK | | Enable the IRQ of SERDES PLL lock detection. | 0x0 | R/W |
| | | | | 1 | If IRQEN_SERPLLLOCK goes high, it latches and pulls IRQ low. | | |
| | | | | 0 | IRQEN_SERPLLLOCK shows current status. | | |
| | | 2 | RESERVED | | Reserved. | 0x0 | R/W |
| | | 1 | IRQEN_LANEFIFOERR | | Enable the IRQ of lane FIFO error detection. | 0x0 | R/W |
| | | | | 1 | If IRQEN_LANEFIFOERR goes high, it latches and pulls IRQ low. | | |
| | | | | 0 | IRQEN_LANEFIFOERR shows current status. | | |
| | | 0 | IRQEN_DRDLFIFOERR | | Enable the IRQ of DRDL FIFO error detection. | 0x0 | R/W |
| | | | | 1 | If IRQEN_DRDLFIFOERR goes high, it latches and pulls IRQ low. | | |
| | | | | 0 | IRQEN_DRDLFIFOERR shows current status. | | |
| 0x020 | IRQ_ENABLE1 | 7 | IRQEN_PARMBAD | | Enable the interrupt of bad parameter. | 0x0 | R/W |
| | | | | 1 | If IRQEN_PARMBAD goes high, it latches and pulls IRQ low. | | |
| | | | | 0 | IRQEN_PARMBAD shows current status. | | |
| | | 6 | IRQEN_LANEFIFO | | Enable the interrupt of lane FIFO empty/full. | 0x0 | R/W |
| | | | | 1 | If IRQEN_LANEFIFO goes high, it latches and pulls IRQ low. | | |
| | | | | 0 | IRQEN_LANEFIFO shows current status. | | |
| | | 5 | IRQEN_DLYBUF | | Enable the interrupt of delay buffer empty/full. | 0x0 | R/W |
| | | | | 1 | If IRQEN_DLYBUF goes high, it latches and pulls IRQ low. | | |
| | | | | 0 | IRQEN_DLYBUF shows current status. | | |
| | | 4 | IRQEN_DATAREADY | | Enable the interrupt of data ready. | 0x0 | R/W |
| | | | | 1 | If IRQEN_DATAREADY goes high, it latches and pulls IRQ low. | | |
| | | | | 0 | IRQEN_DATAREADY shows current status. | | |
| | | 3 | IRQEN_OVERFLOW | | Enable the interrupt of data path modules overflow. | 0x0 | R/W |
| | | | | 1 | If IRQEN_OVERFLOW goes high, it latches and pulls IRQ low. | | |
| | | | | 0 | IRQEN_OVERFLOW shows current status. | | |
| | | 2 | RESERVED | | Reserved. | 0x0 | R/W |
| | | 1 | IRQEN_PRBSQ | | Enable the interrupt of Q DAC PRBS. | 0x0 | R/W |
| | | | | 1 | If IRQEN_PRBSQ goes high, it latches and pulls IRQ low. | | |
| | | | | 0 | IRQEN_PRBSQ shows current status. | | |
| | | 0 | IRQEN_PRBSI | | Enable the interrupt of Q DAC PRBS. | 0x0 | R/W |
| | | | | 1 | If IRQEN_ PRBSI goes high, it latches and pulls IRQ low. | | |
| | | | | 0 | IRQEN_PRBSI shows current status. | | |
| 0x021 | IRQ_ENABLE2 | 7 | IRQEN_PDPERR | | Enable the interrupt of PDP error. | 0x0 | R/W |
| | | | | 1 | If IRQEN_PDPERR goes high, it latches and pulls IRQ low. | | |
| | | | | 0 | IRQEN_PDPERR shows current status. | | |
| | | [6:5] | RESERVED | | Reserved. | 0x0 | R |
| | | 4 | IRQEN_NCOALIGN | | Enable the interrupt of NCO alignment. | 0x0 | R/W |
| | | | | 1 | If IRQEN_NCOALIGN goes high, it latches and pulls IRQ low. | | |
| | | | | 0 | IRQEN_NCOALIGN shows current status. | | |
| | | 3 | IRQEN_SYNCLOCK | | Enable the interrupt of link alignment lock. | 0x0 | R/W |
| | | | | 1 | If IRQEN_SYNCLOCK goes high, it latches and pulls IRQ low. | | |
| | | | | 0 | IRQEN_SYNCLOCK shows current status. | | |
| | | 2 | IRQEN_SYNCROTATE | | Enable the interrupt of link alignment rotate. | 0x0 | R/W |

| Addr. | Name | Bits | Bit Name | Settings | Description | Reset | Access |
|---|---|---|---|---|---|---|---|
| | | | | 1 | If IRQEN_SYNCROTATE goes high, it latches and pulls IRQ low. | | |
| | | | | 0 | IRQEN_SYNCROTATE shows current status. | | |
| | | 1 | IRQEN_SYNCWLIM | | Enable the interrupt of link alignment outside limit window. | 0x0 | R/W |
| | | | | 1 | If IRQEN_SYNCWLIM goes high, it latches and pulls IRQ low. | | |
| | | | | 0 | IRQEN_SYNCWLIM shows current status. | | |
| | | 0 | IRQEN_SYNCTRIP | | Enable the interrupt of link alignment tripped. | 0x0 | R/W |
| | | | | 1 | If SYNCTRIP goes high, it latches and pulls IRQ low. | | |
| | | | | 0 | SYNCTRIP shows current status. | | |
| 0x023 | IRQ_STATUS0 | 7 | DACPLLLOST | | DAC PLL lost status. If IRQEN _DACPLLLOST is low, this bit shows current status. If not, this bit latches on a rising edge and pulls IRQ low. When latched, write a 1 to clear this bit. | 0x0 | R/W |
| | | | | 1 | DAC PLL lock was lost. | | |
| | | 6 | DACPLLLOCK | | DAC PLL lock status. If IRQEN _DACPLLLOCK is low, this bit shows current status. If not, this bit latches on a rising edge and pulls IRQ low. When latched, write a 1 to clear this bit. | 0x0 | R/W |
| | | | | 1 | DAC PLL locked. | | |
| | | 5 | RESERVED | | Reserved. | 0x0 | R |
| | | 4 | SERPLLLOST | | SERDES PLL lost status. If IRQEN _SERPLLLOST is low, this bit shows current status. If not, this bit latches on a rising edge and pulls IRQ low. When latched, write a 1 to clear this bit. | 0x0 | R/W |
| | | | | 1 | SERDES PLL lock was lost. | | |
| | | 3 | SERPLLLOCK | | SERDES PLL lock status. If IRQEN _SERPLLLOCK is low, this bit shows current status. If not, this bit latches on a rising edge and pulls IRQ low. When latched, write a 1 to clear this bit. | 0x0 | R/W |
| | | | | 1 | SERDES PLL locked. | | |
| | | 2 | RESERVED | | Reserved. | 0x0 | R |
| | | 1 | LANEFIFOERR | | Lane FIFO error status. If IRQEN_LANEFIFOERR is low, this bit shows current status. If not, this bit latches on a rising edge and pulls IRQ low.<br>A lane FIFO error occurs when there is a full or empty condition on any of the FIFOs between the deserializer block and the digital core. This error requires a link disable and reenable to remove it. The status of the lane FIFOs can be found in Register 0x30C (FIFO full), and Register 0x30D (FIFO empty). | 0x0 | R/W |
| | | | | 1 | Lane FIFO error. | | |
| | | 0 | DRDLFIFOERR | | DRDL FIFO status. If IRQEN_DRDLFIFOERR is low, this bit shows current status. If not, this bit latches on a rising edge and pulls IRQ low. When latched, write a 1 to clear this bit. | 0x0 | R/W |
| | | | | 1 | DRDL FIFO error. | | |
| 0x024 | IRQ_STATUS1 | 7 | PARMBAD | | BAD parameter status. If IRQEN_PARMBAD is low, this bit shows current status. If not, this bit latches on a rising edge and pulls IRQ low. When latched, write a 1 to clear this bit. | 0x0 | R/W |
| | | | | 1 | Bad parameter. | | |
| | | 6 | LANEFIFO | | Lane FIFO empty/full Status. If IRQEN_LANEFIFO is low, this bit shows current status. If not, this bit latches on a rising edge and pulls IRQ low. When latched, write a 1 to clear this bit. | 0x0 | R/W |
| | | | | 1 | Lane FIFO empty/full. | | |
| | | 5 | DLYBUF | | Delay buffer empty/full Status. If IRQEN_DLYBUF is low, this bit shows current status. If not, this bit latches on a rising edge and pulls IRQ low. When latched, write a 1 to clear this bit. | 0x0 | R/W |

| Addr. | Name | Bits | Bit Name | Settings | Description | Reset | Access |
|---|---|---|---|---|---|---|---|
| | | | | 1 | Delay buffer empty/full. | | |
| | | 4 | DATAREADY | | Data ready status. If IRQEN_DATAREADY is low, this bit shows current status. If not, this bit latches on a rising edge and pulls IRQ low. When latched, write a 1 to clear this bit. | 0x0 | R/W |
| | | | | 1 | Data ready. | | |
| | | 3 | OVERFLOW | | Data path over flow interrupt. | 0x0 | R/W |
| | | 2 | RESERVED | | Reserved. | 0x0 | R |
| | | 1 | PRBSQ | | DACQ PRBS error status. If IRQEN_PRBSQ is low, this bit shows current status. If not, this bit latches on a rising edge and pulls IRQ low. When latched, write a 1 to clear this bit. | 0x0 | R/W |
| | | | | 1 | DACQ failed PRBS. | | |
| | | 0 | PRBSI | | DACI PRBS error status. If IRQEN_PRBSI is low, this bit shows current status. If not, this bit latches on a rising edge and pulls IRQ low. When latched, write a 1 to clear this bit. | 0x0 | R/W |
| | | | | 1 | DACI failed PRBS. | | |
| 0x025 | IRQ_STATUS2 | 7 | PAERR | | PDP error. If IRQEN_ PAERR is low, this bit shows current status. If not, this bit latches on a rising edge and pulls IRQ low. When latched, write a 1 to clear this bit. | 0x0 | R/W |
| | | | | 1 | Data into datapath over power threshold. | | |
| | | [6:5] | RESERVED | | Reserved. | 0x0 | R/W |
| | | 4 | NCOALIGN | | NCO align tripped status. If IRQEN_ NCOALIGN is low, this bit shows current status. If not, this bit latches on a rising edge and pulls IRQ low. When latched, write a 1 to clear this bit. | 0x0 | R/W |
| | | | | 1 | NCO alignment tripped. | | |
| | | 3 | SYNCLOCK | | LMFC alignment locked status. If IRQEN_SYNCLOCK is low, this bit shows current status. If not, this bit latches on a rising edge and pulls IRQ low. When latched, write a 1 to clear this bit. | 0x0 | R/W |
| | | | | 1 | LMFC alignment locked | | |
| | | 2 | SYNCROTATE | | LMFC alignment rotate status. If IRQEN_SYNCROTATE is low, this bit shows current status. If not, this bit latches on a rising edge and pulls IRQ low. When latched, write a 1 to clear this bit. | 0x0 | R/W |
| | | | | 1 | LMFC alignment rotated. | | |
| | | 1 | SYNCWLIM | | Outside window status. If IRQEN_SMODE_ SYNC_WLIM0 is low, this bit shows current status. If not, this bit latches on a rising edge and pulls IRQ low. When latched, write a 1 to clear this bit. | 0x0 | R/W |
| | | | | 1 | LMFC alignment phase outside of limit window. | | |
| | | 0 | SYNCTRIP | | LMFC alignment tripped status. If IRQEN_SYNCTRIP is low, this bit shows current status. If not, this bit latches on a rising edge and pulls IRQ low. When latched, write a 1 to clear this bit. | 0x0 | R/W |
| | | | | 1 | LMFC alignment tripped. | | |
| 0x026 | OVERFLOW_STATUS0 | 7 | PFIR_OVERFLOW | | The overflow status of PFIR filter. | 0x0 | R |
| | | 6 | INT1_OVERFLOW | | The overflow status of INT1 filter. | 0x0 | R |
| | | 5 | INT2_OVERFLOW | | The overflow status of INT2 filter. | 0x0 | R |
| | | 4 | INT3_OVERFLOW | | The overflow status of INT3 filter. | 0x0 | R |
| | | 3 | COARSE_MOD_BY8_OVERFLOW | | The overflow status of $f_S/8$ coarse modulation. | 0x0 | R |
| | | 2 | FINE_MOD_OVERFLOW | | The overflow status of fine modulation. | 0x0 | R |
| | | 1 | PHASE_ADJ_OVERFLOW | | The overflow status of phase adjustment. | 0x0 | R |
| | | 0 | GAIN_ADJ_OVERFLOW | | The overflow status of gain adjustment. | 0x0 | R |
| 0x027 | OVERFLOW_STATUS1 | [7:1] | RESERVED | | Reserved. | 0x0 | R |
| | | 0 | DC_OFFSET_OVERFLOW | | The overflow status of DC offset. | 0x0 | R |

| Addr. | Name | Bits | Bit Name | Settings | Description | Reset | Access |
|---|---|---|---|---|---|---|---|
| 0x030 | JESD_CHECKS | [7:6] | RESERVED | | Reserved. | 0x0 | R |
| | | 5 | ERR_DLYOVER | | Error: LMFC_DELAY > JESD_K parameter. | 0x0 | R |
| | | | | 1 | LMFC_DELAY > JESD_K. | | |
| | | 4 | ERR_WINLIMIT | | Unsupported window limit. | 0x0 | R |
| | | | | 1 | Unsupported SYSREF window limit. | | |
| | | 3 | ERR_JESDBAD | | Unsupported M/L/S/F selection. | 0x0 | R |
| | | | | 1 | This JESD combination is not supported. | | |
| | | 2 | ERR_KUNSUPP | | Unsupported K values. 16 and 32 are supported. | 0x0 | R |
| | | | | 1 | K value unsupported. | | |
| | | 1 | ERR_SUBCLASS | | Unsupported subclass value. 0 and 1 are supported. | 0x0 | R |
| | | | | 1 | Unsupported subclass value. | | |
| | | 0 | ERR_INTSUPP | | Unsupported interpolation rate factor. 1, 2, 4, 8 are supported. | 0x0 | R |
| | | | | 1 | Unsupported interpolation rate factor. | | |
| 0x032 | SYNC_DACDELAY_L | [7:0] | DAC_DELAY_L | | Sync DAC delay. | 0x0 | R/W |
| 0x033 | SYNC_DACDELAY_H | [7:1] | RESERVED | | Reserved. | 0x0 | R |
| | | 0 | DAC_DELAY_H | | DAC delay, Bit 8. | 0x0 | R/W |
| 0x034 | SYNC_ERRWINDOW | [7:3] | RESERVED | | Reserved. | 0x0 | R |
| | | [2:0] | ERRWINDOW | | LMFC sync error window. The error window allows the SYSREF± sample phase to vary within the confines of the window without triggering a clock adjustment. This is useful if SYSREF± cannot be guaranteed to always arrive in the same period of the device clock associated with the target phase. Error window tolerance = ± ERRWINDOW in DACCLKs. | 0x0 | R/W |
| | | | | 000 | Error window tolerance ± 0. | | |
| | | | | 001 | Error window tolerance ± 1. | | |
| | | | | 010 | Error window tolerance ± 2. | | |
| | | | | 011 | Error window tolerance ± 3. | | |
| | | | | 100 | Error window tolerance ± 4. | | |
| | | | | 101 | Error window tolerance ± 5. | | |
| | | | | 110 | Error window tolerance ± 6. | | |
| | | | | 111 | Error window tolerance ± 7. | | |
| 0x035 | SYNC_DLYCOUNT | [7:0] | DLYCOUNT | | Pulse mode delay. Specifies minimum number of LMFC counts before a SYSREF± sync cycle is considered active. | 0x0 | R/W |
| 0x036 | SYNC_REFCOUNT | [7:0] | REFCOUNT | | Pulse mode reference count. Specifies count of SYSREF± pulses to cause a rotate. | 0x0 | R/W |
| 0x038 | SYNC_LASTERR_L | [7:0] | LASTERROR_L | | Sync last error, Bits[7:0]. | 0x0 | R |
| 0x039 | SYNC_LASTERR_H | 7 | LASTUNDER | | LMFC sync last error under flag. | 0x0 | R |
| | | | | 1 | Last phase error was beyond lower window tolerance boundary. | | |
| | | 6 | LASTOVER | | LMFC sync last error over flag. | 0x0 | R |
| | | | | 1 | Last phase error was beyond upper window tolerance boundary. | | |
| | | [5:1] | RESERVED | | Reserved. | 0x0 | R |
| | | 0 | LASTERROR_H | | Sync last error, Bit 8 and flags. | 0x0 | R |
| 0x03A | SYNC_CONTROL | 7 | SYNCENABLE | | Sync logic enable. | 0x0 | R/W |
| | | | | 1 | Enable sync logic. | | |
| | | | | 0 | Disable sync logic. | | |
| | | 6 | SYNCARM | | Sync arm strobe. | 0x0 | R/W |
| | | | | 1 | Sync one shot armed. | | |
| | | 5 | SYNCCLRSTKY | | LMFC sync sticky bit clear. On a rising edge, this bit clears SYNC_ROTATE and SYNC_TRIP. | 0x0 | R/W |
| | | 4 | SYNCCLRLAST | | LMFC sync clear last error. On a rising edge, this bit clears LASTERROR, LASTUNDER, and LASTOVER. | 0x0 | R/W |
| | | [3:0] | SYNCMODE | | LMFC sync mode. | 0x0 | R/W |
| | | | | 0b0001 | Sync one shot mode. | | |

| Addr. | Name | Bits | Bit Name | Settings | Description | Reset | Access |
|---|---|---|---|---|---|---|---|
| | | | | 0b0010 | Sync continuous mode. | | |
| | | | | 0b1000 | Sync monitor only mode. | | |
| | | | | 0b1001 | Sync one shot, then monitor. | | |
| 0x03B | SYNC_STATUS | 7 | SYNCBUSY | | Sync machine busy. | 0x0 | R |
| | | | | 1 | Sync logic SM is busy. | | |
| | | [6:4] | RESERVED | | Reserved. | 0x0 | R |
| | | 3 | SYNCLOCK | | Sync alignment locked. | 0x0 | R |
| | | | | 1 | Sync logic aligned within window. | | |
| | | 2 | SYNCROTATE | | Sync rotated. | 0x0 | R |
| | | | | 1 | Sync logic rotated with SYSREF± (sticky). | | |
| | | 1 | SYNCWLIM | | Sync alignment limit Range. | 0x0 | R |
| | | | | 1 | Phase error outside window threshold. | | |
| | | 0 | SYNCTRIP | | Sync tripped after arming. | 0x0 | R |
| | | | | 1 | Sync received SYSREF± pulse (sticky). | | |
| 0x03C | SYNC_CURRERR_L | [7:0] | CURRERROR_L | | LMFC sync alignment error. 9-bit twos complement value that represents the phase error in number of DAC clock cycles (that is, number of DAC clocks between LMFC edge and SYSREF± edge). When an adjustment of the clocks is made on any given SYSREF±, the value of the phase error is placed into SYNC_ LASTERR, and SYNC_CURRERR is forced to 0. | 0x0 | R |
| 0x03D | SYNC_CURRERR_H | 7 | CURRUNDER | | LMFC sync current error under flag. | 0x0 | R |
| | | | | 1 | Current phase error is beyond lower window tolerance boundary. | | |
| | | 6 | CURROVER | | LMFC sync current error over flag. | 0x0 | R |
| | | | | 1 | Current phase error is beyond upper window tolerance boundary. | | |
| | | [5:1] | RESERVED | | Reserved. | 0x0 | R |
| | | 0 | CURRERROR_H | | SYNC_CURRERR, Bit 8. | 0x0 | R |
| 0x03E | ERROR_THERM | 7 | THRMOLD | | Error is from a prior sample. | 0x0 | R |
| | | | | 1 | From an old sample. | | |
| | | [6:5] | RESERVED | | Reserved. | 0x0 | R |
| | | 4 | THRMOVER | | Error > +WinLimit. | 0x0 | R |
| | | | | 1 | Error > +WinLimit. | | |
| | | 3 | THRMPOS | | Error > 0. | 0x0 | R |
| | | | | 1 | Error > 0. | | |
| | | 2 | THRMZERO | | Error = 0. | 0x0 | R |
| | | | | 1 | Error = 0. | | |
| | | 1 | THRMNEG | | Error < 0. | 0x0 | R |
| | | | | 1 | Error < 0. | | |
| | | 0 | THRMUNDER | | Error < −WinLimit. | 0x0 | R |
| | | | | 1 | Error < −WinLimit. | | |
| 0x040 | DAC_GAIN1_I | [7:2] | RESERVED | | Reserved. | 0x0 | R/W |
| | | [1:0] | DACFSC_I[9:8] | | 2 MSBs of I DAC gain.<br>1111111111 = 20 mA.<br>0000000000 = 4 mA. | 0x3 | R/W |
| 0x041 | DAC_GAIN0_I | [7:0] | DACFSC_I[7:0] | | 8 LSBs of I DAC gain. | 0xFF | R/W |
| 0x042 | DAC_GAIN1_Q | [7:2] | RESERVED | | Reserved. | 0x0 | R/W |
| | | [1:0] | DACFSC_Q[9:8] | | 2 MSBs of Q DAC gain.<br>1111111111 = 20 mA.<br>0000000000 = 4 mA. | 0x3 | R/W |
| 0x043 | DAC_GAIN0_Q | [7:0] | DACFSC_Q[7:0] | | 8 LSBs of Q DAC Gain. | 0xFF | R/W |
| 0x047 | COARSE_GROUP_DLY | [7:4] | COARSE_GROUP_DLY_I | | Coarse Group Delay. 0 = minimum delay, 15 = maximum delay.<br>The range of the delay is −4 DAC clock periods to +3 DAC clock periods and the resolution is 1/2 DAC clock period. | 0x08 | R/W |

| Addr. | Name | Bits | Bit Name | Settings | Description | Reset | Access |
|---|---|---|---|---|---|---|---|
| | | [3:1] | COARSE_GROUP_DLY_Q | | Coarse Group Delay. 0 = minimum delay, 15 = maximum delay.<br>The range of the delay is −4 DAC clock periods to +3 DAC clock periods and the resolution is 1/2 DAC clock period. | 0x08 | |
| 0x050 | NCOALIGN_MODE | 7 | NCO_ALIGN_ARM | | Arm NCO align. On a rising edge, arms the NCO align operation. | 0x0 | R/W |
| | | 6 | RESERVED | | Reserved. | 0x0 | R |
| | | 5 | NCO_ALIGN_MTCH | | NCO align data match. | 0x0 | R |
| | | | | 1 | Key NCO align data match. | | |
| | | | | 0 | If finished, NCO not aligned on data match. | | |
| | | 4 | NCO_ALIGN_PASS | | NCO align pass. | 0x0 | R |
| | | | | 1 | NCO align takes effect. | | |
| | | | | 0 | Clear not taken effect yet. | | |
| | | 3 | NCO_ALIGN_FAIL | | NCO align fail. | 0x0 | R |
| | | | | 1 | NCO reset during rotate. | | |
| | | | | 0 | Not finished yet. | | |
| | | 2 | RESERVED | | Reserved. | 0x0 | R |
| | | [1:0] | NCO_ALIGN_MODE | | NCO align mode. | 0x0 | R/W |
| | | | | 00 | NCO align disabled. | | |
| | | | | 10 | NCO align on data key. | | |
| | | | | 01 | NCO align on SYSREF±. | | |
| 0x051 | NCOKEY_ILSB | [7:0] | NCOKEYI[7:0] | | NCO data key LSB for I. | 0x0 | R/W |
| 0x052 | NCOKEY_IMSB | [7:0] | NCOKEYI[15:8] | | NCO data key MSB for I. | 0x0 | R/W |
| 0x053 | NCOKEY_QLSB | [7:0] | NCOKEYQ[7:0] | | NCO data key LSB for Q. | 0x0 | R/W |
| 0x054 | NCOKEY_QMSB | [7:0] | NCOKEYQ[15:8] | | NCO data key MSB for Q. | 0x0 | R/W |
| 0x060 | PDP_THRES0 | [7:0] | PDP_THRESHOLD[7:0] | | PDP_THRESHOLD is the average power threshold for comparison. If the moving average of signal power crosses this threshold, PDP_PROTECT is set high. | 0x0 | R/W |
| 0x061 | PDP_THRES1 | [7:5] | RESERVED | | Reserved. | 0x0 | R |
| | | [4:0] | PDP_THRESHOLD[12:8] | | See Register 0x060. | 0x0 | R/W |
| 0x062 | PDP_AVG_TIME | 7 | PDP_ENABLE | 1 | Enable average power calculation. | 0x0 | R/W |
| | | [6:4] | RESERVED | | Reserved. | 0x0 | R |
| | | [3:0] | PDP_AVG_TIME | | Can be set from 0 to 10. Averages across $2^{(9 + PDP\_AVG\_TIME)}$ IQ sample pairs. | 0x0 | R/W |
| 0x063 | PDP_POWER0 | [7:0] | PDP_POWER[7:0] | | If PDP_POWER has not gone over PDP_THRESHOLD, PDP_POWER reads back the moving average of the signal power ($I^2 + Q^2$). Only 6 data MSBs are used in calculating power. | 0x0 | R |
| 0x064 | PDP_POWER1 | [7:5] | RESERVED | | Reserved. | 0x0 | R |
| | | [4:0] | PDP_POWER[12:8] | | See Register 0x063. | 0x0 | R |
| 0x065 | PA_OFFGAIN0 | 7 | EN_UKCIRQOFFGAIN | | Enable off gain function when unexpected K-characters error counter reaches the threshold. | 0x0 | R/W |
| | | [6:2] | RESERVED | | Reserved. | 0x0 | R/W |
| | | 1 | EN_DELAYBUFFEROFF-GAIN | | Enable off gain function when delay buffer has error. | 0x0 | R/W |
| | | 0 | EN_LANEFIFOOFFGAIN | | Enable off gain when lane FIFO has error. | 0x0 | R/W |
| 0x066 | PA_OFFGAIN1 | 7 | EN_CMMIRQOFFGAIN | | Enable off gain function when ILAS configuration on Lane 0 is mismatched. | 0x0 | R/W |
| | | 6 | EN_CGSIRQGAIN | | Enable off gain function when CGS failed. | 0x0 | R/W |
| | | 5 | EN_FSIRQOFFGAIN | | Enable off gain function when frame synchronization failed. | 0x0 | R/W |
| | | 4 | EN_GCSIRQOFFGAIN | | Enable off gain function when checksum failed. | 0x0 | R/W |
| | | 3 | EN_ILSIRQOFFGAIN | | Enable off gain function when initial lane alignment failed. | 0x0 | R/W |
| | | 2 | EN_ILDIRQOFFGAIN | | Enable off gain function when lane deskew failed. | 0x0 | R/W |
| | | 1 | EN_DISIRQOFFGAIN | | Enable off gain function when disparity error count exceeds the threshold. | 0x0 | R/W |
| | | 0 | EN_NITIRQOFFGAIN | | Enable off gain because NIT error count exceeded the threshold. | 0x0 | R/W |

| Addr. | Name | Bits | Bit Name | Settings | Description | Reset | Access |
|---|---|---|---|---|---|---|---|
| 0x080 | CLKCFG0 | [7:3] | RESERVED | | Reserved. | 0x0 | R/W |
| | | 2 | DUTY_EN | | Enable duty cycle control of clock receiver | 0x1 | R/W |
| | | [1:0] | RESERVED | | Reserved. | 0x0 | R/W |
| 0x081 | SYSREF_ACTRL0 | [7:5] | RESERVED | | Reserved. | 0x0 | R/W |
| | | 4 | PD_SYSREF | | Power-down SYSREF± buffer. This bit powers down the SYSREF± receiver. For Subclass 1 operation to work, this buffer must be enabled. | 0x1 | R/W |
| | | 3 | HYS_ON | | Hysteresis enabled. This bit enables the programmable hysteresis control for the SYSREF± receiver. Using hysteresis gives some noise resistance, but delays the SYSREF± edge an amount depending on HYS_CNTRL and the SYSREF± edge rate. The SYSREF± KOW is not guaranteed when using hysteresis. | 0x0 | R/W |
| | | 2 | SYSREF_RISE | | Select DAC clock edge to sample SYSREF±. | 0x0 | R/W |
| | | | | 0 | Use falling edge of DAC clock to sample SYSREF± for alignment | | |
| | | | | 1 | Use rising edge of DAC clock to sample SYSREF± for alignment | | |
| | | [1:0] | HYS_CNTRL1 | | Hysteresis control, Bits[9:8]. HYS_CNTRL is a 10-bit thermometer-coded number. Each bit set adds 10 mV of differential hysteresis to the SYSREF± receiver. | 0x0 | R/W |
| 0x082 | SYSREF_ACTRL1 | [7:0] | HYS_CNTRL0 | | Hysteresis control, Bits[7:0]. | 0x0 | R/W |
| 0x083 | DACPLLCNTRL | 7 | RECAL_DACPLL | | Recalibrate DAC PLL. On a rising edge of this bit, recalibrate the DAC PLL. | 0x0 | R/W |
| | | [6:5] | RESERVED | | Reserved. | 0x0 | R/W |
| | | 4 | ENABLE_DACPLL | | Synthesizer enable. This bit enables and calibrates the DAC PLL. | 0x0 | R/W |
| | | [3:0] | RESERVED | | Reserved. | 0x0 | R/W |
| 0x084 | DACPLLSTATUS | 7 | CP_OVERRANGE_H | | Charge pump high overrange. This bit indicates that the DAC PLL hit the upper edge of its operating band. Recalibrate. | 0x0 | R |
| | | | | 1 | Control voltage too high. | | |
| | | 6 | CP_OVERRANGE_L | | This bit indicates that the DAC PLL hit the lower edge of its operating band. Recalibrate. | 0x0 | R |
| | | | | 1 | Control voltage too low. | | |
| | | 5 | CP_CAL_VALID | | Charge pump calibration valid | 0x0 | R |
| | | | | 0 | If CP_CAL_EN is low, this stays low. | | |
| | | | | 1 | If CP_CAL_EN high (def), this happens when charge pump is calibrated. | | |
| | | 4 | VCO_CAL_PROGRESS | | VCO calibration in progress. | 0x0 | R |
| | | | | 0 | VCO not calibrating. | | |
| | | | | 1 | VCO calibrating. | | |
| | | 3 | CURRENTS_READY | | Indicating DAC PLL bias current status. | 0x0 | R |
| | | | | 0 | Bias not ready. | | |
| | | | | 1 | Bias ready. | | |
| | | 2 | RESERVED | | Reserved. | 0x0 | R/W |
| | | 1 | DAC_PLL_LOCK | | DAC PLL lock bit. This bit is set high by the PLL when it has achieved lock. | 0x0 | R |
| | | 0 | RESERVED | | Reserved. | 0x0 | R/W |
| 0x085 | DACINTEGERWORD0 | [7:0] | B_COUNT | | Integer division word. This bit controls the integer feedback divider for the DAC PLL. Determine the frequency of the DAC clock by the following equations (see the Clock Multiplication Relationships section for more details): $f_{DAC} = f_{REF}/(REF\_DIVRATE) \times 2 \times B\_COUNT$ $f_{VCO} = f_{REF}/(REF\_DIVRATE) \times 2 \times B\_COUNT \times LO\_DIV\_MODE$ Minimum value is 6. | 0x6 | R/W |
| 0x087 | DACLOOPFILT1 | [7:4] | LF_C2_WORD | | C2 control word. | 0x8 | R/W |
| | | [3:0] | LF_C1_WORD | | C1 control word. | 0x8 | R/W |

| Addr. | Name | Bits | Bit Name | Settings | Description | Reset | Access |
|---|---|---|---|---|---|---|---|
| 0x088 | DACLOOPFILT2 | [7:4] | LF_R1_WORD | | R1 control word. | 0x8 | R/W |
| | | [3:0] | LF_C3_WORD | | C3 control word. | 0x8 | R/W |
| 0x089 | DACLOOPFILT3 | 7 | LF_BYPASS_R3 | | Bypass R3 resistor. When this bit is set, bypass the R3 capacitor (set to 0 pF) when R3_WORD is set to 0. | 0x0 | R/W |
| | | 6 | LF_BYPASS_R1 | | Bypass R1 resistor. When this bit is set, bypass the R1 capacitor (set to 0 pF) when R1_WORD is set to 0. | 0x0 | R/W |
| | | 5 | LF_BYPASS_C2 | | Bypass C2 capacitor. When this bit is set, bypass the C2 capacitor (set to 0 pF) when C2_WORD is set to 0. | 0x0 | R/W |
| | | 4 | LF_BYPASS_C1 | | Bypass C1 capacitor. When this bit is set, bypass the C1 capacitor (set to 0 pF) when C1_WORD is set to 0. | 0x0 | R/W |
| | | [3:0] | LF_R3_WORD | | R3 control word. | 0x8 | R/W |
| 0x08A | DACCPCNTRL | [7:6] | RESERVED | | Reserved. | 0x0 | R/W |
| | | [5:0] | CP_CURRENT | | Charge pump current control. | 0x20 | R/W |
| 0x08B | DACLOGENCNTRL | [7:2] | RESERVED | | Reserved. | 0x0 | R/W |
| | | [1:0] | LO_DIV_MODE | | This range controls the RF clock divider between the VCO and DAC clock rates. The options are 4×, 8×, or 16× division. Choose the LO_DIV_MODE so that 6 GHz < $f_{VCO}$ < 12 GHz (see the Clock Multiplication Relationships section for more details): | 0x0 | R/W |
| | | | | 01 | DAC clock = VCO/4 | | |
| | | | | 10 | DAC clock = VCO/8 | | |
| | | | | 11 | DAC clock = VCO/16 | | |
| 0x08C | DACLDOCNTRL1 | [7:3] | RESERVED | | Reserved. | 0x0 | R/W |
| | | [2:0] | REF_DIV_MODE | | Reference clock division ratio. This field controls the amount of division that is done to the input clock at the REFCLK+/ REFCLK− pins before it is presented to the PLL as a reference clock. The reference clock frequency must be between 35 MHz and 80 MHz, but the REFCLK+/REFCLK− input frequency can range from 35 MHz to 1 GHz. The user sets this division to achieve a 35 MHz to 80 MHz PLL reference frequency. For more details see the Clock Multiplication Relationships section. | 0x0 | R/W |
| | | | | 001 | 2×. | | |
| | | | | 010 | 4×. | | |
| | | | | 011 | 8×. | | |
| | | | | 100 | 16×. | | |
| 0x08E | CLK_DETECT | [7:5] | RESERVED | | Reserved. | 0x0 | R/W |
| | | 4 | PD_DAC_ONDIFF | | Automatically power-down DACs if clock is not differential. | 0x0 | R/W |
| | | | | 0 | Only report clock errors. | | |
| | | | | 1 | Automatically force DAC PD is clock if not differential. | | |
| | | 3 | PD_DAC_ONDET | | Automatically power-down DACs if clock is lost. | 0x0 | R/W |
| | | | | 0 | Only report clock errors. | | |
| | | | | 1 | Automatically force DAC PD if clock is lost. | | |
| | | 2 | CLK_ON | | Indicate if DACCLK is on. | 0x0 | R |
| | | | | 0 | The clock is not on. | | |
| | | | | 1 | The clock is on. | | |
| | | 1 | IS_DIFF | | Indicate if the clock is differential. | 0x0 | R |
| | | | | 0 | The clock is not differential. | | |
| | | | | 1 | The clock is differential. | | |
| | | 0 | CLK_DET_EN | | Enable Clock Detector. | 0x0 | R/W |
| | | | | 0 | Disable clock detect circuit. | | |
| | | | | 1 | Enable clock detect circuit. | | |
| 0x0F7 | DIG_TEST0 | [7:2] | RESERVED | | Reserved. | 0x7 | R/W |
| | | 1 | DC_TEST_MOD | | DC test mode enable. | 0x0 | R/W |
| | | 0 | DIG_CLK_PD | | Power down top digital clock. | 0x0 | R/W |
| 0x0F8 | DC_TEST_VALUEI0 | [7:0] | DC_TEST_VALUEI[7:0] | | DC value LSB of dc test mode for I DAC. | 0x0 | R/W |

| Addr. | Name | Bits | Bit Name | Settings | Description | Reset | Access |
|---|---|---|---|---|---|---|---|
| 0x0F9 | DC_TEST_VALUEI1 | [7:0] | DC_TEST_VALUEI [15:8] | | DC value MSB of dc test mode for I DAC. | 0x0 | R/W |
| 0x0FA | DC_TEST_VALUEQ0 | [7:0] | DC_TEST_VALUEQ[7:0] | | DC value LSB of dc test mode for Q DAC. | 0x0 | R/W |
| 0x0FB | DC_TEST_VALUEQ1 | [7:0] | DC_TEST_VALUEQ[15:8] | | DC value MSB of dc test mode for Q DAC. | 0x0 | R/W |
| 0x110 | DATA_FORMAT | 7 | BINARY_FORMAT | | Binary or twos complementary format on the data bus. | 0x0 | R/W |
| | | | | 0 | Input data is twos complement. | | |
| | | | | 1 | Input data is offset binary. | | |
| | | [6:0] | RESERVED | | Reserved. | 0x0 | R/W |
| 0x111 | DATAPATH_CTRL | 7 | INVSINC_ENABLE | 1 | Enable inverse sinc filter. | 0x0 | R/W |
| | | 6 | RESERVED | | Reserved. | 0x0 | R/W |
| | | 5 | DIG_GAIN_ENABLE | 1 | Enable digital gain function. | 0x1 | R/W |
| | | 4 | PHASE_ADJ_ENABLE | 1 | Enable phase adjust compensation. | 0x0 | R/W |
| | | [3:2] | MODULATION_TYPE | | Selects type Of modulation operation. | 0x0 | R/W |
| | | | | 00 | No modulation. | | |
| | | | | 01 | NCO fine modulation (uses FTW). | | |
| | | | | 10 | fs/4 coarse modulation. | | |
| | | | | 11 | fs/8 coarse modulation. | | |
| | | 1 | SEL_SIDEBAND | | Spectrum inversion control. Can be used with both fine modulation and coarse modulation. This causes the negative sideband to be selected and is equivalent to changing the sign of FTW. | 0x0 | R/W |
| | | 0 | RESERVED | | Reserved. | 0x0 | R/W |
| 0x112 | INTERP_MODE | 7 | SINGLE_DAC_EN | | Enable single DAC mode. If this bit is high, only Mode 9 and Mode 10 in Table 43 can be supported and Q DAC powers down automatically. | 0x0 | R/W |
| | | [6:2] | RESERVED | | Reserved. | 0x0 | R/W |
| | | [1:0] | INTERP_MODE | | Interpolation mode. | 0x1 | R/W |
| | | | | 00 | 1× (bypass). | | |
| | | | | 01 | 2× mode. | | |
| | | | | 10 | 4× mode. | | |
| | | | | 11 | 8× mode. | | |
| 0x113 | NCO_FTW_UPDATE | [7:2] | RESERVED | | Reserved. | 0x0 | R/W |
| | | 1 | FTW_UPDATE_ACK | | Frequency tuning word update acknowledge. This readback is high when an FTW has been updated. | 0x0 | R |
| | | 0 | FTW_UPDATE_REQ | | Frequency tuning word update request from SPI. Unlike most registers, those relating to fine NCO modulation (Register 0x114 to Register 0x11B) are not updated immediately upon writing to them. Once the desired FTW and phase offset values are written, set this bit. These registers update on the rising edge of this bit. It is only after this update that the internal state matches Register 0x114 to Register 0x11B. Confirmation that this update has occurred can be made by reading back Bit 1 of this register and ensuring it is set high for the update acknowledge. | 0x0 | R/W |
| 0x114 | FTW0 | [7:0] | FTW[7:0] | | NCO frequency tuning word. | 0x0 | R/W |
| 0x115 | FTW1 | [7:0] | FTW[15:8] | | NCO frequency tuning word. | 0x0 | R/W |
| 0x116 | FTW2 | [7:0] | FTW[23:16] | | NCO frequency tuning word. | 0x0 | R/W |
| 0x117 | FTW3 | [7:0] | FTW[31:24] | | NCO frequency tuning word. | 0x0 | R/W |
| 0x118 | FTW4 | [7:0] | FTW[39:32] | | NCO frequency tuning word. | 0x0 | R/W |
| 0x119 | FTW5 | [7:0] | FTW[47:40] | | NCO frequency tuning word. | 0x10 | R/W |
| 0x11A | NCO_PHASE_OFFSET0 | [7:0] | NCO_PHASE_OFFSET[7:0] | | 8 LSBs of NCO Phase Offset. NCO_PHASE_OFFSET changes the phase of both I and Q data, and is only functional when using NCO fine modulation. It is a 16-bit, twos complement number ranging from −180° to+180° in steps of 0.0055°. | 0x0 | R/W |
| 0x11B | NCO_PHASE_OFFSET1 | [7:0] | NCO_PHASE_ OFFSET[15:8] | | 8 MSBs of NCO phase offset. | 0x0 | R/W |
| 0x11C | IQ_PHASE_ADJ0 | [7:0] | PHASE_ADJ[7:0] | | 8 LSBs of phase compensation word. Phase compensation changes the phase between the I and Q data. PHASE_ADJ is a 13-bit, twos | 0x0 | R/W |

| Addr. | Name | Bits | Bit Name | Settings | Description | Reset | Access |
|---|---|---|---|---|---|---|---|
| | | | | | complement value. The control ranges from −14° to +14° with 0.0035° resolution steps. | | |
| 0x11D | IQ_PHASE_ADJ1 | [7:5] | RESERVED | | Reserved. | 0x0 | R/W |
| | | [4:0] | PHASE_ADJ[12:8] | | 5 MSBs of phase compensation word. | | |
| 0x11F | TXEN_SM_0 | [7:6] | FALL_COUNTERS | | Fall Counters. The number of counters to use to delay TX_PROTECT fall from TXEN falling edge. Must be set to 1 or 2. | 0x2 | R/W |
| | | [5:4] | RISE_COUNTERS | | Rise Counters. The number of counters to use to delay TX_PROTECT rise from TXEN rising edge. | 0x0 | R/W |
| | | 3 | RESERVED | | Reserved. | 0x0 | R/W |
| | | 2 | PROTECT_OUT_INVERT | | PROTECT_OUT invert. | 0x0 | R/W |
| | | | | 0 | PROTECT_OUT is low when error happens. Suitable for enabling downstream components during transmission. | | |
| | | | | 1 | PROTECT_OUT is high when error happens. Suitable for disabling downstream components when not transmitting. | | |
| | | 1 | RESERVED | | Reserved. | 0x1 | R/W |
| | | 0 | TXEN_SM_EN | | Enable TXEN state machine. | 0x1 | R/W |
| 0x125 | DACOUT_ON_DOWN | [7:2] | RESERVED | | Reserved. | 0x0 | R/W |
| | | 1 | DACOUT_SHUTDOWN | | Shut down DAC output. 1 means DAC is shut down manually. 1 = shut down, 0 = enable DAC. | 0x0 | R/W |
| | | 0 | DACOUT_ON_TRIGGER | | Turn on DAC output manually. Self clear signal. Cannot turn on the DAC if DAC is shut down by Bit 1, DACOUT_SHUTDOWN. | 0x0 | R/W |
| 0x12C | DACOFF | 7 | PROTECT_MODE | | If this bit is high then DAC is in protect mode, and DAC is shut down automatically when some errors happen. | 0x1 | R/W |
| | | [6:1] | RESERVED | | Reserved. | 0x0 | R/W |
| | | 0 | DACOFF_AVG_PW | | If this bit is high and Bit 7 is high, then if input average power is bigger than given threshold (see Register 0x60, Register 0x61) within a given time window, DAC output shuts down automatically. | 0x1 | R/W |
| 0x12F | DIE_TEMP_CTRL0 | 7 | RESERVED | | Reserved. | 0x0 | R/W |
| | | [6:4] | FS_CURRENT | | Aux ADC full-scale current (LSB 12.5 µA). Must write the default value for proper operation. | 0x2 | R/W |
| | | | | 000 | Lowest current (50 µA). | | |
| | | | | 111 | Highest current (137.5 µA). | | |
| | | [3:1] | RESERVED | | Reserved. | 0x0 | R/W |
| | | 0 | TEMP_SENSOR_ENABLE | | 1 = Enable temperature sensor. | 0x0 | R/W |
| 0x132 | DIE_TEMP0 | [7:0] | DIE_TEMP[7:0] | | Die temperature code readback. | 0x0 | R |
| 0x133 | DIE_TEMP1 | [7:0] | DIE_TEMP[15:8] | | Die temperature code readback. | 0x0 | R |
| 0x134 | DIE_TEMP_UPDATE | [7:1] | RESERVED | | Reserved. | 0x0 | R/W |
| | | 0 | DIE_TEMP_UPDATE | | Die temperature code update. On a rising edge of this bit, a new temperature code is generated. | 0x0 | R/W |
| 0x135 | DC_OFFSET_CTRL | [7:1] | RESERVED | | Reserved. | 0x0 | R/W |
| | | 0 | DC_OFFSET_ON | 1 | Enables dc offset module. | 0x0 | R/W |
| 0x136 | IPATH_DC_OFFSET_1PART0 | [7:0] | LSB_OFFSET_I[7:0] | | 8 LSBs of I path DC offset. LSB_OFFSET_I is a 16-bit, twos complement number that is added to incoming data. | 0x0 | R/W |
| 0x137 | IPATH_DC_OFFSET_1PART1 | [7:0] | LSB_OFFSET_I[15:8] | | 8 MSBs of I path DC offset. Offset. LSB_OFFSET_I is a 16-bit, twos complement number that is added to incoming I data. | 0x0 | R/W |
| 0x138 | QPATH_DC_OFFSET_1PART0 | [7:0] | LSB_OFFSET_Q[7:0] | | 8 LSBs of Q path DC offset. LSB_OFFSET_Q is a 16-bit, twos complement number that is added to incoming Q data. | 0x0 | R/W |
| 0x139 | QPATH_DC_OFFSET_1PART1 | [7:0] | LSB_OFFSET_Q[15:8] | | 8 MSBs of Q path DC offset. LSB_OFFSET_Q is a 16-bit, twos complement number that is added to incoming Q data. | 0x0 | R/W |
| 0x13A | IPATH_DC_OFFSET_2PART | [7:5] | RESERVED | | Reserved. | 0x0 | R/W |
| | | [4:0] | SIXTEENTH_OFFSET_I | | SIXTEENTH_OFFSET_I is a 5-bit twos complement number in 16ths of an LSB that is added to | 0x0 | R/W |

| Addr. | Name | Bits | Bit Name | Settings | Description | Reset | Access |
|---|---|---|---|---|---|---|---|
| | | | | | incoming I data. | | |
| | | | | x | x/16 LSB DC offset. | | |
| 0x13B | QPATH_DC_OFFSET_2PART | [7:5] | RESERVED | | Reserved. | 0x0 | R/W |
| | | [4:0] | SIXTEENTH_OFFSET_Q | | SIXTEENTH_OFFSET_Q is a 5-bit twos complement number in 16ths of an LSB that is added to incoming Q data. | 0x0 | R/W |
| | | | | x | x/16 LSB DC offset. | | |
| 0x13C | IDAC_DIG_GAIN0 | [7:0] | IDAC_DIG_GAIN[7:0] | | LSB of I DAC digital gain. | 0x0 | R/W |
| 0x13D | IDAC_DIG_GAIN1 | [7:4] | RESERVED | | Reserved. | 0x0 | R/W |
| | | [3:0] | IDAC_DIG_GAIN[11:8] | | 4 MSBs of I DAC digital gain. | 0x8 | R/W |
| 0x13E | QDAC_DIG_GAIN0 | [7:0] | QDAC_DIG_GAIN[7:0] | | 8 LSBs of Q DAC digital gain. QDAC_DIG_GAIN is the digital gain of the Q DAC. The digital gain is a multiplier from 0 to 4095/2048 in steps of 1/2048. | 0x0 | R/W |
| 0x13F | QDAC_DIG_GAIN1 | [7:4] | RESERVED | | Reserved. | 0x0 | R/W |
| | | [3:0] | QDAC_DIG_GAIN[11:8] | | 4 MSBs of Q DAC digital gain. | 0x8 | R/W |
| 0x140 | GAIN_RAMP_UP_STEP0 | [7:0] | GAIN_RAMP_UP_STEP[7:0] | | 8 LSBs of gain ramp up step. GAIN_RAMP_UP_STEP controls the amplitude step size of the BSMs ramping feature when the gain is being ramped to its assigned value. | 0x4 | R/W |
| | | | | 0x0 | Smallest ramp up step size. | | |
| | | | | 0xFFF | Largest ramp up step size. | | |
| 0x141 | GAIN_RAMP_UP_STEP1 | [7:4] | RESERVED | | Reserved. | 0x0 | R/W |
| | | [3:0] | GAIN_RAMP_UP_STEP[11:8] | | 4 MSBs of gain ramp up step. See Register 0x140 for description. | 0x0 | R/W |
| 0x142 | GAIN_RAMP_DOWN_STEP0 | [7:0] | GAIN_RAMP_DOWN_STEP[7:0] | | 8 LSBs of gain ramp down step. GAIN_RAMP_DOWN_STEP controls the amplitude step size of the BSMs ramping feature when the gain is being ramped to zero. | 0x9 | R/W |
| | | | | 0 | Smallest ramp down step size. | | |
| | | | | 0xFFF | Largest ramp down step size. | | |
| 0x143 | GAIN_RAMP_DOWN_STEP1 | [7:4] | RESERVED | | Reserved. | 0x0 | R/W |
| | | [3:0] | GAIN_RAMP_DOWN_STEP[7:0] | | MSB of digital gain drops | 0x0 | R/W |
| 0x14B | PRBS | 7 | PRBS_GOOD_Q | | Good data indicator imaginary channel. | 0x0 | R |
| | | | | 0 | Incorrect sequence detected. | | |
| | | | | 1 | Correct PRBS sequence detected. | | |
| | | 6 | PRBS_GOOD_I | | Good data indicator real channel. | 0x0 | R |
| | | | | 0 | Incorrect sequence detected. | | |
| | | | | 1 | Correct PRBS sequence detected. | | |
| | | 5 | RESERVED | | Reserved. | 0x0 | R/W |
| | | 4 | PRBS_INV_Q | | Data inversion imaginary channel. | 0x1 | R/W |
| | | | | 0 | Expect normal data. | | |
| | | | | 1 | Expect inverted data. | | |
| | | 3 | PRBS_INV_I | | Data inversion real channel. | 0x0 | R/W |
| | | | | 0 | Expect normal data. | | |
| | | | | 1 | Expect inverted data. | | |
| | | 2 | PRBS_MODE | | Polynomial select. | 0x0 | R/W |
| | | | | 0 | 7-bit: $x^7 + x^6 + 1$. | | |
| | | | | 1 | 15-bit: $x^{15} + x^{14} + 1$. | | |
| | | 1 | PRBS_RESET | | Reset error counters. | 0x0 | R/W |
| | | | | 0 | Normal operation. | | |
| | | | | 1 | Reset counters. | | |
| | | 0 | PRBS_EN | | Enable PRBS checker. | 0x0 | R/W |
| | | | | 0 | Disable | | |
| | | | | 1 | Enable | | |
| 0x14C | PRBS_ERROR_I | [7:0] | PRBS_COUNT_I | | Error count value real channel. | 0x0 | R |
| 0x14D | PRBS_ERROR_Q | [7:0] | PRBS_COUNT_Q | | Error count value imaginary channel. | 0x0 | R |

| Addr. | Name | Bits | Bit Name | Settings | Description | Reset | Access |
|---|---|---|---|---|---|---|---|
| 0x151 | DATAPATH_CTRL2 | [7:6] | RESERVED | | Reserved. | 0x0 | R/W |
| | | 5 | PFIR_DEMOD4_ENABLE | | Programmable FIR demodulation enable. | 0x0 | R/W |
| | | 4 | PFIR_ENABLE | | Programmable FIR enable. | 0x0 | R/W |
| | | [3:2] | RESERVED | | Reserved. | 0x0 | R/W |
| | | 1 | NEG_DDS_FREQ | | Negative DDS frequency. | 0x0 | R/W |
| | | 0 | MODULUS_ENABLE | | Modulus enable. | 0x0 | R/W |
| 0x152 | ACC_MODULUS0 | [7:0] | ACC_MODULUS[7:0] | | Tuning Word B for modulus DDS. | 0x0 | R/W |
| 0x153 | ACC_MODULUS1 | [7:0] | ACC_MODULUS[15:8] | | Tuning Word B for modulus DDS. | 0x0 | R/W |
| 0x154 | ACC_MODULUS2 | [7:0] | ACC_MODULUS[23:16] | | Tuning Word B for modulus DDS. | 0x0 | R/W |
| 0x155 | ACC_MODULUS3 | [7:0] | ACC_MODULUS[31:24] | | Tuning Word B for modulus DDS. | 0x0 | R/W |
| 0x156 | ACC_MODULUS4 | [7:0] | ACC_MODULUS[39:32] | | Tuning Word B for modulus DDS. | 0x0 | R/W |
| 0x157 | ACC_MODULUS5 | [7:0] | ACC_MODULUS[47:40] | | Tuning Word B for modulus DDS. | 0x0 | R/W |
| 0x158 | ACC_DELTA0 | [7:0] | ACC_DELTA[7:0] | | Tuning Word A for modulus DDS. | 0x0 | R/W |
| 0x159 | ACC_DELTA1 | [7:0] | ACC_DELTA[15:8] | | Tuning Word A for modulus DDS. | 0x0 | R/W |
| 0x15A | ACC_DELTA2 | [7:0] | ACC_DELTA[23:16] | | Tuning Word A for modulus DDS. | 0x0 | R/W |
| 0x15B | ACC_DELTA3 | [7:0] | ACC_DELTA[31:24] | | Tuning Word A for modulus DDS. | 0x0 | R/W |
| 0x15C | ACC_DELTA4 | [7:0] | ACC_DELTA[39:32] | | Tuning Word A for modulus DDS. | 0x0 | R/W |
| 0x15D | ACC_DELTA5 | [7:0] | ACC_DELTA[47:40] | | Tuning Word A for modulus DDS. | 0x0 | R/W |
| 0x17A | PFIR_COEFF0_L | [7:0] | PFIR_COEFF0[7:0] | | PFIR Coefficient 0, Bits[7:0]. | 0x0 | R/W |
| 0x17B | PFIR_COEFF0_H | [7:1] | RESERVED | | Reserved. | 0x0 | R/W |
| | | 0 | PFIR_COEFF0[8] | | PFIR Coefficient 0, Bit 8. | 0x0 | R/W |
| 0x17C | PFIR_COEFF1_L | [7:0] | PFIR_COEFF1[7:0] | | PFIR Coefficient 1, Bits[7:0]. | 0x0 | R/W |
| 0x17D | PFIR_COEFF1_H | [7:1] | RESERVED | | Reserved. | 0x0 | R/W |
| | | 0 | PFIR_COEFF1[8] | | PFIR Coefficient 1, Bit 8. | 0x0 | R/W |
| 0x17E | PFIR_COEFF2_L | [7:0] | PFIR_COEFF2[7:0] | | PFIR Coefficient 0, Bits[7:0]. | 0x0 | R/W |
| 0x17F | PFIR_COEFF2_H | [7:1] | RESERVED | | Reserved. | 0x0 | R/W |
| | | 0 | PFIR_COEFF2[8] | | PFIR Coefficient 1, Bit 8. | 0x0 | R/W |
| 0x180 | PFIR_COEFF3_L | [7:0] | PFIR_COEFF3[7:0] | | PFIR Coefficient 0, Bits[7:0]. | 0x0 | R/W |
| 0x181 | PFIR_COEFF3_H | [7:1] | RESERVED | | Reserved. | 0x0 | R/W |
| | | 0 | PFIR_COEFF3[8] | | PFIR Coefficient 1, Bit 8. | 0x0 | R/W |
| 0x182 | PFIR_COEFF_UPDATE | [7:1] | RESERVED | | Reserved. | 0x0 | R/W |
| | | 0 | PFIR_COEFF_UPDATE | | PFIR coefficient update. | 0x0 | R/W |
| 0x1B4 | DACPLLT4 | 7 | BYP_LOAD_DELAY | | Bypass load delay. | 0x0 | R/W |
| | | [6:3] | VCO_CAL_OFFSET | | Starting offset for VCO calibration | 0xF | R/W |
| | | 2 | RESERVED | | Reserved. | 0x0 | R/W |
| | | 1 | EXT_BAND_EN | | Force VCO tuning band externally | 0x0 | R/W |
| | | | | 0 | Normal auto calibration mode. | | |
| | | | | 1 | Manual for VCO band. | | |
| | | 0 | EXT_BAND2 | | External band MSB. | 0x0 | W |
| 0x1B5 | DACPLLT5 | [7:4] | INIT_ALC_VALUE | | Initial ALC sweep value. | 0x8 | R/W |
| | | [3:0] | VCO_VAR | | Varactor KVO setting. | 0x3 | R/W |
| 0x1B6 | DACPLLT6 | 7 | RESERVED | | Reserved. | 0x0 | R/W |
| | | 6 | PORESETB_VCO | | $\overline{\text{RESET}}$ for VCO logic. | 0x1 | R/W |
| | | [5:4] | EXT_VCO_BITSEL | | Bit select; Does nothing. | 0x0 | R/W |
| | | [3:0] | VCO_LVL_OUT | | VCO amplitude control. | 0xA | R/W |
| 0x1BB | DACPLLTB | [7:5] | RESERVED | | Reserved. | 0x0 | R/W |
| | | [4:3] | VCO_BIAS_TCF | | Temperature coefficient for VCO bias. | 0x1 | R/W |
| | | [2:0] | VCO_BIAS_REF | | VCO Bias control | 0x4 | R/W |
| 0x1C5 | DACPLLT18 | [7:4] | RESERVED | | Reserved. | 0x0 | R/W |
| | | [3:0] | VCO_VAR_REF | | VCO varactor reference | 0x8 | R/W |
| 0x1FE | TEST_MODE | [7:3] | RESERVED | | Reserved. | 0x0 | R |
| | | [2:0] | TSTWINDOW | | Sync error window. Sync alignment tolerance in ±DACCLKs. | 0x0 | R/W |
| 0x200 | MASTER_PD | [7:1] | RESERVED | | Reserved. | 0x0 | R/W |
| | | 0 | SPI_PD_MASTER | | Power down the entire JESD204B receiver analog | 0x1 | R/W |

| Addr. | Name | Bits | Bit Name | Settings | Description | Reset | Access |
|---|---|---|---|---|---|---|---|
| | | | | | (all four channels plus bias). | | |
| 0x201 | PHY_PD | [7:4] | RESERVED | | Reserved. | 0x0 | R/W |
| | | [3:0] | SPI_PD_PHY | | SPI override to power down the individual PHYs. Set Bit x to power down the corresponding SERDINx± PHY | 0x0 | R/W |
| 0x203 | GENERIC_PD | [7:2] | RESERVED | | Reserved. | 0x0 | R/W |
| | | 1 | SPI_SYNC_PD | | Power down LVDS buffer for $\overline{\text{SYNCOUT}}$±. | 0x0 | R/W |
| | | 0 | RESERVED | | Reserved. | 0x0 | R/W |
| 0x206 | CDR_RESET | [7:1] | RESERVED | | Reserved. | 0x0 | R/W |
| | | 0 | SPI_CDR_RESETN | | Resets the digital control logic for all PHYs. | 0x1 | R/W |
| | | | | 0 | Hold CDR in reset | | |
| | | | | 1 | Enable CDR | | |
| 0x230 | CDR_OPERATING_ MODE_REG_0 | [7:6] | RESERVED | | Reserved. | 0x0 | R/W |
| | | 5 | ENHALFRATE | | Enables half-rate CDR operation. Set to 1 when 5.75 Gbps ≤ lane rate ≤ 12.38 Gbps. | 0x1 | R/W |
| | | [4:2] | RESERVED | | Must write the default value for proper operation. | 0x2 | R/W |
| | | 1 | CDR_OVERSAMP | | Enables oversampling of the input data. Set to 1 when 1.44 Gbps ≤ lane rate ≤ 3.09 Gbps. | 0x0 | R/W |
| | | 0 | SYNCOUTB_SWING | | This bit is to adjust $\overline{\text{SYNCOUT}}$± LVDS output swing. | 0x0 | R/W |
| | | | | 0 | $\overline{\text{SYNCOUT}}$± swing VOD is about 170 mV. | | |
| | | | | 1 | $\overline{\text{SYNCOUT}}$± swing VOD is about 350 mV. | | |
| 0x268 | EQ_BIAS_REG | [7:6] | EQ_POWER_MODE | | Control the equalizer power/insertion loss capability. | 0x1 | R/W |
| | | | | 00 | Normal mode. | | |
| | | | | 01 | Low power mode. | | |
| | | [5:0] | RESERVED | | Reserved. Must write the default value for proper operation. | 0x22 | R/W |
| 0x280 | SERDESPLL_ENABLE_ CNTRL | [7:3] | RESERVED | | Reserved. | 0x0 | R/W |
| | | 2 | RECAL_SERDESPLL | | Recalibrate SERDES PLL. On a rising edge, recalibrate the SERDES PLL. | 0x0 | R/W |
| | | 1 | RESERVED | | Reserved. | 0x0 | R/W |
| | | 0 | ENABLE_SERDESPLL | | Enable the SERDES PLL. Setting this bit enables and calibrates the SERDES PLL. | 0x0 | R/W |
| 0x281 | SERDES_PLL_STATUS | [7:6] | RESERVED | | Reserved. | 0x0 | R |
| | | 5 | SERDES_CP_OVER_ RANGE_H | | Charge pump high overrange. This bit indicates that the SERDES PLL hit the lower edge of its operating band. Recalibrate. | 0x0 | R |
| | | 4 | SERDES_CP_OVER_ RANGE_L | | Charge pump low overrange. This bit indicates that the SERDES PLL hit the lower edge of its operating band. Recalibrate. | 0x0 | R |
| | | 3 | SERDES_PLL_CAL_VALID | | SERDES PLL calibration valid. This bit indicates that the SERDES PLL has been successfully calibrated. | 0x0 | R |
| | | 2 | SERDES_VCO_CAL_ PROGRESS | | This bit set indicates that a VCO calibration is running. | 0x0 | R |
| | | | | 0 | VCO calibration is not running. | | |
| | | | | 1 | VCO calibration is running. | | |
| | | 1 | SERDES_PLL_CURRENTS_READY | | | 0x0 | R |
| | | | | 0 | PLL bias currents are not ready | | |
| | | | | 1 | PLL bias currents are ready | | |
| | | 0 | SERDES_PLL_LOCK | | SERDES PLL lock. This bit is set high by the PLL when it has achieved lock. | 0x0 | R |
| 0x289 | REF_CLK_DIVIDER_LDO | [7:2] | RESERVED | | Must be set to 1 for proper SERDES PLL configuration. | 0x1 | R/W |
| | | [1:0] | SERDES_PLL_DIV_MODE | | SERDES PLL reference clock division factor. This field controls the division of the SERDES PLL reference clock before it is fed into the SERDES PLL phase frequency detector (PFD). It must be set so $f_{REF}$/DivFactor is between 35 MHz and 80 MHz. | 0x0 | R/W |
| | | | | 00 | Divide by 4 for 5.75 Gbps to 12.38 Gbps lane rate | | |

| Addr. | Name | Bits | Bit Name | Settings | Description | Reset | Access |
|---|---|---|---|---|---|---|---|
| | | | | 01 | Divide by 2 for 2.88 Gbps to 6.19 Gbps lane rate | | |
| | | | | 10 | Divide by 1 for 1.44 Gbps to 3.09 Gbps lane rate | | |
| 0x2A7 | TERM_BLK1_CTRLREG0 | [7:1] | RESERVED | | Reserved. | 0x0 | R |
| | | 0 | SPI_I_TUNE_R_CAL_ TERMBLK1 | | Termination calibration. The rising edge of this bit calibrates PHY terminations to 50 Ω. | 0x0 | R/W |
| 0x300 | GENERAL_JRX_CTRL_0 | 7 | RESERVED | | Reserved. | 0x0 | R |
| | | 6 | CHECKSUM_MODE | | Checksum mode. This bit controls the locally generated JESD204B link parameter checksum method. The value is stored in the FCMP registers (Register 0x40E, Register 0x416, Register 0x41E, and Register 0x426). | 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 (Σ[0x450:0x45C] modulo 256). | | |
| | | [5:1] | RESERVED | | Reserved. | 0x0 | R |
| | | 0 | LINK_EN | | Link enable. Enable the link only after the following has occurred: all JESD204B parameters are set, the DAC PLL is enabled and locked (Register 0x084[1] = 1), and the JESD204B PHY is enabled (Register 0x200 = 0x00) and calibrated (Register 0x281[2] = 0). | 0x0 | R/W |
| 0x301 | GENERAL_JRX_CTRL_1 | [7:3] | RESERVED | | Reserved. | 0x0 | R/W |
| | | [2:0] | SUBCLASSV_LOCAL | | JESD204B Subclass. | 0x1 | R/W |
| | | | | 000 | Subclass 0. | | |
| | | | | 001 | Subclass 1. | | |
| 0x302 | DYN_LINK_LATENCY_0 | [7:5] | RESERVED | | Reserved. | 0x0 | R/W |
| | | [4:0] | 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. See the Deterministic Latency section. | 0x0 | R |
| 0x304 | LMFC_DELAY_0 | [7:5] | RESERVED | | Reserved. | 0x0 | R/W |
| | | [4:0] | LMFC_DELAY_0 | | LMFC delay: Link 0 Delay from the LMFC to $LMFC_{Rx}$ for Link 0. In units of frame clock cycles for subclass 1 and PCLK cycles for subclass 0. See the Deterministic Latency section. | 0x0 | R/W |
| 0x306 | LMFC_VAR_0 | [7:5] | RESERVED | | Reserved. | 0x0 | R/W |
| | | [4:0] | LMFC_VAR_0 | | Variable delay buffer: Link 0. Sets when data is read from a buffer to be consistent across links and power cycles. In units of PCLK cycles. See the Deterministic Latency section.<br>This setting must not be more than 10. | 0x6 | R/W |
| 0x308 | XBAR_LN_0_1 | [7:6] | RESERVED | | Reserved. | 0x0 | R/W |
| | | [5:3] | LOGICAL_LANE1_SRC | | Logical Lane 1 source. Selects a physical lane to be mapped onto Logical Lane 1. | 0x1 | R/W |
| | | | | x | Data is from SERDINx. | | |
| | | [2:0] | LOGICAL_LANE0_SRC | | Logical Lane 0 source. Selects a physical lane to be mapped onto Logical Lane 0. | 0x0 | R/W |
| | | | | x | Data is from SERDINx. | | |
| 0x309 | XBAR_LN_2_3 | [7:6] | RESERVED | | Reserved. | 0x0 | R/W |
| | | [5:3] | LOGICAL_LANE3_SRC | | Logical Lane 3 source. Selects a physical lane to be mapped onto Logical Lane 3. | 0x3 | R/W |
| | | | | x | Data is from SERDINx. | | |
| | | [2:0] | LOGICAL_LANE2_SRC | | Logical Lane 2 source. Selects a physical lane to be mapped onto Logical Lane 2. | 0x2 | R/W |
| | | | | x | Data is from SERDINx. | | |
| 0x30C | FIFO_STATUS_REG_0 | [7:4] | RESERVED | | Reserved. | 0x0 | R |
| | | [3:0] | LANE_FIFO_FULL | | FIFO full flags for each logical lane. A full FIFO indicates an error in the JESD204B configuration or with a system clock. If the FIFO for Lane x is full, Bit x in this register is high. | 0x0 | R |

| Addr. | Name | Bits | Bit Name | Settings | Description | Reset | Access |
|---|---|---|---|---|---|---|---|
| 0x30D | FIFO_STATUS_REG_1 | [7:4] | RESERVED | | Reserved. | 0x0 | R |
| | | [3:0] | LANE_FIFO_EMPTY | | FIFO empty flags for each logical lane. An empty FIFO indicates an error in the JESD204B configuration or with a system clock. If the FIFO for Logical Lane x is empty, Bit x in this register is high. | 0x0 | R |
| 0x311 | SYNCB_GEN_0 | [7:4] | RESERVED | | Reserved. | 0x0 | R/W |
| | | 3 | RESERVED | | Reserved. | 0x0 | R/W |
| | | 2 | EOMF_MASK_0 | | Mask EOMF from QBD. Assert $\overline{\text{SYNCOUT}}\pm$ based on loss of multiframe sync. | 0x0 | R/W |
| | | | | 0 | Do not assert $\overline{\text{SYNCOUT}}\pm$ on Loss of multiframe. | | |
| | | | | 1 | Assert $\overline{\text{SYNCOUT}}\pm$ on loss of multiframe. | | |
| | | 1 | RESERVED | | Reserved. | 0x0 | R/W |
| | | 0 | EOF_MASK_0 | | Mask EOF from QBD. Assert $\overline{\text{SYNCOUT}}\pm$ based on loss of frame sync. | 0x0 | R/W |
| | | | | 0 | Do not assert $\overline{\text{SYNCOUT}}\pm$ on loss of frame. | | |
| | | | | 1 | Assert $\overline{\text{SYNCOUT}}\pm$ on loss of frame. | | |
| 0x312 | SYNCB_GEN_1 | [7:4] | SYNCB_ERR_DUR | | Duration of $\overline{\text{SYNCOUT}}\pm$ low for error. A sync error is asserted at the end of a multiframe whenever one or more disparity, not in table, or unexpected control character errors are encountered. | 0x0 | R/W |
| | | | | 0 | ½ PCLK cycle. | | |
| | | | | 1 | 1 PCLK cycle. | | |
| | | | | 2 | 2 PCLK cycles. | | |
| | | [3:0] | SYNCB_SYNCREQ_DUR | | Duration of $\overline{\text{SYNCOUT}}\pm$ low for purpose of sync request. 0 means a duration > 5 frame + 9 octets. Add an additional PCLK = 4 octets for each increment of the value. | 0x0 | R/W |
| 0x313 | SYNCB_GEN_3 | [7:0] | LMFC_PERIOD | | LMFC period in PCLK cycle. This is to report the global LMFC period based on PCLK. | 0x0 | R |
| 0x314 | SERDES_SPI_REG | [7:0] | SERDES_SPI_CONFIG | | SERDES SPI configuration. Must be written to 0x01 as part of the physical layer setup step. | 0x0 | R/W |
| 0x315 | PHY_PRBS_TEST_EN | [7:4] | RESERVED | | Reserved. | 0x0 | R/W |
| | | [3:0] | PHY_TEST_EN | | PHY test enable. Enables the PHY BER test. Set Bit x to enable the PHY test for Lane x. | 0x0 | R/W |
| 0x316 | PHY_PRBS_TEST_CTRL | 7 | RESERVED | | Reserved. | 0x0 | R/W |
| | | [6:4] | PHY_SRC_ERR_CNT | | PHY error count source. Selects which PHY errors are being reported in Register 0x31A to Register 0x31C. | 0x0 | R/W |
| | | | | x | Report Lane x error count. | | |
| | | [3:2] | PHY_PRBS_PAT_SEL | | PHY PRBS pattern select. Selects the PRBS pattern for the PHY BER test. | 0x0 | R/W |
| | | | | 00 | PRBS7. | | |
| | | | | 01 | PRBS15. | | |
| | | | | 10 | PRBS31. | | |
| | | | | 11 | Reserved. | | |
| | | 1 | PHY_TEST_START | | PHY PRBS test start. Starts and stops the PHY PRBS test. | 0x0 | R/W |
| | | | | 0 | Test stopped. | | |
| | | | | 1 | Test in progress. | | |
| | | 0 | PHY_TEST_RESET | | PHY PRBS test reset. Resets the PHY PRBS test state machine and error counters. | 0x0 | R/W |
| | | | | 0 | Enable PHY PRBS test state machine. | | |
| | | | | 1 | Hold PHY PRBS test state machine in reset. | | |
| 0x317 | PHY_PRBS_TEST_THRESHOLD_LOBITS | [7:0] | PHY_PRBS_THRESHOLD[7:0] | | 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[15:8] | | 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[23:16] | | 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_ | [7:0] | PHY_PRBS_ERR_CNT[7:0] | | Bits[7:0] of the 24-bit reported PHY BERT error | 0x0 | R |

| Addr. | Name | Bits | Bit Name | Settings | Description | Reset | Access |
|---|---|---|---|---|---|---|---|
| | ERRCNT_LOBITS | | | | count from the selected lane. | | |
| 0x31B | PHY_PRBS_TEST_ ERRCNT_MIDBITS | [7:0] | PHY_PRBS_ERR_ CNT[15:8] | | Bits[15:8] of the 24-bit reported PHY BERT error count from the selected lane. | 0x0 | R |
| 0x31C | PHY_PRBS_TEST_ ERRCNT_HIBITS | [7:0] | PHY_PRBS_ERR_ CNT[23:16] | | Bits[23:16] of the 24-bit reported PHY BERT error count from the selected lane. | 0x0 | R |
| 0x31 D | PHY_PRBS_TEST_ STATUS | [7:4] | RESERVED | | Reserved. | 0x0 | R |
| | | [3:0] | PHY_PRBS_PASS | | PHY PRBS test pass/fail.<br>Bit x corresponds to PHY PRBS pass/fail for Physical Lane x. The bit is set to 1 while the error count for Physical Lane x is less than PHY_PRBS_THRESHOLD. | 0xF | R |
| 0x32C | SHORT_TPL_TEST_0 | [7:6] | RESERVED | | Reserved. | 0x0 | R/W |
| | | [5:4] | SHORT_TPL_SP_SEL | | Short transport layer sample select. Selects which sample to check from the DAC selected via Bits[3:2]. | 0x0 | R/W |
| | | | | x | Sample x. | | |
| | | [3:2] | SHORT_TPL_DAC_SEL | | Short transport layer test DAC select. Selects which DAC to sample. | 0x0 | R/W |
| | | | | x | Sample from DAC x. | | |
| | | 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. See the Subclass 0 section for details on how to perform this test. | 0x0 | R/W |
| | | | | 0 | Disable. | | |
| | | | | 1 | Enable. | | |
| 0x32 D | SHORT_TPL_TEST_1 | [7:0] | SHORT_TPL_REF_SP_LSB | | Short transport layer test reference, sample LSB. This is the lower eight bits of the expected DAC sample. It is used to compare with the received DAC sample at the output of the JESD204B receiver. | 0x0 | R/W |
| 0x32E | SHORT_TPL_TEST_2 | [7:0] | SHORT_TPL_REF_SP_MSB | | Short transport layer test reference, sample MSB. This is the upper eight bits of the expected DAC sample. It is used to compare with the received DAC sample at the output of the JESD204B receiver. | 0x0 | R/W |
| 0x32F | SHORT_TPL_TEST_3 | [7:1] | RESERVED | | Reserved. | 0x0 | R |
| | | 0 | SHORT_TPL_FAIL | | Short transport layer test fail. This bit shows whether the selected DAC sample matches the reference sample. If they match, it is a test pass, otherwise it is a test fail. | 0x0 | R |
| | | | | 0 | Test pass. | | |
| | | | | 1 | Test fail. | | |
| 0x334 | JESD_BIT_INVERSE_CTRL | [7:4] | RESERVED | | Reserved. | 0x0 | R/W |
| | | [3:0] | JESD_BIT_INVERSE | | Logical lane invert. Set Bit x high to invert the JESD204B deserialized data on Logical Lane x. | 0x0 | R/W |
| 0x400 | DID_REG | [7:0] | DID_RD | | Device identification number. Link information received on Lane 0 as specified in Section 8.3 of JESD204B. | 0x0 | R |
| 0x401 | BID_REG | [7:4] | ADJCNT_RD | | Adjustment resolution to DAC LMFC. Link information received on Lane 0 as specified in Section 8.3 of JESD204B. | 0x0 | R |
| | | [3:0] | BID_RD | | Bank identification: extension to DID. Link information received on Lane 0 as specified in Section 8.3 of JESD204B. | 0x0 | R |
| 0x402 | LID0_REG | 7 | RESERVED | | Reserved. | 0x0 | R |
| | | 6 | ADJDIR_RD | | Direction to adjust DAC LMFC. Link information received on Lane 0 as specified in Section 8.3 of JESD204B. Must be 0. | 0x0 | R |
| | | 5 | PHADJ_RD | | Phase adjustment request to DAC. Link information received on Lane 0 as specified in Section 8.3 of JESD204B. Must be 0. | 0x0 | R |

| Addr. | Name | Bits | Bit Name | Settings | Description | Reset | Access |
|---|---|---|---|---|---|---|---|
| | | [4:0] | LID0_RD | | Lane identification for Lane 0. Link information received on Lane 0 as specified in Section 8.3 of JESD204B. | 0x0 | R |
| 0x403 | SCR_L_REG | 7 | SCR_RD | | Transmit scrambling status. Link information received on Lane 0 as specified in Section 8.3 of JESD204B. | 0x0 | R |
| | | | | 0 | Scrambling is disabled. | | |
| | | | | 1 | Scrambling is enabled. | | |
| | | [6:5] | RESERVED | | Reserved. | 0x0 | R |
| | | [4:0] | L-1_RD | | Number of lanes per converter device. Link information received on Lane 0 as specified in Section 8.3 of JESD204B. | 0x0 | R |
| | | | | 0 | One lane per converter device. | | |
| | | | | 1 | Two lanes per converter device. | | |
| 0x404 | F_REG | [7:0] | F-1_RD | | Number of octets per frame. Settings of 1, 2, and 4 octets per frame are valid. Link information received on Lane 0 as specified in Section 8.3 of JESD204B. | 0x0 | R |
| | | | | 0 | One octet per frame. | | |
| | | | | 1 | Two octets per frame. | | |
| | | | | 3 | Four octets per frame. | | |
| 0x405 | K_REG | [7:5] | RESERVED | | Reserved. | 0x0 | R |
| | | [4:0] | K-1_RD | | Number of frames per multiframe. Settings of 16 or 32 are valid. Link information received on Lane 0 as specified in Section 8.3 of JESD204B. | 0x0 | R |
| | | | | 0x0F | 16 frames per multiframe. | | |
| | | | | 0x1F | 32 frames per multiframe. | | |
| 0x406 | M_REG | [7:0] | M-1_RD | | Number of converters per device. Link information received on Lane 0 as specified in Section 8.3 of JESD204B. Must be 0 or 1. | 0x0 | R |
| | | | | 0 | One converter per device. | | |
| | | | | 1 | Two converters per device. | | |
| 0x407 | CS_N_REG | [7:6] | CS_RD | | Number of control bits per sample. Link information received on Lane 0 as specified in Section 8.3 of JESD204B. CS_RD must be set to 0. | 0x0 | R |
| | | 5 | RESERVED | | Reserved. | 0x0 | R |
| | | [4:0] | N-1_RD | | Converter resolution. Link information received on Lane 0 as specified in Section 8.3 of JESD204B. Converter resolution of 16. | 0x0 | R |
| | | | | 0x0F | Converter resolution of 16. | | |
| 0x408 | NP_REG | [7:5] | SUBCLASSV_RD | | Device subclass version. Link information received on Lane 0 as specified in Section 8.3 of JESD204B. | 0x0 | R |
| | | | | 0 | Subclass 0. | | |
| | | | | 1 | Subclass 1. | | |
| | | [4:0] | NP-1_RD | | Total number of bits per sample. Link information received on Lane 0 as specified in Section 8.3 of JESD204B. Must be 16 bits per sample. | 0x0 | R |
| | | | | 0x0F | 16 bits per sample. | | |
| 0x409 | S_REG | [7:5] | JESDV_RD | | JESD204 version. Link information received on Lane 0 as specified in Section 8.3 of JESD204B. | 0x0 | R |
| | | | | 000 | JESD204A. | | |
| | | | | 001 | JESD204B. | | |
| | | [4:0] | S-1_RD | | Number of samples per converter per frame cycle. Settings of one and two are valid. See Table 33. Link information received on Lane 0 as specified in Section 8.3 of JESD204B. | 0x0 | R |
| | | | | 0 | One sample per converter per frame. | | |
| | | | | 1 | Two samples per converter per frame. | | |
| 0x40A | HD_CF_REG | 7 | HD_RD | | High density format. See Section 5.1.3 of the JESD294B standard. Link information received on Lane 0 as specified in Section 8.3 of JESD204B. | 0x0 | R |
| | | | | 0 | Low density mode. | | |
| | | | | 1 | High density mode: link information received on Lane 0 as specified in Section 8.3 of JESD204B. | | |
| | | [6:5] | RESERVED | | Reserved. | 0x0 | R |

| Addr. | Name | Bits | Bit Name | Settings | Description | Reset | Access |
|---|---|---|---|---|---|---|---|
| | | [4:0] | CF_RD | | Number of control words per frame clock period per link. Link information received on Lane 0 as specified in Section 8.3 of JESD204B. Bits[4:0] must be 0. | 0x0 | R |
| 0x40B | RES1_REG | [7:0] | RES1_RD | | Reserved Field 1. Link information received on Lane 0 as specified in Section 8.3 of JESD204B. | 0x0 | R |
| 0x40C | RES2_REG | [7:0] | RES2_RD | | Reserved Field 2. Link information received on Lane 0 as specified in Section 8.3 of JESD204B. | 0x0 | R |
| 0x40D | CHECKSUM_REG | [7:0] | FCHK0_RD | | Checksum for Lane 0. Link information received on Lane 0 as specified in Section 8.3 of JESD204B. | 0x0 | R |
| 0x40E | COMPSUM0_REG | [7:0] | FCMP0_RD | | Computed checksum for Lane 0. The JESD204B receiver computes the checksum of the link information received on Lane 0 as specified in Section 8.3 of JESD204B. The computation method is set by the CHECKSUM_MODE bit (Address 0x300[6]) and must match the likewise calculated checksum in Register 0x40D. | 0x0 | R |
| 0x412 | LID1_REG | [7:5] | RESERVED | | Reserved. | 0x0 | R |
| | | [4:0] | LID1_RD | | Lane identification for Lane 1. Link information received on Lane 0 as specified in Section 8.3 of JESD204B. | 0x0 | R |
| 0x415 | CHECKSUM1_REG | [7:0] | FCHK1_RD | | Checksum for Lane 1. Link information received on Lane 0 as specified in Section 8.3 of JESD204B. | 0x0 | R |
| 0x416 | COMPSUM1_REG | [7:0] | FCMP1_RD | | Computed checksum for Lane 1. See the description for Register 0x40E. | 0x0 | R |
| 0x41A | LID2_REG | [7:5] | RESERVED | | Reserved. | 0x0 | R |
| | | [4:0] | LID2_RD | | Lane identification for Lane 2. | 0x0 | R |
| 0x41D | CHECKSUM2_REG | [7:0] | FCHK2_RD | | Checksum for Lane 2. | 0x0 | R |
| 0x41E | COMPSUM2_REG | [7:0] | FCMP2_RD | | Computed checksum for Lane 2 (see the description for Register 0x40E). | 0x0 | R |
| 0x422 | LID3_REG | [7:5] | RESERVED | | Reserved. | 0x0 | R |
| | | [4:0] | LID3_RD | | Lane identification for Lane 3. | 0x0 | R |
| 0x425 | CHECKSUM3_REG | [7:0] | FCHK3_RD | | Checksum for Lane 3. | 0x0 | R |
| 0x426 | COMPSUM3_REG | [7:0] | FCMP3_RD | | Computed checksum for Lane 3 (see the description for Register 0x40E). | 0x0 | R |
| 0x450 | ILS_DID | [7:0] | DID | | Device identification number. Link information received on Lane 0 as specified in Section 8.3 of JESD204B. Must be set to the value read in Register 0x400. | 0x0 | R/W |
| 0x451 | ILS_BID | [7:4] | ADJCNT | | Adjustment resolution to DAC LMFC. Must be set to 0. | 0x0 | R/W |
| | | [3:0] | BID | | Bank identification: extension to DID. Must be set to the value read in Register 0x401[3:0]. | 0x0 | R/W |
| 0x452 | ILS_LID0 | 7 | RESERVED | | Reserved. | 0x0 | R/W |
| | | 6 | ADJDIR | | Direction to adjust DAC LMFC. Must be set to 0. | 0x0 | R/W |
| | | 5 | PHADJ | | Phase adjustment request to DAC. Must be set to 0. | 0x0 | R/W |
| | | [4:0] | LID0 | | Lane identification for Lane 0. Must be set to the value read in Register 0x402[4:0]. | 0x0 | R/W |
| 0x453 | ILS_SCR_L | 7 | SCR | | Receiver descrambling enable. | 0x1 | R/W |
| | | | | 0 | Descrambling is disabled. | | |
| | | | | 1 | Descrambling is enabled. | | |
| | | [6:5] | RESERVED | | Reserved. | 0x0 | R/W |
| | | [4:0] | L-1 | | Number of lanes per converter device. See Table 33. | 0x3 | R/W |
| | | | | 0 | One lane per converter. | | |
| | | | | 1 | Two lanes per converter. | | |
| 0x454 | ILS_F | [7:0] | F-1 | | Number of octets per lane per frame. Settings of 1, 2, and 4 (octets per lane) per frame are valid. See Table 33. | 0x0 | R/W |
| | | | | 0 | (One octet per lane) per frame. | | |
| | | | | 1 | (Two octets per lane) per frame. | | |
| | | | | 3 | (Four octets per lane) per frame. | | |

| Addr. | Name | Bits | Bit Name | Settings | Description | Reset | Access |
|---|---|---|---|---|---|---|---|
| 0x455 | ILS_K | [7:5] | RESERVED | | Reserved. | 0x0 | R/W |
| | | [4:0] | K-1 | | Number of frames per multiframe. Settings of 16 or 32 are valid. Must be set to 32 when F = 1 (Register 0x476). | 0x1F | R/W |
| | | | | 0x0F | 16 frames per multiframe. | | |
| | | | | 0x1F | 32 frames per multiframe. | | |
| 0x456 | ILS_M | [7:0] | M-1 | | Number of converters per device. See Table 33. | 0x1 | R/W |
| | | | | 0 | One converter per link. | | |
| | | | | 1 | Two converters per link. | | |
| 0x457 | ILS_CS_N | [7:6] | CS | | Number of control bits per sample. Must be set to 0. Control bits are not supported. | 0x0 | R/W |
| | | | | 0 | Zero control bits per sample. | | |
| | | 5 | RESERVED | | Reserved. | 0x0 | R/W |
| | | [4:0] | N-1 | | Converter Resolution. Must be set to 16 bits of resolution. | 0xF | R/W |
| | | | | 0xF | Converter resolution of 16. | | |
| 0x458 | ILS_NP | [7:5] | SUBCLASSV | | Device subclass version. | 0x1 | R/W |
| | | | | 0 | Subclass 0. | | |
| | | | | 1 | Subclass 1. | | |
| | | [4:0] | NP-1 | | Total number of bits per sample. Must be set to 16 bits per sample. | 0xF | R/W |
| | | | | 0xF | 16 bits per sample. | | |
| 0x459 | ILS_S | [7:5] | JESDV | | JESD204 version. | 0x1 | R/W |
| | | | | 000 | JESD204A. | | |
| | | | | 001 | JESD204B. | | |
| | | [4:0] | S-1 | | Number of samples per converter per frame cycle. Settings of one and two are valid. See Table 33. | 0x0 | R/W |
| | | | | 0 | One sample per converter per frame. | | |
| | | | | 1 | Two samples per converter per frame. | | |
| 0x45A | ILS_HD_CF | 7 | HD | | High density format. If F = 1, HD must be set to 1. Otherwise, HD must be set to 0. See Section 5.1.3 of JESD204B standard. | 0x1 | R/W |
| | | | | 0 | Low density mode. | | |
| | | | | 1 | High density mode. | | |
| | | [6:5] | RESERVED | | Reserved. | 0x0 | R/W |
| | | [4:0] | CF | | Number of control words per frame clock period per link. Must be set to 0. Control bits are not supported. | 0x0 | R/W |
| 0x45B | ILS_RES1 | [7:0] | RES1 | | Reserved Field 1. | 0x0 | R/W |
| 0x45C | ILS_RES2 | [7:0] | RES2 | | Reserved Field 2. | 0x0 | R/W |
| 0x45D | ILS_CHECKSUM | [7:0] | FCHK0 | | Checksum for Lane 0. Calculated checksum. Calculation depends on 0x300[6]. | 0x45 | R/W |
| 0x46B | ERRCNTRMON_RB | [7:0] | READERRORCNTR | | Read JESD204B error counter. After selecting the lane and error counter by writing to LANESEL and CNTRSEL (both in this same register), the selected error counter is read back here. | 0x0 | R |
| 0x46B | ERRCNTRMON | 7 | RESERVED | | Reserved. | 0x0 | R |
| | | [6:4] | LANESEL | | Lane select for JESD204B error counter. Selects the lane whose errors are read back in this register. | 0x0 | W |
| | | | | x | Selects Lane x. | | |
| | | [3:2] | RESERVED | | Reserved. | 0x0 | R |
| | | [1:0] | CNTRSEL | | JESD204B error counter select. Selects the type of error that are read back in this register. | 0x0 | W |
| | | | | 00 | BADDISCNTR: bad running disparity counter. | | |
| | | | | 01 | NITCNTR: not in table error counter. | | |
| | | | | 10 | UCCCNTR: Unexpected control character counter. | | |
| 0x46C | LANEDESKEW | [7:0] | LANEDESKEW | | Lane deskew. Setting Bit x deskews Lane x. | 0xF | R/W |
| 0x46D | BADDISPARITY_RB | [7:0] | BADDIS | | Bad disparity character error (BADDIS). Bit x is set when the bad disparity error count for Lane x reaches the threshold in Register 0x47C. | 0x0 | R |
| 0x46D | BADDISPARITY | 7 | RST_IRQ_DIS | | BADDIS IRQ reset. Reset BADDIS IRQ for the lane selected via Bits[2:0] by writing 1 to this bit. | 0x0 | W |

| Addr. | Name | Bits | Bit Name | Settings | Description | Reset | Access |
|---|---|---|---|---|---|---|---|
| | | 6 | DISABLE_ERR_CNTR_DIS | | BADDIS error counter disable. Disable the BADDIS error counter for the lane selected via Bits[2:0] by writing 1 to this bit. | 0x0 | W |
| | | 5 | RST_ERR_CNTR_DIS | | BADDIS error counter reset. Reset the BADDIS error counter for the lane selected via Bits[2:0] by writing 1 to this bit. | 0x0 | W |
| | | [4:3] | RESERVED | | Reserved. | 0x0 | R |
| | | [2:0] | LANE_ADDR_DIS | | Lane address for functions described in Bits[7:5]. | 0x0 | W |
| 0x46E | NIT_RB | [7:0] | NIT | | Not in table character error (NIT). Bit x is set when the NIT error count for Lane x reaches the threshold in Register 0x47C. | 0x0 | R |
| 0x46E | NIT_W | 7 | RST_IRQ_NIT | | IRQ reset. Reset the IRQ for the lane selected via Bits[2:0] by writing 1 to this bit. | 0x0 | W |
| | | 6 | DISABLE_ERR_CNTR_NIT | | Disable error counter. Disable the error counter for the lane selected via Bits[2:0] by writing 1 to this bit. | 0x0 | W |
| | | 5 | RST_ERR_CNTR_NIT | | Reset error counter. Reset error counter for the lane selected via Bits[2:0] by writing 1 to this bit. | 0x0 | W |
| | | [4:3] | RESERVED | | Reserved. | 0x0 | R |
| | | [2:0] | LANE_ADDR_NIT | | Lane address for functions described in Bits[7:5]. | 0x0 | W |
| 0x46F | UNEXPECTED_CONTROL_RB | [7:0] | UCC | | Unexpected control character error (UCC). Bit x is set when the UCC error count for Lane x reaches the threshold in Register 0x47C. | 0x0 | R |
| 0x46F | UNEXPECTED_CONTROL_W | 7 | RST_IRQ_UCC | | IRQ reset. Reset IRQ for the lane selected via Bits[2:0] by writing 1 to this bit. | 0x0 | W |
| | | 6 | DISABLE_ERR_CNTR_UCC | | Disable error counter. Disable the error counter for the lane selected via Bits[2:0] by writing 1 to this bit. | 0x0 | W |
| | | 5 | RST_ERR_CNTR_UCC | | Reset error counter. Reset error counter for the lane selected via Bits[2:0] by writing 1 to this bit. | 0x0 | W |
| | | [4:3] | RESERVED | | Reserved. | 0x0 | R |
| | | [2:0] | LANE_ADDR_UCC | | Lane Address for functions described in Bits[7:5]. | 0x0 | W |
| 0x470 | CODEGRPSYNCFLG | [7:0] | CODEGRPSYNC | | Code group sync flag (from each instantiated lane). Writing 1 to Bit 7 resets the IRQ. The associated IRQ flag is located in Register 0x47A[0]. A loss of CODEGRPSYNC triggers sync request assertion. See the SYNCOUT and SYSREF Signals section and the Deterministic Latency section. | 0x0 | R/W |
| | | | | 0 | Synchronization is lost. | | |
| | | | | 1 | Synchronization is achieved. | | |
| 0x471 | FRAMESYNCFLG | [7:0] | FRAMESYNC | | Frame sync flag (from each instantiated lane). This register indicates the live status for each lane. Writing 1 to Bit 7 resets the IRQ. A loss of frame sync automatically initiates a synchronization sequence. | 0x0 | R/W |
| | | | | 0 | Synchronization is lost. | | |
| | | | | 1 | Synchronization is achieved. | | |
| 0x472 | GOODCHKSUMFLG | [7:0] | GOODCHECKSUM | | Good checksum flag (from each instantiated lane). Writing 1 to Bit 7 resets the IRQ. The associated IRQ flag is located in Register 0x47A[2]. | 0x0 | R/W |
| | | | | 0 | Last computed checksum is not correct. | | |
| | | | | 1 | Last computed checksum is correct. | | |
| 0x473 | INITLANESYNCFLG | [7:0] | INITIALLANESYNC | | Initial lane syncflag (from each instantiated lane). Writing 1 to Bit 7 resets the IRQ. The associated IRQ flag is located in Register 0x47A[3]. Loss of synchronization is also reported on SYNCOUT±. See the SYNCOUT±, SYSREF±, and DACCLK±/REFCLK± Signals section and the Deterministic Latency section. | 0x0 | R/W |
| 0x476 | CTRLREG1 | [7:0] | F | | Number of Octets per Frame. Settings of 1, 2, and 4 are valid. See Table 33. | 0x1 | R/W |
| | | | | 1 | One octet per frame. | | |
| | | | | 2 | Two octets per frame. | | |
| | | | | 4 | Four octets per frame. | | |
| 0x477 | CTRLREG2 | 7 | ILAS_MODE | | ILAS test mode. Defined in Section 5.3.3.8 of JESD204B specification. | 0x0 | R/W |

| Addr. | Name | Bits | Bit Name | Settings | Description | Reset | Access |
|---|---|---|---|---|---|---|---|
| | | | | 1 | JESD204B receiver is constantly receiving ILAS frames. | | |
| | | | | 0 | Normal link operation. | | |
| | | [6:4] | RESERVED | | Reserved. | 0x0 | R/W |
| | | 3 | THRESHOLD_MASK_EN | | Threshold mask enable. Set this bit if using SYNC_ASSERTION_MASK (Register 0x47B[7:5]). | 0x0 | R/W |
| | | [2:0] | RESERVED | | Reserved. | 0x0 | R/W |
| 0x478 | KVAL | [7:0] | KSYNC | | Number of K multiframes during ILAS (divided by four). Sets the number of multiframes to send initial lane alignment sequence. Cannot be set to 0. | 0x1 | R/W |
| | | | | x | 4x multiframes during ILAS. | | |
| 0x47A | IRQVECTOR_FLAG | 7 | BADDIS_FLAG | | Bad disparity error Count. | 0x0 | R |
| | | | | 1 | Bad disparity character count reached ERRORTHRESH (0x47C) on at least one lane. Read Register 0x46D to determine which lanes are in error. | | |
| | | 6 | NIT_FLAG | | Not in table error count. | 0x0 | R |
| | | | | 1 | Not in table character count reached ERRORTHRESH (0x47C) on at least one lane. Read Register 0x46E to determine which lanes are in error. | | |
| | | 5 | UCC_FLAG | | Unexpected control character error count. | 0x0 | R |
| | | | | 1 | Unexpected control character count reached ERRORTHRESH (0x47C) on at least one lane. Read Register 0x46F to determine which lanes are in error. | | |
| | | 4 | RESERVED | | Reserved. | 0x0 | R |
| | | 3 | INITIALLANESYNC_FLAG | | Initial lane sync fag. | 0x0 | R |
| | | | | 1 | Initial lane sync failed on at least one lane. Read Register 0x473 to determine which lanes are in error. | | |
| | | 2 | BADCHECKSUM_FLAG | | Bad checksum flag. | 0x0 | R |
| | | | | 1 | Bad checksum on at least one lane. Read Register 0x472 to determine which lanes are in error. | | |
| | | 1 | FRAMESYNC_FLAG | | Frame sync flag. | 0x0 | R |
| | | | | 1 | Frame sync failed on at least one lane. Read Register 0x471 to determine which lanes are in error. | | |
| | | 0 | CODEGRPSYNC_FLAG | | Code group sync flag. | 0x0 | R |
| | | | | 1 | Code group sync failed on at least one lane. Read Register 0x470 to determine which lanes are in error. | | |
| 0x47A | IRQVECTOR_MASK | 7 | BADDIS_MASK | | Bad disparity mask. | 0x0 | W |
| | | | | 1 | If the bad disparity count reaches ERRORTHRESH on any lane, $\overline{IRQ}$ is pulled low. | | |
| | | 6 | NIT_MASK | | Not in table mask. | 0x0 | W |
| | | | | 1 | If the not in table character count reaches ERRORTHRESH on any lane, $\overline{IRQ}$ is pulled low. | | |
| | | 5 | UCC_MASK | | Unexpected control character mask. | 0x0 | W |
| | | | | 1 | If the unexpected control character count reaches ERRORTHRESH on any lane, $\overline{IRQ}$ is pulled low. | | |
| | | 4 | RESERVED | | Reserved. | 0x0 | R |
| | | 3 | INITIALLANESYNC_MASK | | Initial lane sync mask. | 0x0 | W |
| | | | | 1 | If initial lane sync (0x473) fails on any lane, $\overline{IRQ}$ is pulled low. | | |
| | | 2 | BADCHECKSUM_MASK | | Bad checksum mask. | 0x0 | W |
| | | | | 1 | If there is a bad checksum (0x472) on any lane, $\overline{IRQ}$ is pulled low. | | |
| | | 1 | FRAMESYNC_MASK | | Frame sync mask. | 0x0 | W |
| | | | | 1 | If frame sync (0x471) fails on any lane, $\overline{IRQ}$ is pulled low. | | |
| | | 0 | CODEGRPSYNC_MASK | | Code group sync machine mask. | 0x0 | W |
| | | | | 1 | If code group sync (0x470) fails on any lane, $\overline{IRQ}$ is pulled low. | | |
| 0x47B | SYNCASSERTIONMASK | 7 | BADDIS_S | | Bad disparity error on sync. | 0x0 | R/W |
| | | | | 1 | The deframer asserts the $\overline{SYNCOUT}\pm$ signal when the bad disparity error count reaches the threshold in Register 0x47C. | | |

| Addr. | Name | Bits | Bit Name | Settings | Description | Reset | Access |
|---|---|---|---|---|---|---|---|
| | | | | 0 | The deframer does not assert the SYNCOUT± when the bad disparity error count reaches the threshold in Register 0x47C. | | |
| | | 6 | NIT_S | | Not in table disparity character error on sync. | 0x0 | R/W |
| | | | | 1 | The deframer asserts the SYNCOUT± signal when the not in table disparity character error count reaches the threshold in Register 0x47C. | | |
| | | | | 0 | The deframer does not assert the SYNCOUT± when the not in table disparity character error count reaches the threshold on Register 0x47C. | | |
| | | 5 | UCC_S | | Unexpected K character error on sync. | 0x0 | R/W |
| | | | | 1 | The deframer asserts the SYNCOUT± signal when the unexpected K character error count reaches the threshold in Register 0x47C. | | |
| | | | | 0 | The deframer does not assert the SYNCOUT± when the unexpected K character error count reaches the threshold in Register 0x47C. | | |
| | | 4 | CMM | | Writing a 1 resets the CMM_IRQ (applicable if Register 0x47B, Bit 3 = 1) Configuration mismatch flag when read. | 0x0 | R/W |
| | | | | 1 | Lane 0 configuration registers (Register 0x450 to Register 0x45D) do not match the comparable JESD204B transmit settings as reported in Register 0x400 to Register 0x40D. | | |
| | | | | 0 | Lane 0 configuration registers (Register 0x450 to Register 0x45D) match the comparable JESD Tx settings as reported in Register 0x400 to Register 0x40D. | | |
| | | 3 | CMM_ENABLE | | Configuration mismatch IRQ enable. | 0x1 | R/W |
| | | | | 1 | Enables IRQ generation if a configuration mismatch is detected. | | |
| | | | | 0 | Configuration mismatch IRQ disabled. | | |
| | | [2:0] | RESERVED | | | 0x0 | R/W |
| 0x47C | ERRORTHRES | [7:0] | ETH | | Error threshold. Bad disparity, not in table, and unexpected control character errors are counted and compared to the error threshold value. When the count reaches the threshold, either an IRQ is generated or the SYNCOUT± signal is asserted per the mask register settings, or both. Function is performed in all lanes. | 0xFF | R/W |
| 0x47D | LANEENABLE | [7:4] | RESERVED | | Reserved. | 0x0 | R |
| | | [3:0] | LANE_ENA | | Lane Enable. Setting Bit x enables Lane x. This register must be programmed before receiving the code group pattern for proper operation. | 0xF | R/W |
| 0x47E | RAMP_ENA | [7:1] | RESERVED | | Reserved. | 0x0 | R |
| | | 0 | ENA_RAMP_CHECK | | Enable ramp checking at the beginning of ILAS. | 0x0 | W |
| | | | | 0 | Disable ramp checking at beginning of ILAS; ILAS data need not be a ramp. | | |
| | | | | 1 | Enable ramp checking; ILAS data needs to be a ramp starting at 00-01-02; otherwise, the ramp ILAS fails and the device does not start up. | | |

# OUTLINE DIMENSIONS

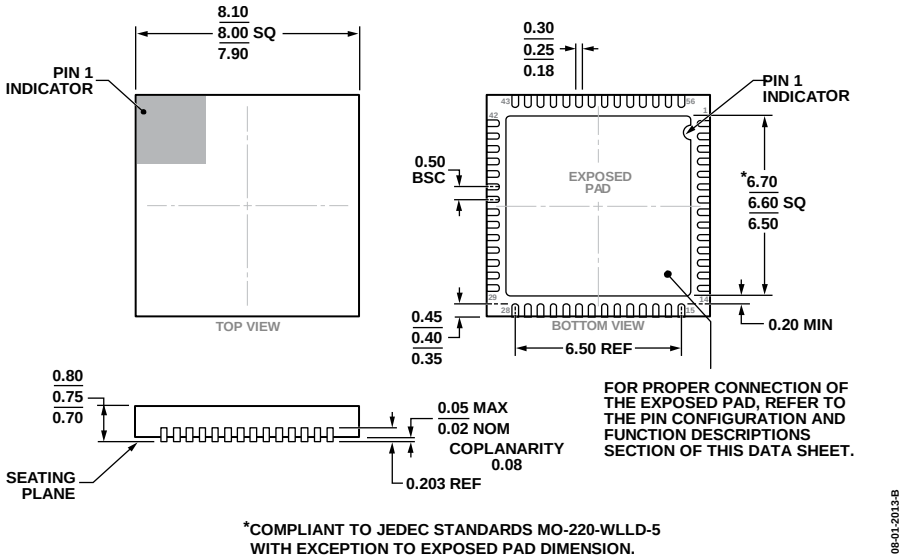


*Figure 78. 56-Lead Lead Frame Chip Scale Package [LFCSP]*
*8 mm × 8 mm Body and 0.75 mm Package Height*
*(CP-56-9)*
*Dimensions shown in millimeters*

## ORDERING GUIDE

| Model[1] | Temperature Range | Package Description | Package Option |
|---|---|---|---|
| AD9152BCPZ | −40°C to +85°C | 56-Lead Lead Frame Chip Scale Package [LFCSP] | CP-56-9 |
| AD9152BCPZRL | −40°C to +85°C | 56-Lead Lead Frame Chip Scale Package [LFCSP] | CP-56-9 |
| AD9152-EBZ | | DPG3 Evaluation Board | |
| AD9152-FMC-EBZ | | FMC Evaluation Board | |
| AD9152-M6720-EBZ | | DPG3 Evaluation Board with ADRF6720 Modulator | |

[1] Z = RoHs Compliant Part.