# 14-Bit, 2.5 GSPS, RF Digital-to-Analog Converter

Data Sheet **AD9739**

## FEATURES

- **Direct RF synthesis at 2.5 GSPS update rate**
  - **DC to 1.25 GHz in baseband mode**
  - **1.25 GHz to 3.0 GHz in mix mode**
- **Industry leading single/multicarrier IF or RF synthesis**
  - **$f_{OUT}$ = 350 MHz, ACLR =80 dBc**
  - **$f_{OUT}$ = 950 MHz, ACLR = 78 dBc**
  - **$f_{OUT}$ = 2100 MHz, ACLR = 69 dBc**
- **Dual-port LVDS data interface**
  - **Up to 1.25 GSPS operation**
  - **Source synchronous DDR clocking**
- **Pin-compatible with the AD9739A**
- **Multichip synchronization capability**
- **Programmable output current: 8.7 mA to 31.7 mA**
- **Low power: 1.16 W at 2.5 GSPS**

## APPLICATIONS

- **Broadband communications systems**
- **Military jammers**
- **Instrumentation, automatic test equipment**
- **Radar, avionics**

## FUNCTIONAL BLOCK DIAGRAM

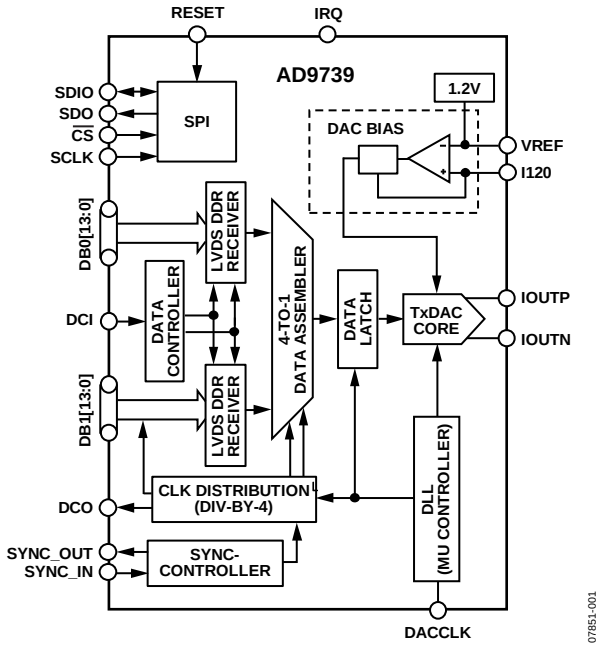



*Figure 1.*

## GENERAL DESCRIPTION

The AD9739 is a 14-bit, 2.5 GSPS high performance RF digital-to-analog converter (DAC) capable of synthesizing wideband signals from dc up to 3.0 GHz. Its DAC core features a quad-switch architecture that provides exceptionally low distortion performance with an industry-leading direct RF synthesis capability. This feature enables multicarrier generation up to the Nyquist frequency in baseband mode as well as second and third Nyquist zones in mix mode. The output current can be programmed over the 8.66 mA to 31.66 mA range.

The inclusion of on-chip controllers simplifies system integration. A dual-port, source synchronous, LVDS interface simplifies the digital interface with existing FGPA/ASIC technology. On-chip controllers are used to manage external and internal clock domain variations over temperature to ensure reliable data transfer from the host to the DAC core. Multichip synchronization is possible with an on-chip synchronization controller. A serial peripheral interface (SPI) is used for device configuration as well as readback of status registers.

The AD9739 is manufactured on a 0.18 µm CMOS process and operates from 1.8 V and 3.3 V supplies. It is supplied in a 160-ball chip scale ball grid array for reduced package parasitics.

## PRODUCT HIGHLIGHTS

1. Ability to synthesize high quality wideband signals with bandwidths of up to 1.25 GHz in the first or second Nyquist zone.
2. A proprietary quad-switch DAC architecture provides exceptional ac linearity performance while enabling mix mode operation.
3. A dual-port, double data rate, LVDS interface supports the maximum conversion rate of 2500 MSPS.
4. On-chip controllers manage external and internal clock domain skews.
5. A multichip synchronization capability.
6. Programmable differential current output with an 8.66 mA to 31.66 mA range.

**Rev. D** **Document Feedback**


**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

**5/2017—Rev. C to Rev. D**

Changes to Table 32 ..... 48

**2/2015—Rev. B to Rev. C**

Moved Revision History ..... 3
Changes to Figure 6 ..... 11
Changes to Table 19 ..... 25
Changes to Theory of Operation Section ..... 28
Changes to Figure 52 ..... 36
Changes to Clock Input Considerations Section ..... 40
Deleted Figure 60 ..... 40
Changes to Table 32 ..... 48

**1/2012—Rev. A to Rev. B**

Changes to Features Section, Applications Section, General Description Section, Figure 1, Product Highlights Section ..... 1
Changes to DC Specifications Section ..... 4
Changed Digital Specifications Section to LVDS Digital Specifications Section ..... 5
Changes to LVDS Digital Specifications Section ..... 5
Added Serial Port Specifications Section and Table 3; Renumbered Sequentially ..... 6
Changes to AC Specifications Section ..... 7
Changes to Table 5 ..... 8
Changes to Table 7 ..... 10
Deleted Static Linearity Section and Figure 7 to Figure 17; Renumbered Sequentially ..... 11
Changed Dynamic Performance Normal Mode, 20 mA Full Scale (Unless Otherwise Noted) Section to AC (Normal Mode) Section ..... 12
Changes to AC (Normal Mode) Section ..... 12
Changed Dynamic Performance Mix Mode, 20 mA Full Scale Section to AC (Mix Mode) Section ..... 15
Changes to AC (Mix Mode) Section ..... 15
Added Serial Port Interface (SPI) Register Section, SPI Register Map Description Section, Reset Section, Table 8, and SPI Operation Section and Figure 34 ..... 18
Deleted DOCSIS Performance Section and Figure 46 to Figure 72 and added Figure 35 through Figure 38; Renumbered Sequentially ..... 19
Changes to SPI Register Map Section and Table 9 ..... 20
Added SPI Port Configuration and Software Reset Section, Power-Down LVDS Interface and TxDAC® Section, Controller Clock Disable Section, Interrupt Request (IRQ) Enable/Status Section, and Table 10 to Table 13 ..... 22
Added TxDAC Full-Scale Current Setting ($I_{OUTFS}$) and Sleep Section, TxDAC Quad-Switch Mode of Operation Section, DCI Phase Alignment Status Section, SYNC_IN Phase Alignment Status Section, Data Receiver Controller Configuration Section, and Table 14 to Table 18 ..... 23
Added Data Receiver Controller_Data Sample Delay Value Section, Data and Sync Receiver Controller_DCI Delay Value/Window and Phase Rotation Section, Data Receiver Controller_Delay Line Status and Sync Controller SYNC_OUT Status Section, and Table 19 to Table 21 ..... 24
Deleted Serial Peripheral Interface Section, General Operation of the Serial Interface Section, Instruction Mode (8-Bit Instruction) Section, and Serial Interface Port Pin Description Section ..... 25
Added Sync and Data Receiver Controller Lock/Tracking Status Section, CLK Input Common Mode Section, Mu Controller Configuration and Status Section, and Table 22 to Table 24 ..... 25
Deleted MSB/LSB Transfers Section, Serial Port Configuration Section, and Figure 74 to Figure 79 ..... 26
Added Part ID Section and Table 25 ..... 26
Changes to Theory of Operation Section ..... 27
Added Figure 39 ..... 27
Deleted SPI Registers Section and Table 8 to Table 31 ..... 28
Moved and Changes to LVDS Data Port Interface Section ..... 28
Added Figure 40 and Figure 41 ..... 28
Changes to Figure 42 ..... 29
Moved and Changes to Figure 43 ..... 29
Added Data Receiver Controller Initialization Description Section, Table 26, and Data Receiver Operation at Lower Clock Rates Section ..... 30
Added LVDS Driver and Receiver Input Section, Figure 44 to Figure 47, and Table 27 ..... 31
Changed and Moved Mu Delay Controller Section to Mu Controller Section ..... 32
Changes to Mu Controller Section, Figure 48, and Figure 49 ..... 32
Added Figure 50 and Table 28 ..... 32
Added Mu Controller Initialization Description Section ..... 33
Changes to Interrupt Requests Section ..... 34
Added Table 29 ..... 34
Changed Synchronization Controller Section to Multiple Device Synchronization Section ..... 35
Added Figure 52 ..... 35
Changes to Figure 53 ..... 36
Added Sync Controller Initialization Description Section ..... 36
Added Synchronization Limitations Section ..... 37
Changed Applications Information to Analog Interface Considerations Section ..... 38
Changes to Analog Modes of Operation Section ..... 38
Deleted Clocking the AD9739 Section, Figure 85, and Figure 86 ..... 39
Added Clock Input Considerations Section, Figure 58 to Figure 60 ..... 39
Deleted Clock Phase Noise Affects on AC Performance Section, Table 32 to Table 34, Applying Data to the AD9739 Section, and Figure 87 ..... 40
Moved Figure 61 ..... 40
Changes to Voltage References Section and Analog Outputs Section ..... 40
Added Equivalent DAC Output and Transfer Function and Figure 63 ..... 40
Deleted Mu Control Operation Section, Search Mode Section, and Figure 89 ..... 41
Moved Figure 64 ..... 41
Added Peak DAC Output Power Capability Section and Figure 65 ..... 41

Deleted Figure 90, Figure 91, Track Mode Section, Mu Delay and Phase Readback Section, Operating the Mu Controller Manually Section, and Calculating Mu Delay Line Step Size Section ..... 42
Added Output Stage Configuration Section and Figure 66 to Figure 70 ..... 42
Added Nonideal Spectral Artifacts Section, Figure 71, and Table 30 ..... 43
Deleted Operation in Master Mode, Figure 93, and Figure 94 ..... 44
Added Lab Evaluation of the AD9739 Section, Power Dissipation and Supply Domains Section, and Figure 72 to Figure 74 ..... 44
Deleted Figure 95, Operation in Slave Mode Section, and Data Receiver Operation in Auto Mode Section ..... 45
Changes to Recommended Start-Up Sequence Section ..... 45
Added Figure 75 ..... 45
Deleted Figure 97, Data Receiver Operation in Manual Mode Section, Calculating the DCI Delay Line Step Size Section, and Maximum Allowable Data Timing Skew/Jitter Section ..... 46
Added Table 31 ..... 46
Deleted Optimizing the Clock Common-Mode Voltage Section, Figure 99, Analog Control Registers Section, Mirror Roll-Off Frequency Control Section, and Figure 101 ..... 47
Added Table 32 ..... 47
Deleted Figure 103, Figure 104, and Figure 106 ..... 48
Updated Outline Dimensions ..... 48
Deleted Figure 107 to Figure 109 ..... 49
Deleted Table 35 to Table 44 ..... 50

**7/2011—Rev 0 to Rev A**

Changes to Table 2, DAC CLOCK INPUT (DACCLK_P, DACCLK_N), Added DAC Clock Rate ..... 4
Changes to Table 3, Added Dynamic Performance Parameters ..... 5
Change to Ordering Guide ..... 53

**2/2009—Revision 0: Initial Version**

# SPECIFICATIONS

## DC SPECIFICATIONS

VDDA = VDD33 = 3.3 V, VDDC = VDD = 1.8 V, $I_{OUTFS}$ = 20 mA.

**Table 1.**

| Parameter | Min | Typ | Max | Unit |
|---|---|---|---|---|
| RESOLUTION | | 14 | | Bits |
| ACCURACY | | | | |
| Integral Nonlinearity (INL) | | ±1.3 | | LSB |
| Differential Nonlinearity (DNL) | | ±0.8 | | LSB |
| ANALOG OUTPUTS | | | | |
| Gain Error (with Internal Reference) | | 5.5 | | % |
| Full-Scale Output Current | 8.66 | 20.2 | 31.66 | mA |
| Output Compliance Range | −1.0 | | +1.0 | V |
| Common-Mode Output Resistance | | 10 | | MΩ |
| Differential Output Resistance | | 70 | | Ω |
| Output Capacitance | | 1 | | pF |
| DAC CLOCK INPUT (DACCLK_P, DACCLK_N) | | | | |
| Differential Peak-to-Peak Voltage | 1.2 | 1.6 | 2.0 | V |
| Common-Mode Voltage | | 900 | | mV |
| DAC Clock Rate | 0.8 | | 2.5 | GHz |
| TEMPERATURE DRIFT | | | | |
| Gain | | 60 | | ppm/°C |
| Reference Voltage | | 20 | | ppm/°C |
| REFERENCE | | | | |
| Internal Reference Voltage | 1.15 | 1.2 | 1.25 | V |
| Output Resistance | | 5 | | kΩ |
| ANALOG SUPPLY VOLTAGES | | | | |
| VDDA | 3.1 | 3.3 | 3.5 | V |
| VDDC | 1.70 | 1.8 | 1.90 | V |
| DIGITAL SUPPLY VOLTAGES | | | | |
| VDD33 | 3.10 | 3.3 | 3.5 | V |
| VDD | 1.70 | 1.8 | 1.90 | V |
| SUPPLY CURRENTS AND POWER DISSIPATION, 2.0 GSPS | | | | |
| $I_{VDDA}$ | | 37 | 38 | mA |
| $I_{VDDC}$ | | 159 | 166 | mA |
| $I_{VDD33}$ | | 34 | 37 | mA |
| $I_{VDD}$ | | 233 | 238 | mA |
| Power Dissipation | | 0.940 | 0.975 | W |
| Sleep Mode, $I_{VDDA}$ | | 2.5 | 2.75 | mA |
| Power-Down Mode (Register 0x01 = 0x33 and Register 0x02 = 0x80) | | | | |
| $I_{VDDA}$ | | 0.02 | | mA |
| $I_{VDDC}$ | | 3.8 | | mA |
| $I_{VDD33}$ | | 0.5 | | mA |
| $I_{VDD}$ | | 0.1 | | mA |
| SUPPLY CURRENTS AND POWER DISSIPATION, 2.5 GSPS | | | | |
| $I_{VDDA}$ | | 37 | | mA |
| $I_{VDDC}$ | | 223 | | mA |
| $I_{VDD33}$ | | 34 | | mA |
| $I_{VDD}$ | | 290 | | mA |
| Power Dissipation | | 1.16 | | W |

## LVDS DIGITAL SPECIFICATIONS

VDDA = VDD33 = 3.3 V, VDDC = VDD = 1.8 V, $I_{OUTFS}$ = 20 mA. LVDS drivers and receivers are compliant to the IEEE Standard 1596.3-1996 reduced range link, unless otherwise noted.

**Table 2.**

| Parameter | Min | Typ | Max | Unit |
|---|---|---|---|---|
| LVDS DATA INPUTS (DB0[13:0], DB1[13:0])[1] | | | | |
| Input Common-Mode Voltage Range, $V_{COM}$ | 825 | | 1575 | mV |
| Logic High Differential Input Threshold, $V_{IH\_DTH}$ | 175 | 400 | | mV |
| Logic Low Differential Input Threshold, $V_{IL\_DTH}$ | −175 | −400 | | mV |
| Receiver Differential Input Impedance, $R_{IN}$ | 80 | | 120 | Ω |
| Input Capacitance | | 1.2 | | pF |
| LVDS Input Rate | 1250 | | | MSPS |
| LVDS Minimum Data Valid Period, $t_{VALID}$ (See Figure 41) | | | 344 | ps |
| LVDS CLOCK INPUT (DCI and SYNC_IN)[2] | | | | |
| Input Common-Mode Voltage Range, $V_{COM}$ | 825 | | 1575 | mV |
| Logic High Differential Input Threshold, $V_{IH\_DTH}$ | 175 | 400 | | mV |
| Logic Low Differential Input Threshold, $V_{IL\_DTH}$ | −175 | −400 | | mV |
| Receiver Differential Input Impedance, $R_{IN}$ | 80 | | 120 | Ω |
| Input Capacitance | | 1.2 | | pF |
| Maximum Clock Rate | 625 | | | MHz |
| LVDS CLOCK OUTPUT (DCO and SYNC_OUT)[3] | | | | |
| Output Voltage High (x_P or x_N) | | | 1375 | mV |
| Output Voltage Low (x_P or x_N) | 1025 | | | mV |
| Output Differential Voltage, $\|V_{OD}\|$ | 150 | 200 | 250 | mV |
| Output Offset Voltage, $V_{OS}$ | 1150 | | 1250 | mV |
| Output Impedance, Single-Ended, $R_O$ | 80 | 100 | 120 | Ω |
| $R_O$ Single-Ended Mismatch | | | 10 | % |
| Maximum Clock Rate | 625 | | | MHz |

[1] DB0[x]P, DB0[x]N, DB1[x]P, and DB1[x]N pins.
[2] DCI_P and DCI_N pins, as well as SYNC_IN_P and SYNC_IN_N pins.
[3] DCO_P and DCO_N pins, as well as SYNC_OUT_P/SYNC_OUT_N pins with 100 Ω differential termination.

## SERIAL PORT SPECIFICATIONS

VDDA = VDD33 = 3.3 V, VDDC = VDD = 1.8 V.

**Table 3.**

| Parameter | Min | Typ | Max | Unit |
|---|---|---|---|---|
| WRITE OPERATION (See Figure 36) | | | | |
| SCLK Clock Rate, $f_{SCLK}$ (or /$t_{SCLK}$) | | | 20 | MHz |
| SCLK Clock High, $t_{HI}$ | 18 | | | ns |
| SCLK Clock Low, $t_{LOW}$ | 18 | | | ns |
| SDIO to SCLK Setup Time, $t_{DS}$ | 2 | | | ns |
| SCLK to SDIO Hold Time, $t_{DH}$ | 1 | | | ns |
| $\overline{CS}$ to SCLK Setup Time, $t_S$ | 3 | | | ns |
| SCLK to $\overline{CS}$ Hold Time, $t_H$ | 2 | | | ns |
| READ OPERATION (See Figure 37 and Figure 38) | | | | |
| SCLK Clock Rate, $f_{SCLK}$ (or /$t_{SCLK}$) | | | 20 | MHz |
| SCLK Clock High, $t_{HI}$ | 18 | | | ns |
| SCLK Clock Low, $t_{LOW}$ | 18 | | | ns |
| SDIO to SCLK Setup Time, $t_{DS}$ | 2 | | | ns |
| SCLK to SDIO Hold Time, $t_{DH}$ | 1 | | | ns |
| $\overline{CS}$ to SCLK Setup Time, $t_S$ | 3 | | | ns |
| SCLK to SDIO (or SDO) Data Valid Time, $t_{DV}$ | | | 15 | ns |
| $\overline{CS}$ to SDIO (or SDO) Output Valid to High-Z, $t_{EZ}$ | | 2 | | ns |
| INPUTS (SDIO, SCLK, $\overline{CS}$) | | | | |
| Voltage in High, $V_{IH}$ | 2.0 | 3.3 | | V |
| Voltage in Low, $V_{IL}$ | | 0 | 0.8 | V |
| Current in High, $I_{IH}$ | −10 | | +10 | µA |
| Current in Low, $I_{IL}$ | −10 | | +10 | µA |
| OUTPUT (SDIO) | | | | |
| Voltage Out High, $V_{OH}$ | 2.4 | | 3.5 | V |
| Voltage Out Low, $V_{OL}$ | 0 | | 0.4 | V |
| Current Out High, $I_{OH}$ | | 4 | | mA |
| Current Out Low, $I_{OL}$ | | 4 | | mA |

## AC SPECIFICATIONS

VDDA = VDD33 = 3.3 V, VDDC = VDD = 1.8 V, $I_{OUTFS}$ = 20 mA, $f_{DAC}$ = 2400 MSPS.

**Table 4.**

| Parameter | Min | Typ | Max | Unit |
|---|---|---|---|---|
| DYNAMIC PERFORMANCE | | | | |
| DAC Clock Rate | 800 | | 2500 | MSPS |
| Adjusted DAC Update Rate[1] | 800 | | 2500 | MSPS |
| Output Settling Time ($t_{st}$) to 0.1% | | 13 | | ns |
| SPURIOUS-FREE DYNAMIC RANGE (SFDR) | | | | |
| $f_{OUT}$ = 100 MHz | | 69.5 | | dBc |
| $f_{OUT}$ = 350 MHz | | 58.5 | | dBc |
| $f_{OUT}$ = 550 MHz | | 54 | | dBc |
| $f_{OUT}$ = 950 MHz | | 60 | | dBc |
| TWO-TONE INTERMODULATION DISTORTION (IMD), $f_{OUT2} = f_{OUT1} + 1.25$ MHz | | | | |
| $f_{OUT}$ = 100 MHz | | 94 | | dBc |
| $f_{OUT}$ = 350 MHz | | 78 | | dBc |
| $f_{OUT}$ = 550 MHz | | 72 | | dBc |
| $f_{OUT}$ = 950 MHz | | 68 | | dBc |
| NOISE SPECTRAL DENSITY (NSD), 0 dBFS SINGLE TONE | | | | |
| $f_{OUT}$ = 100 MHz | | −166 | | dBm/Hz |
| $f_{OUT}$ = 350 MHz | | −161 | | dBm/Hz |
| $f_{OUT}$ = 550 MHz | | −160 | | dBm/Hz |
| $f_{OUT}$ = 850 MHz | | −160 | | dBm/Hz |
| WCDMA ACLR (SINGLE CARRIER), ADJACENT/ALTERNATE ADJACENT CHANNEL | | | | |
| $f_{DAC}$ = 2457.6 MSPS $f_{OUT}$ = 350 MHz | | 80/80 | | dBc |
| $f_{DAC}$ = 2457.6 MSPS, $f_{OUT}$ = 950 MHz | | 78/79 | | dBc |
| $f_{DAC}$ = 2457.6 MSPS, $f_{OUT}$ = 1700 MHz (Mix Mode) | | 74/74 | | dBc |
| $f_{DAC}$ = 2457.6 MSPS, $f_{OUT}$ = 2100 MHz (Mix Mode) | | 69/72 | | dBc |

[1] Adjusted DAC updated rate is calculated as $f_{DAC}$ divided by the minimum required interpolation factor. For the AD9739, the minimum interpolation factor is 1. Thus, with $f_{DAC}$ = 2500 MSPS, $f_{DAC}$ adjusted = 2500 MSPS.

# ABSOLUTE MAXIMUM RATINGS

**Table 5.**

| Parameter | With Respect To | Rating |
|---|---|---|
| VDDA | VSSA | −0.3 V to +3.6 V |
| VDD33 | VSS | −0.3 V to +3.6 V |
| VDD | VSS | −0.3 V to +1.98 V |
| VDDC | VSSC | −0.3 V to +1.98 V |
| VSSA | VSS | −0.3 V to +0.3 V |
| VSSA | VSSC | −0.3 V to +0.3 V |
| VSS | VSSC | −0.3 V to +0.3 V |
| DACCLK_P, DACCLK_N | VSSC | −0.3 V to VDDC + 0.18 V |
| DCI, DCO, SYNC_IN, SYNC_OUT | VSS | −0.3 V to VDD33 + 0.3 V |
| LVDS Data Inputs | VSS | −0.3 V to VDD33 + 0.3 V |
| IOUTP, IOUTN | VSSA | −1.0 V to VDDA + 0.3 V |
| I120, VREF | VSSA | −0.3 V to VDDA + 0.3 V |
| IRQ, $\overline{CS}$, SCLK, SDO, SDIO, RESET | VSS | −0.3 V to VDD33 + 0.3 V |
| Junction Temperature | | 150°C |
| Storage Temperature | | −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

$\theta_{JA}$ is specified for the worst-case conditions, that is, a device soldered in a circuit board for surface-mount packages.

**Table 6. Thermal Resistance**

| Package Type | $\theta_{JA}$ | $\theta_{JC}$ | Unit |
|---|---|---|---|
| 160-Ball CSP_BGA | 31.2 | 7.0 | °C/W[1] |

[1] With no airflow movement.

## 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 CONFIGURATIONS AND FUNCTION DESCRIPTIONS

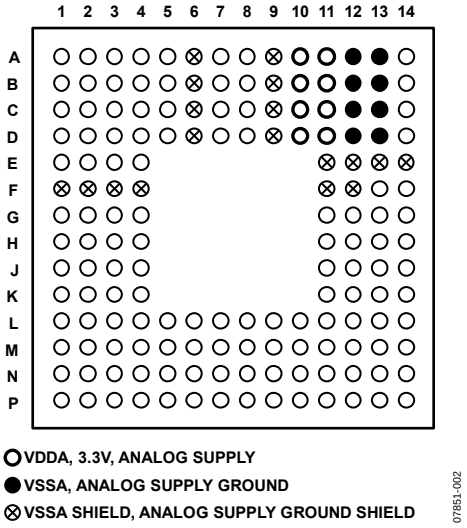



*Figure 2. Analog Supply Pins (Top View)*

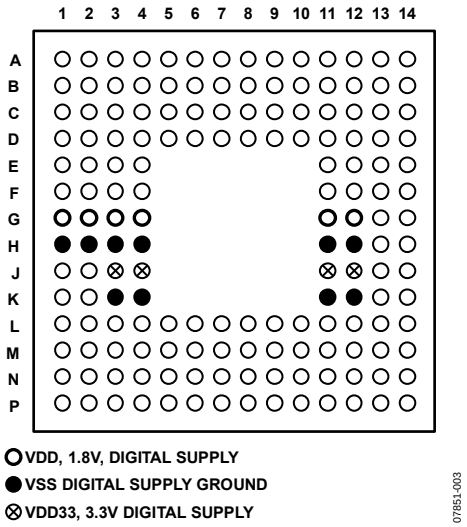



*Figure 3. Digital Supply Pins (Top View)*

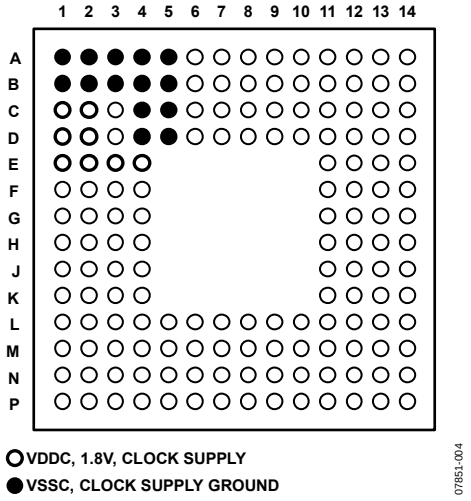



*Figure 4. Digital LVDS Clock Supply Pins (Top View)*

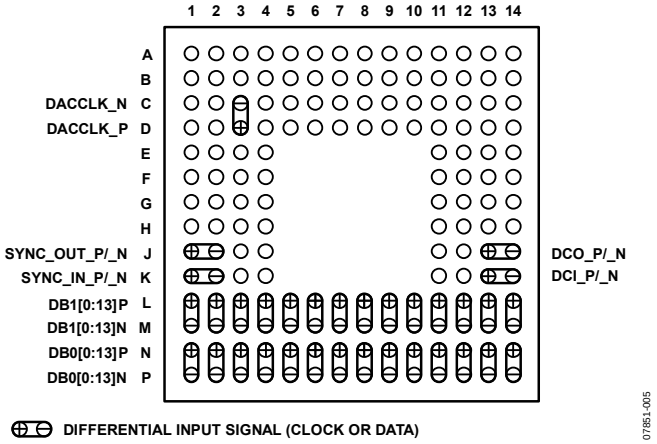



*Figure 5. Digital LVDS Input, Clock I/O (Top View)*

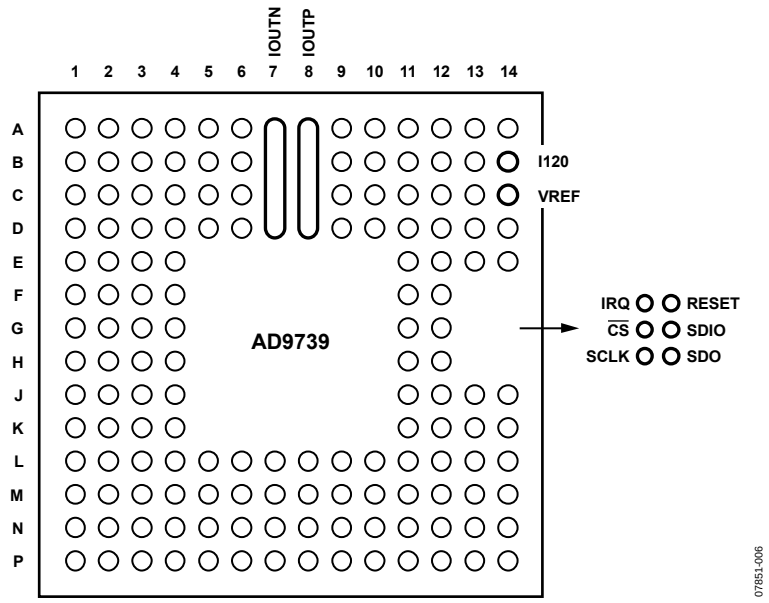



*Figure 6. Analog I/O and SPI Control Pins (Top View)*

**Table 7. AD9739 Pin Function Descriptions**

| Pin No. | Mnemonic | Description |
|---|---|---|
| C1, C2, D1, D2, E1, E2, E3, E4 | VDDC | 1.8 V Clock Supply Input. |
| A1, A2, A3, A4, A5, B1, B2, B3, B4, B5, C4, C5, D4, D5 | VSSC | Clock Supply Return. |
| A10, A11, B10, B11, C10, C11, D10, D11 | VDDA | 3.3 V Analog Supply Input. |
| A12, A13, B12, B13, C12, C13, D12, D13, | VSSA | Analog Supply Return. |
| A6, A9, B6, B9, C6, C9, D6, D9, F1, F2, F3, F4, E11, E12, E13, E14, F11, F12 | VSSA Shield | Analog Supply Return Shield. Tie to VSSA at the DAC. |
| A14 | NC | No Connect. Do not connect to this pin. |
| A7, B7, C7, D7 | IOUTN | DAC Negative Current Output Source. |
| A8, B8, C8, D8 | IOUTP | DAC Positive Current Output Source. |
| B14 | I120 | Nominal 1.2 V Reference. Tie to analog ground via a 10 kΩ resistor to generate a 120 µA reference current. |
| C14 | VREF | Voltage Reference Input/Output. Decouple to VSSA with a 1 nF capacitor. |
| D14 | NC | Factory Test Pin. Do not connect to this pin. |
| C3, D3 | DACCLK_N/DACCLK_P | Negative/Positive DAC Clock Input (DACCLK). |
| F13 | IRQ | Interrupt Request Open Drain Output. Active high. Pull up to VDD33 with a 10 kΩ resistor. |
| F14 | RESET | Reset Input. Active high. Tie to VSS if unused. |
| G13 | $\overline{CS}$ | Serial Port Enable Input. |
| G14 | SDIO | Serial Port Data Input/Output. |
| H13 | SCLK | Serial Port Clock Input. |
| H14 | SDO | Serial Port Data Output. |
| J3, J4, J11, J12 | VDD33 | 3.3 V Digital Supply Input. |
| G1, G2, G3, G4, G11, G12 | VDD | 1.8 V Digital Supply. Input. |
| H1, H2, H3, H4, H11, H12, K3, K4, K11, K12 | VSS | Digital Supply Return. |
| J1, J2 | SYNC_OUT_P/SYNC_OUT_N | Positive/Negative SYNC Output (SYNC_OUT) |
| K1, K2 | SYNC_IN_P/SYNC_IN_N | Positive/Negative SYNC Input (SYNC_IN) |
| J13, J14 | DCO_P/DCO_N | Positive/Negative Data Clock Output (DCO). |
| K13, K14 | DCI_P/DCI_N | Positive/Negative Data Clock Input (DCI). |
| L1, M1 | DB1[0]P/DB1[0]N | Port 1 Positive/Negative Data Input Bit 0. |
| L2, M2 | DB1[1]P/DB1[1]N | Port 1 Positive/Negative Data Input Bit 1. |
| L3, M3 | DB1[2]P/DB1[2]N | Port 1 Positive/Negative Data Input Bit 2. |

| Pin No. | Mnemonic | Description |
|---|---|---|
| L4, M4 | DB1[3]P/DB1[3]N | Port 1 Positive/Negative Data Input Bit 3. |
| L5, M5 | DB1[4]P/DB1[4]N | Port 1 Positive/Negative Data Input Bit 4. |
| L6, M6 | DB1[5]P/DB1[5]N | Port 1 Positive/Negative Data Input Bit 5. |
| L7, M7 | DB1[6]P/DB1[6]N | Port 1 Positive/Negative Data Input Bit 6. |
| L8, M8 | DB1[7]P/DB1[7]N | Port 1 Positive/Negative Data Input Bit 7. |
| L9, M9 | DB1[8]P/DB1[8]N | Port 1 Positive/Negative Data Input Bit 8. |
| L10, M10 | DB1[9]P/DB1[9]N | Port 1 Positive/Negative Data Input Bit 9. |
| L11, M11 | DB1[10]P/DB1[10]N | Port 1 Positive/Negative Data Input Bit 10. |
| L12, M12 | DB1[11]P/DB1[11]N | Port 1 Positive/Negative Data Input Bit 11. |
| L13, M13 | DB1[12]P/DB1[12]N | Port 1 Positive/Negative Data Input Bit 12. |
| L14, M14 | DB1[13]P/DB1[13]N | Port 1 Positive/Negative Data Input Bit 13. |
| N1, P1 | DB0[0]P/DB0[0]N | Port 0 Positive/Negative Data Input Bit 0. |
| N2, P2 | DB0[1]P/DB0[1]N | Port 0 Positive/Negative Data Input Bit 1. |
| N3, P3 | DB0[2]P/DB0[2]N | Port 0 Positive/Negative Data Input Bit 2. |
| N4, P4 | DB0[3]P/DB0[3]N | Port 0 Positive/Negative Data Input Bit 3. |
| N5, P5 | DB0[4]P/DB0[4]N | Port 0 Positive/Negative Data Input Bit 4. |
| N6, P6 | DB0[5]P/DB0[5]N | Port 0 Positive/Negative Data Input Bit 5. |
| N7, P7 | DB0[6]P/DB0[6]N | Port 0 Positive/Negative Data Input Bit 6. |
| N8, P8 | DB0[7]P/DB0[7]N | Port 0 Positive/Negative Data Input Bit 7. |
| N9, P9 | DB0[8]P/DB0[8]N | Port 0 Positive/Negative Data Input Bit 8. |
| N10, P10 | DB0[9]P/DB0[9]N | Port 0 Positive/Negative Data Input Bit 9. |
| N11, P11 | DB0[10]P/DB0[10]N | Port 0 Positive/Negative Data Input Bit 10. |
| N12, P12 | DB0[11]P/DB0[11]N | Port 0 Positive/Negative Data Input Bit 11. |
| N13, P13 | DB0[12]P/DB0[12]N | Port 0 Positive/Negative Data Input Bit 12. |
| N14, P14 | DB0[13]P/DB0[13]N | Port 0 Positive/Negative Data Input Bit 13. |

# TYPICAL PERFORMANCE CHARACTERISTICS

## AC (NORMAL MODE)

$I_{OUTFS}$ = 20 mA, nominal supplies, 25°C, unless otherwise noted.

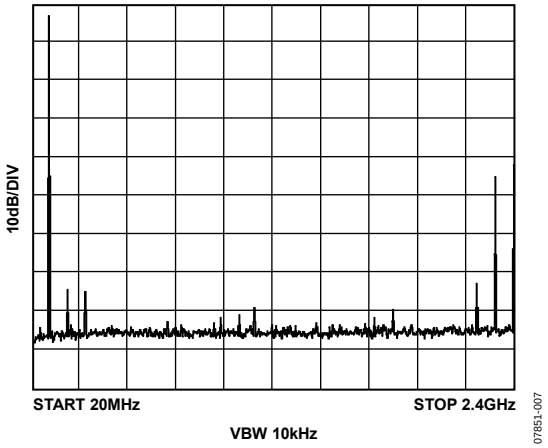



Figure 7. Single-Tone Spectrum at $f_{OUT}$ = 91 MHz, $f_{DAC}$ = 2.4 GSPS

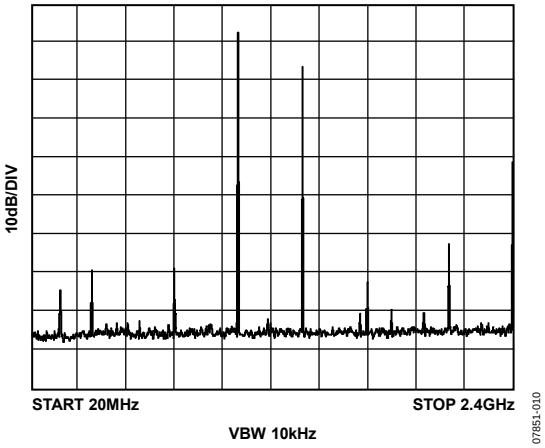



Figure 10. Single-Tone Spectrum at $f_{OUT}$ = 1091 MHz, $f_{DAC}$ = 2.4 GSPS

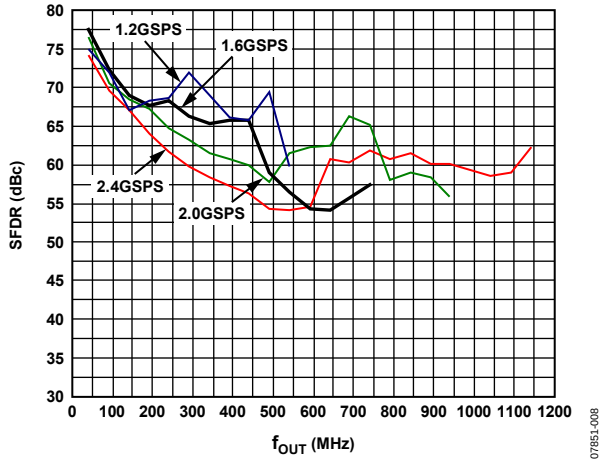



Figure 8. SFDR vs. $f_{OUT}$ over $f_{DAC}$

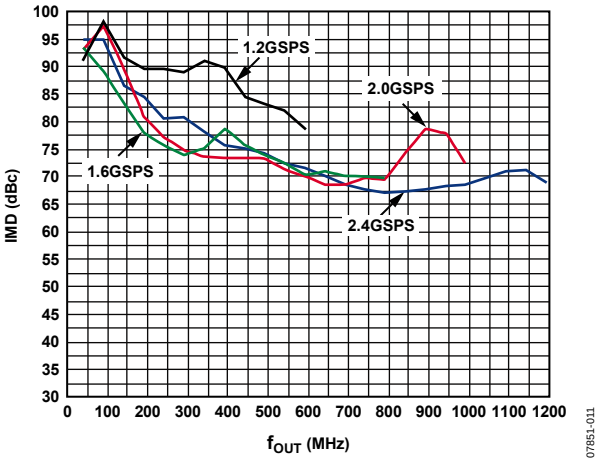



Figure 11. IMD vs. $f_{OUT}$ over $f_{DAC}$

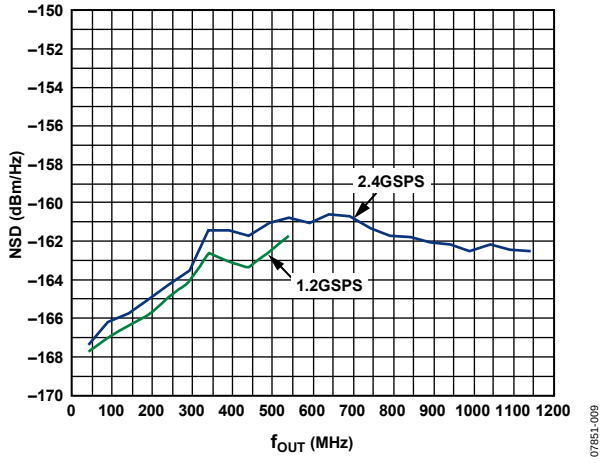



Figure 9. Single-Tone NSD over $f_{OUT}$

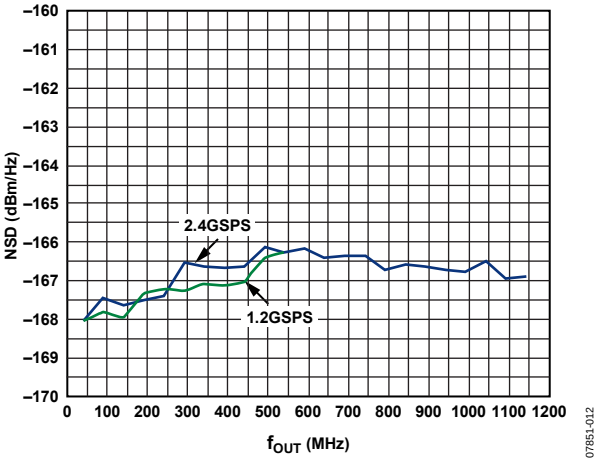



Figure 12. Eight-Tone NSD over $f_{OUT}$

$f_{DAC}$ = 2 GSPS, $I_{OUTFS}$ = 20 mA, nominal supplies, 25°C, unless otherwise noted.

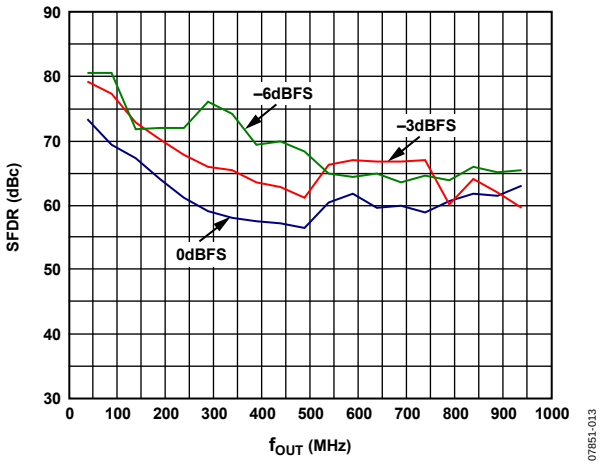



Figure 13. SFDR vs. $f_{OUT}$ over Digital Full Scale

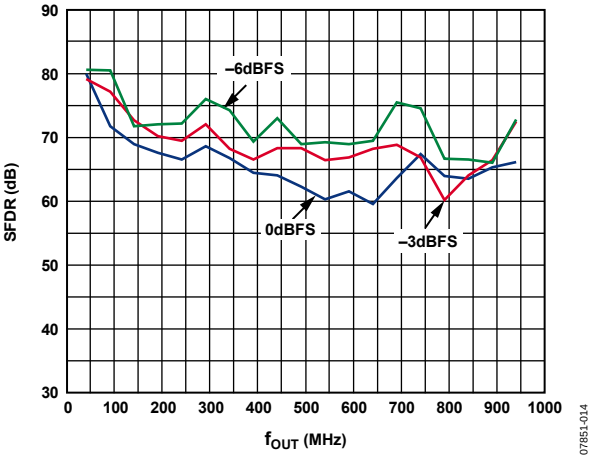



Figure 14. SFDR for Second Harmonic over $f_{OUT}$ vs. Digital Full Scale

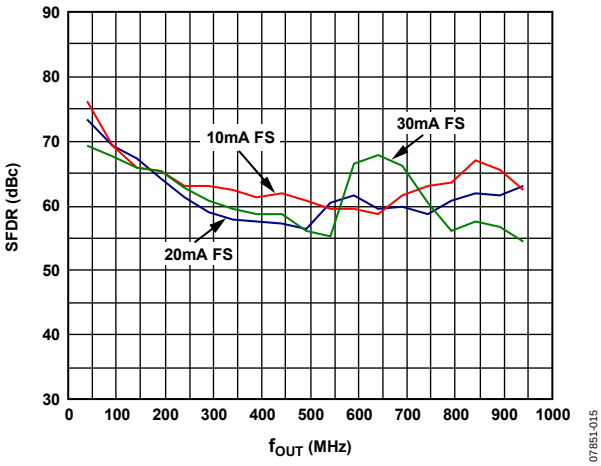



Figure 15. SFDR vs. $f_{OUT}$ over DAC $I_{OUTFS}$

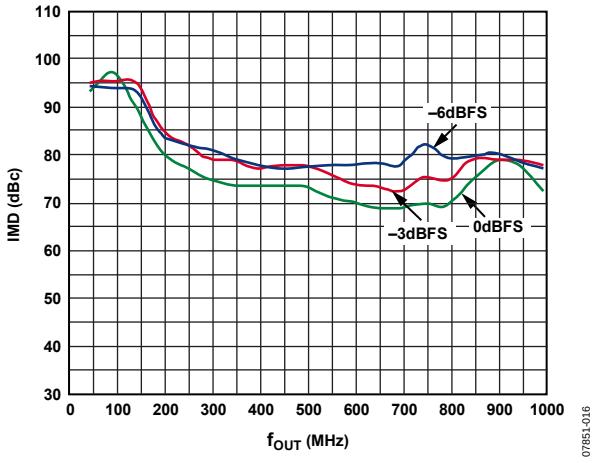



Figure 16. IMD vs. $f_{OUT}$ over Digital Full Scale

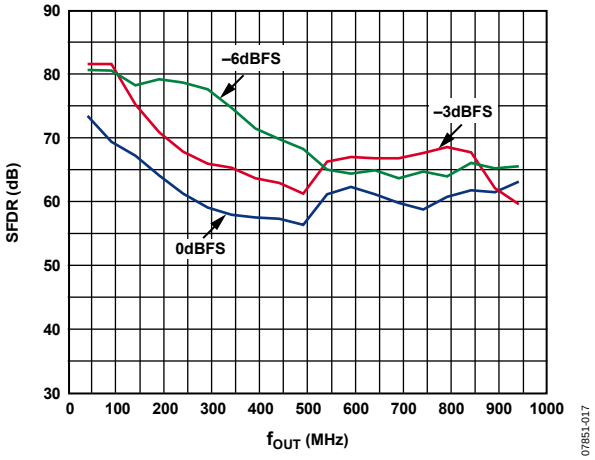



Figure 17. SFDR for Third Harmonic over $f_{OUT}$ vs. Digital Full Scale

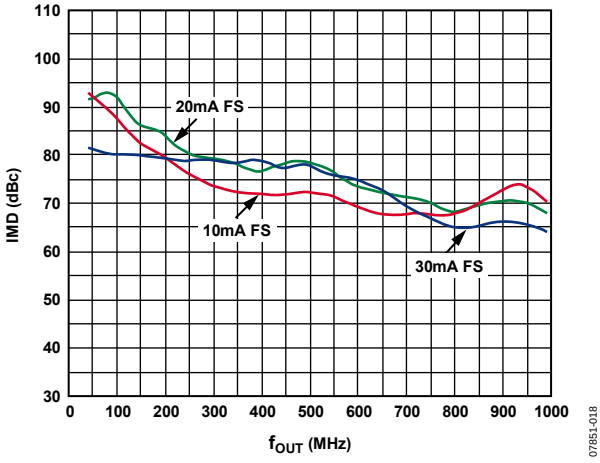



Figure 18. IMD vs. $f_{OUT}$ over DAC $I_{OUTFS}$

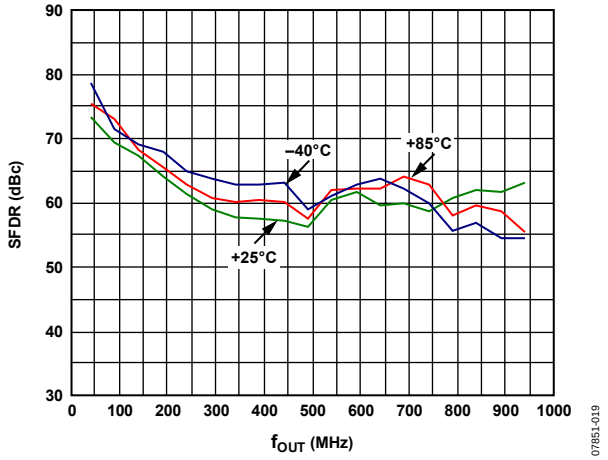



Figure 19. SFDR vs. $f_{OUT}$ over Temperature

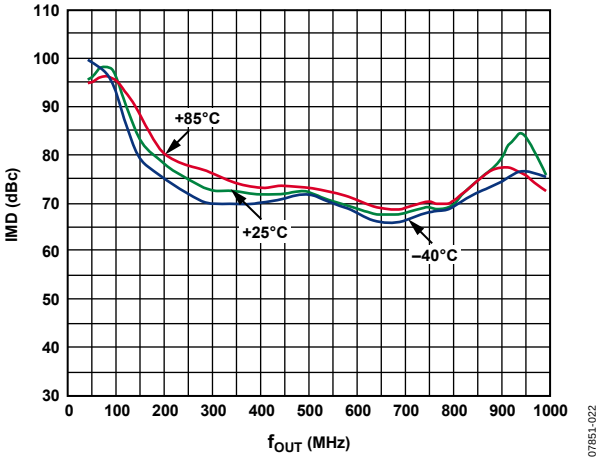



Figure 22. IMD vs. $f_{OUT}$ over Temperature

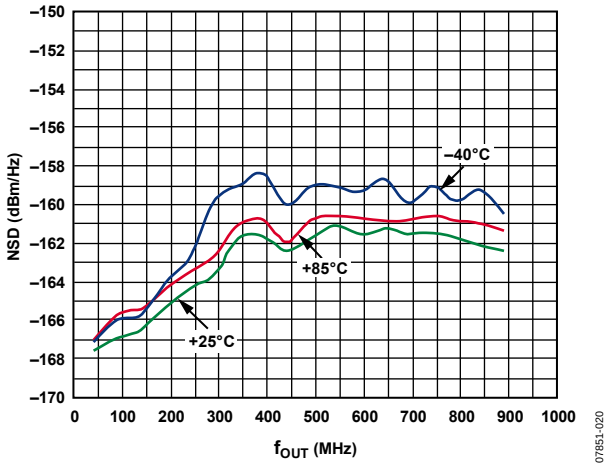



Figure 20. Single-Tone NSD vs. $f_{OUT}$ over Temperature

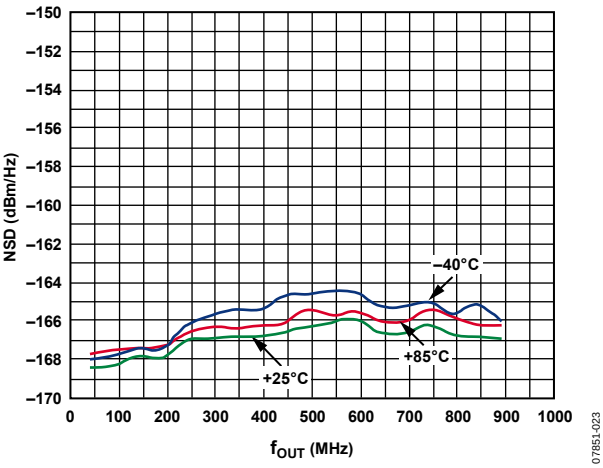



Figure 23. Eight-Tone NSD vs. $f_{OUT}$ over Temperature

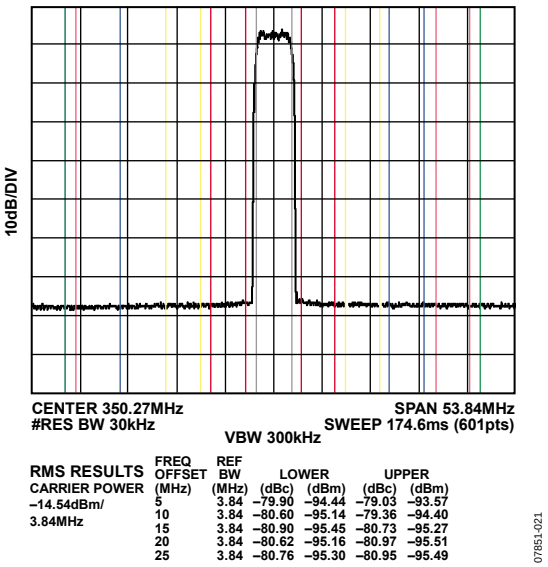



Figure 21. Single-Carrier WCDMA at 350 MHz, $f_{DAC}$ = 2457.6 MSPS

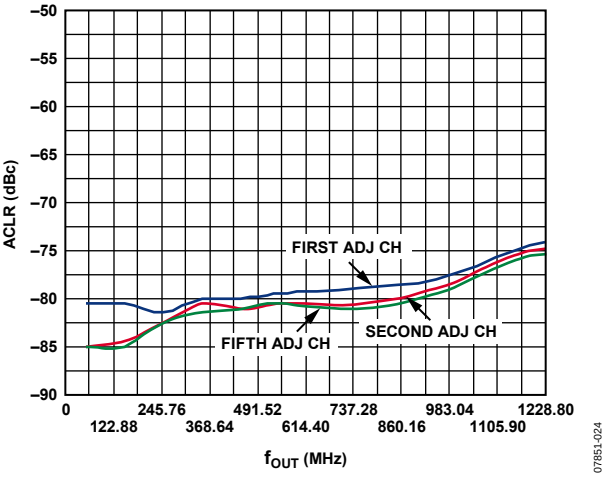



Figure 24. Four-Carrier WCDMA at 350 MHz, $f_{DAC}$ = 2457.6 MSPS

## AC (MIX MODE)

$f_{DAC}$ = 2.4 GSPS, $I_{OUTFS}$ = 20 mA, nominal supplies, 25°C, unless otherwise noted.

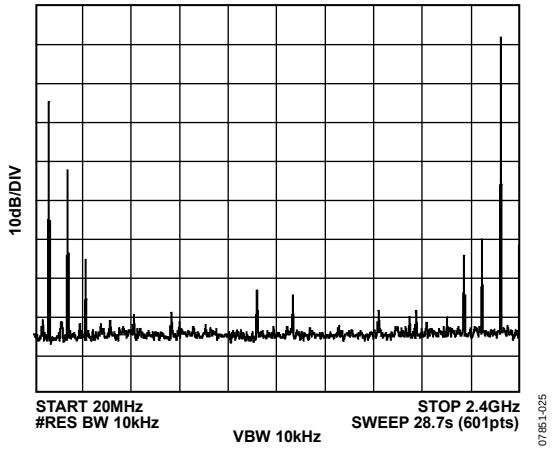



Figure 25. Single-Tone Spectrum at $f_{OUT}$ = 2.31 GHz, $f_{DAC}$ = 2.4 GSPS

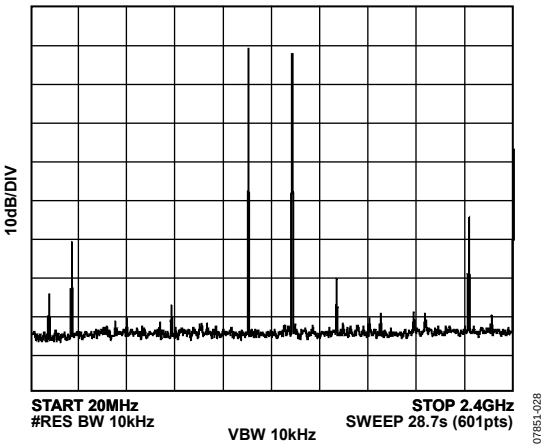



Figure 28. Single-Tone Spectrum in Mix Mode at $f_{OUT}$ = 1.31 GHz, $f_{DAC}$ = 2.4 GSPS

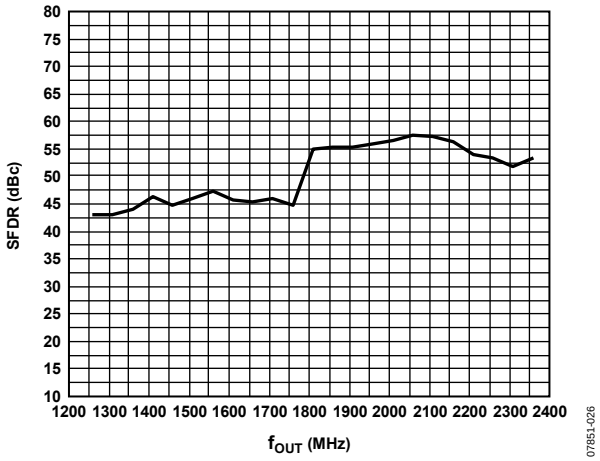



Figure 26. SFDR in Mix Mode vs. $f_{OUT}$ at 2.4 GSPS

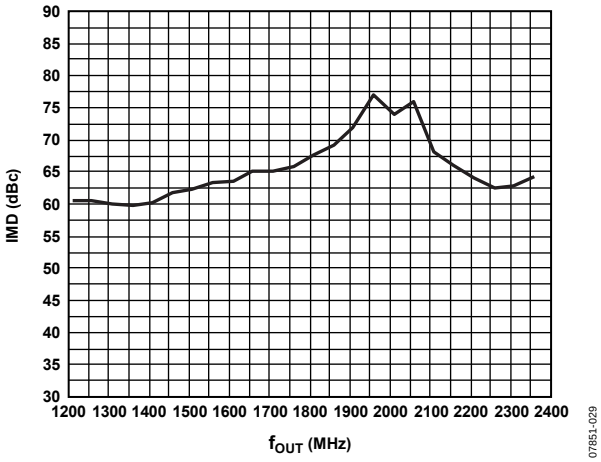



Figure 29. IMD in Mix Mode vs. $f_{OUT}$ at 2.4 GSPS

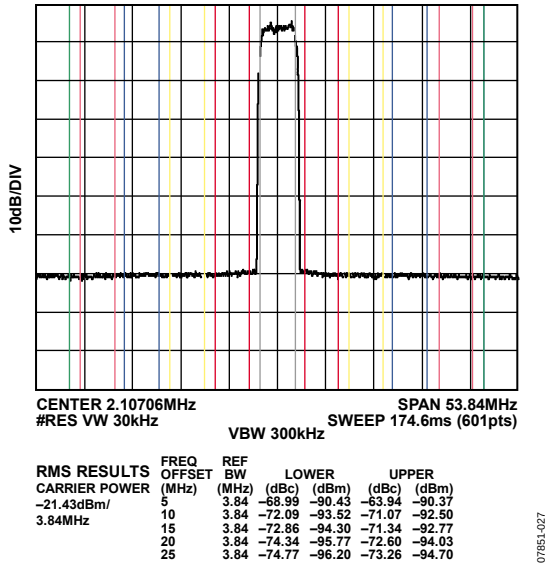



Figure 27. Typical Single-Carrier WCDMA ACLR Performance at 2.1 GHz, $f_{DAC}$ = 2457.6 MSPS (Second Nyquist Zone)

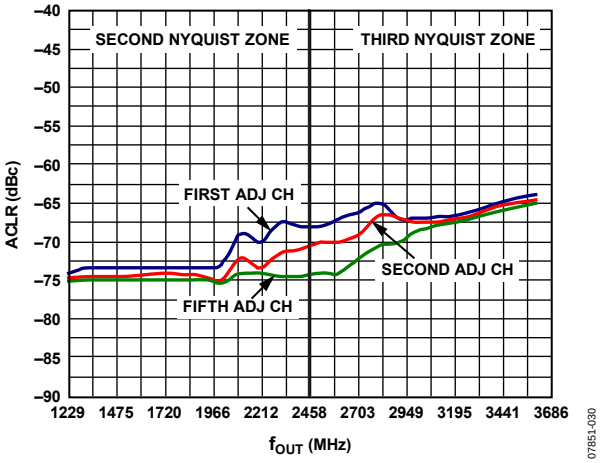
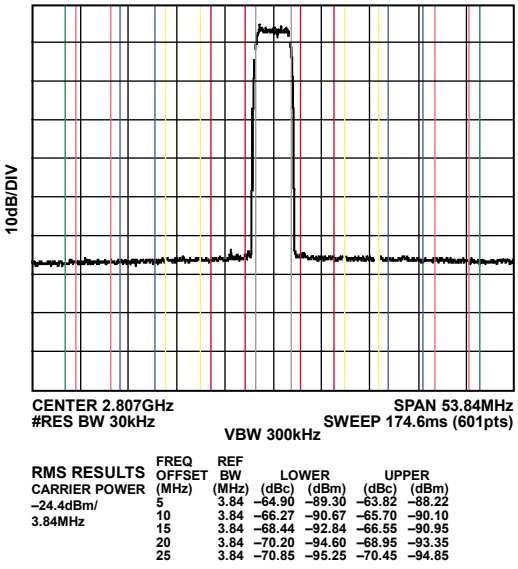
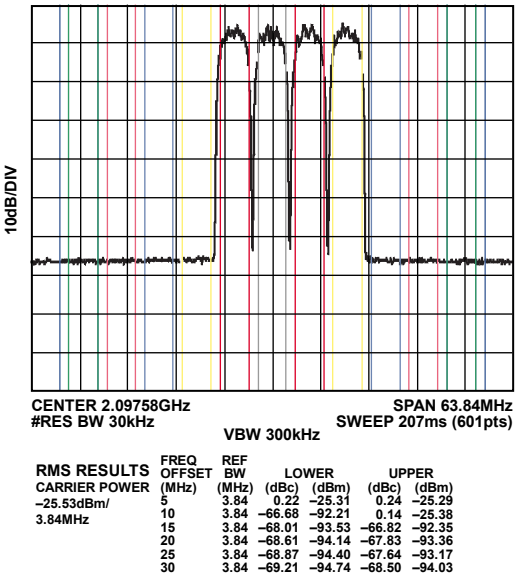



Figure 30. Single-Carrier WCDMA ACLR vs. $f_{OUT}$ at 2457.6 MSPS



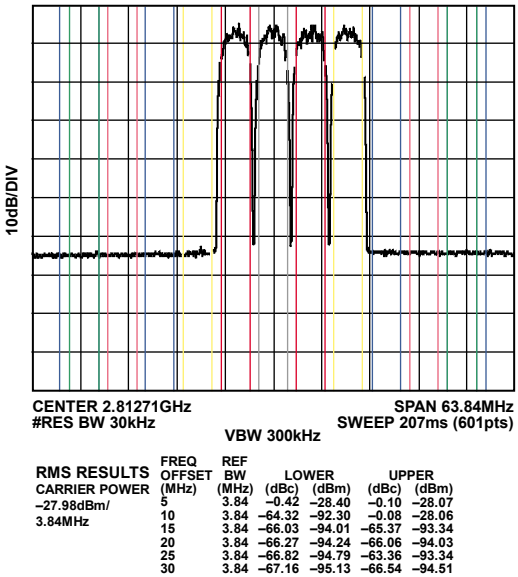

| RMS RESULTS CARRIER POWER | FREQ OFFSET (MHz) | REF BW (MHz) | LOWER (dBc) | LOWER (dBm) | UPPER (dBc) | UPPER (dBm) |
|---|---|---|---|---|---|---|
| –24.4dBm/ 3.84MHz | 5 | 3.84 | –64.90 | –89.30 | –63.82 | –88.22 |
| | 10 | 3.84 | –66.27 | –90.67 | –65.70 | –90.10 |
| | 15 | 3.84 | –68.44 | –92.84 | –66.55 | –90.95 |
| | 20 | 3.84 | –70.20 | –94.60 | –68.95 | –93.35 |
| | 25 | 3.84 | –70.85 | –95.25 | –70.45 | –94.85 |

*Figure 31. Typical Single-Carrier WCDMA ACLR Performance at 2.8 GHz, $f_{DAC}$ = 2457.6 MSPS (Third Nyquist Zone)*




| RMS RESULTS CARRIER POWER | FREQ OFFSET (MHz) | REF BW (MHz) | LOWER (dBc) | LOWER (dBm) | UPPER (dBc) | UPPER (dBm) |
|---|---|---|---|---|---|---|
| –25.53dBm/ 3.84MHz | 5 | 3.84 | 0.22 | –25.31 | 0.24 | –25.29 |
| | 10 | 3.84 | –66.68 | –92.21 | 0.14 | –25.38 |
| | 15 | 3.84 | –68.01 | –93.53 | –66.82 | –92.35 |
| | 20 | 3.84 | –68.61 | –94.14 | –67.83 | –93.36 |
| | 25 | 3.84 | –68.87 | –94.40 | –67.64 | –93.17 |
| | 30 | 3.84 | –69.21 | –94.74 | –68.50 | –94.03 |

*Figure 32. Typical Four-Carrier WCDMA ACLR Performance at 2.1 GHz, $f_{DAC}$ = 2457.6 MSPS (Second Nyquist Zone)*




| RMS RESULTS CARRIER POWER | FREQ OFFSET (MHz) | REF BW (MHz) | LOWER (dBc) | LOWER (dBm) | UPPER (dBc) | UPPER (dBm) |
|---|---|---|---|---|---|---|
| –27.98dBm/ 3.84MHz | 5 | 3.84 | –0.42 | –28.40 | –0.10 | –28.07 |
| | 10 | 3.84 | –64.32 | –92.30 | –0.08 | –28.06 |
| | 15 | 3.84 | –66.03 | –94.01 | –65.37 | –93.34 |
| | 20 | 3.84 | –66.27 | –94.24 | –66.06 | –94.03 |
| | 25 | 3.84 | –66.82 | –94.79 | –63.36 | –93.34 |
| | 30 | 3.84 | –67.16 | –95.13 | –66.54 | –94.51 |

*Figure 33. Typical Four-Carrier WCDMA ACLR Performance at 2.8 GHz, $f_{DAC}$ = 2457.6 MSPS (Third Nyquist Zone)*

# TERMINOLOGY

**Linearity Error (Integral Nonlinearity or INL)**
The maximum deviation of the actual analog output from the ideal output, determined by a straight line drawn from 0 to full scale.

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

**Monotonicity**
A DAC is monotonic if the output either increases or remains constant as the digital input increases.

**Offset Error**
The deviation of the output current from the ideal of 0 is called the offset error. For IOUTP, 0 mA output is expected when the inputs are all 0s. For IOUTN, 0 mA output is expected when all inputs are set to 1.

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

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

**Temperature Drift**
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 °C. For reference drift, the drift is reported in ppm per °C.

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

**Spurious-Free Dynamic Range (SFDR)**
The difference, in decibels (dB), between the rms amplitude of the output signal and the peak spurious signal over the specified bandwidth.

**Total Harmonic Distortion (THD)**
The ratio of the rms sum of the first six harmonic components to the rms value of the measured input signal. It is expressed as a percentage or in decibels (dB).

**Noise Spectral Density (NSD)**
NSD is the converter noise power per unit of bandwidth. This is usually specified in dBm/Hz in the presence of a 0 dBm full-scale signal.

**Adjacent Channel Leakage Ratio (ACLR)**
The adjacent channel leakage (power) ratio is a ratio, in dBc, of the measured power within a channel relative to its adjacent channels.

**Modulation Error Ratio (MER)**
Modulated signals create a discrete set of output values referred to as a constellation. Each symbol creates an output signal corresponding to one point on the constellation. MER is a measure of the discrepancy between the average output symbol magnitude and the rms error magnitude of the individual symbol.

**Intermodulation Distortion (IMD)**
IMD is the result of two or more signals at different frequencies mixing together. Many products are created according to the formula, aF1 ± bF2, where a and b are integer values.

# SERIAL PORT INTERFACE (SPI) REGISTER

## SPI REGISTER MAP DESCRIPTION

The AD9739 contains a set of programmable registers described in Table 10 that are used to configure and monitor various internal parameters. Note the following points when programming the AD9739 SPI registers:

- Registers pertaining to similar functions are grouped together and assigned adjacent addresses.
- Bits that are undefined within a register should be assigned a 0 when writing to that register.
- Registers that are undefined should not be written to.
- A hardware or software reset is recommended upon power-up to place SPI registers in a known state.
- A SPI initialization routine is required as part of the boot process. See Table 31 and Table 32 for example procedures.

### *Reset*

Issuing a hardware or software reset places the AD9739 SPI registers in a known state. All SPI registers (excluding 0x00) are set to their default states as described in Table 10 upon issuing a reset. After issuing a reset, the SPI initialization process need only write to registers that are required for the boot process as well as any other register settings that must be modified, depending on the target application.

Although the AD9739 does feature an internal power-on-reset (POR), it is still recommended that a software or hardware reset be implemented shortly after power-up. The internal reset signal is derived from a logical OR operation from the internal POR signal, the RESET pin, and the software reset state. A software reset can be issued via the reset bit (Register 0x00, Bit 5) by toggling the bit high then low. Note that, because the MSB/LSB format may still be unknown upon initial power-up (that is, internal POR is unsuccessful), it is also recommended that the bit settings for Bits[7:5] be mirrored onto Bits[2:0] for the instruction cycle that issues a software reset. A hardware reset can be issued from a host or external supervisory IC by applying a high pulse with a minimum width of 40 ns to the RESET pin (that is, Pin F14). RESET should be tied to VSS if unused.

**Table 8. SPI Registers Pertaining to SPI Options**

| Address (Hex) | Bit | Description |
|---|---|---|
| 0x00 | 7 | Enable 3-wire SPI |
| | 6 | Enable SPI LSB first |
| | 5 | Software reset |

## SPI OPERATION

The serial port of the AD9739 shown in Figure 34 has a 3- or 4-wire SPI capability, allowing read/write access to all registers that configure the device's internal parameters. It provides a flexible, synchronous serial communications port, allowing easy interface to many industry-standard microcontrollers and microprocessors. The 3.3 V serial I/O is compatible with most synchronous transfer formats, including the Motorola® SPI and the Intel® SSR protocols.

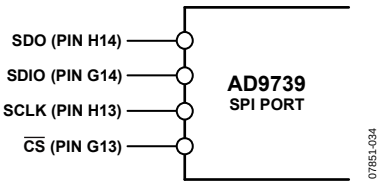



*Figure 34. AD9739 SPI Port*

The default 4-wire SPI interface consists of a clock (SCLK), serial port enable ($\overline{CS}$), serial data input (SDIO), and serial data output (SDO). The inputs to SCLK, $\overline{CS}$, and SDIO contain a Schmitt trigger with a nominal hysteresis of 0.4 V centered about VDD33/2. The maximum frequency for SCLK is 20 MHz. The SDO pin is active only during the transmission of data and remains three-stated at any other time.

A 3-wire SPI interface can be enabled by setting the SDIO_DIR bit (Register 0x00, Bit 7). This causes the SDIO pin to become bidirectional such that output data only appears on the SDIO pin during a read operation. The SDO pin remains three-stated in a 3-wire SPI interface.

### *Instruction Header Information*

| MSB | | | | | | | LSB |
|---|---|---|---|---|---|---|---|
| 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 |
| R/$\overline{W}$ | A6 | A5 | A4 | A3 | A2 | A1 | A0 |

An 8-bit instruction header must accompany each read and write operation. The MSB is a R/$\overline{W}$ indicator bit with logic high indicating a read operation. The remaining seven bits specify the address bits to be accessed during the data transfer portion. The eight data bits immediately follow the instruction header for both read and write operations. For write operations, registers change immediately upon writing to the last bit of each transfer byte. $\overline{CS}$ can be raised after each sequence of eight bits (except the last byte) to stall the bus. The serial transfer resumes when $\overline{CS}$ is lowered. Stalling on nonbyte boundaries resets the SPI.

The AD9739 serial port can support both most significant bit (MSB) first and least significant bit (LSB) first data formats. Figure 35 illustrates how the serial port words are formed for the MSB first and LSB first modes. The bit order is controlled by the SDIO_DIR bit (Register 0x00, Bit 7). The default value is 0, MSB first. When the LSB first bit is set high, the serial port interprets both instruction and data bytes LSB first.

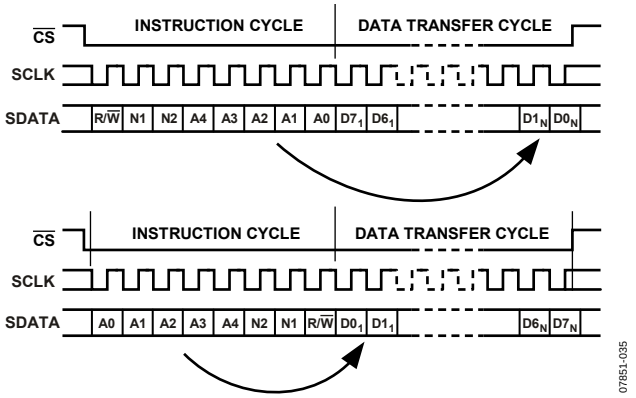



*Figure 35. SPI Timing, MSB First (Upper) and LSB First (Lower)*

Figure 36 illustrates the timing requirements for a write operation to the SPI port. After the serial port enable ($\overline{CS}$) signal goes low, data (SDIO) pertaining to the instruction header is read on the rising edges of the clock (SCLK). To initiate a write operation, the read/not-write bit is set low. After the instruction header is read, the eight data bits pertaining to the specified register are shifted into the SDIO pin on the rising edge of the next eight clock cycles.

Figure 37 illustrates the timing for a 3-wire read operation to the SPI port. After $\overline{CS}$ goes low, data (SDIO) pertaining to the instruction header is read on the rising edges of SCLK. A read operation occurs if the read/not-write indicator is set high. After the address bits of the instruction header are read, the eight data bits pertaining to the specified register are shifted out of the SDIO pin on the falling edges of the next eight clock cycles.

Figure 38 illustrates the timing for a 4-wire read operation to the SPI port. The timing is similar to the 3-wire read operation with the exception that data appears at the SDO pin only, while the SDIO pin remains at high impedance throughout the operation. The SDO pin is an active output only during the data transfer phase and remains three-stated at all other times.

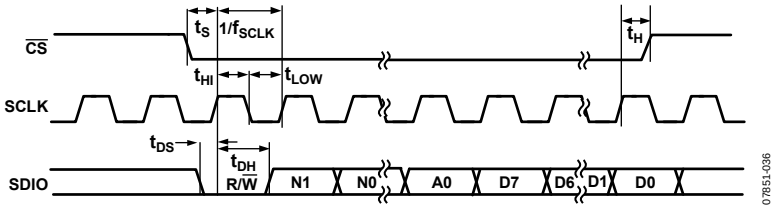



*Figure 36. SPI Write Operation Timing*

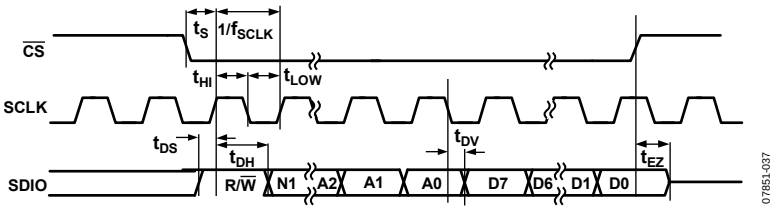



*Figure 37. SPI 3-Wire Read Operation Timing*

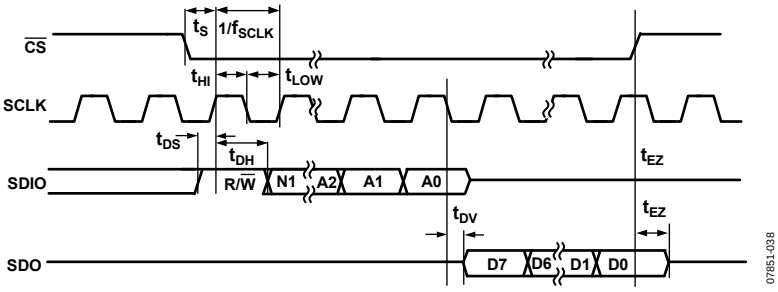



*Figure 38. SPI 4-Wire Read Operation Timing*

# SPI REGISTER MAP

**Table 9. Full Register Map (N/A = Not Applicable)**

| Name | Hex Addr | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Default |
|---|---|---|---|---|---|---|---|---|---|---|
| Mode | 00 | SDIO_DIR | LSB/MSB | Reset | N/A | N/A | N/A | N/A | N/A | 0x00 |
| Power-Down | 01 | N/A | N/A | LVDS_ DRVR_PD | LVDS_ RCVR_PD | N/A | N/A | CLK_ RCVR_PD | DAC_ BIAS_PD | 0x00 |
| CNT_ CLK_DIS | 02 | N/A | N/A | N/A | N/A | CLKGEN_PD | N/A | REC_CNT_ CLK | MU_CNT_ CLK | 0x03 |
| IRQ_EN | 03 | N/A | N/A | SYNC_ LST_EN | SYNC_ LCK_EN | MU_LST_EN | MU_LCK_EN | RCV_ LST_EN | RCV_ LCK_EN | 0x00 |
| IRQ_REQ | 04 | N/A | N/A | SYNC_ LST_IRQ | SYNC_ LCK_IRQ | MU_LST_ IRQ | MU_LCK_ IRQ | RCVLST_ IRQ | RCVLCK_ IRQ | 0x00 |
| RSVD | 05 | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A |
| FSC_1 | 06 | FSC[7] | FSC[6] | FSC[5] | FSC[4] | FSC[3] | FSC[2] | FSC[1] | FSC[0] | 0x00 |
| FSC_2 | 07 | Sleep | N/A | N/A | N/A | N/A | N/A | FSC[9] | FSC[8] | 0x02 |
| DEC_ CNT | 08 | N/A | N/A | N/A | N/A | N/A | N/A | DAC_DEC[1] | DAC_DEC[0] | 0x00 |
| RSVD | 09 | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A |
| LVDS_ CNT | 0A | N/A | N/A | N/A | N/A | HNDOFF_ CHK_RST | N/A | LVDS_ Bias[1] | LVDS_ Bias[0] | 0x00 |
| DIG_ STAT | 0B | HNDOFF_ Fall[3] | HNDOFF_ Fall[2] | HNDOFF_ Fall[1] | HNDOFF_ Fall[0] | HNDOFF_ Rise[3] | HNDOFF_ Rise[2] | HNDOFF_ Rise[1] | HNDOFF_ Rise[0] | RNDM |
| LVDS_ STAT1 | 0C | SUP/HLD_ Edge1 | N/A | DCI_ PHS3 | DCI_ PHS1 | DCI_PRE_ PH2 | DCI_PRE_ PH0 | DCI_PST_ PH2 | DCI_PST_ PH0 | RNDM |
| LVDS_ STAT2 | 0D | SUP/HLD_ SYNC | SUP/HLD_ Edge0 | SYNC_ SAMP1 | SYNC_ SAMP0 | LVDS1_HI | LVDS1_LO | LVDS0_HI | LVDS0_LO | RNDM/0 |
| RSVD | 0E | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A |
| RSVD | 0F | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A |
| LVDS_ REC_ CNT1 | 10 | SYNC_ FLG_RST | SYNC_ LOOP_ON | SYNC_ MST/SLV | SYNC_ CNT_ENA | N/A | RCVR_ FLG_RST | RCVR_ LOOP_ON | RCVR_ CNT_ENA | 0x42 |
| LVDS_ REC_ CNT2 | 11 | SMP_DEL[1] | SMP_ DEL[0] | FINE_ DEL_ MID[3] | FINE_ DEL_ MID[2] | FINE_DEL_ MID[1] | FINE_DEL_ MID[0] | RCVR_ GAIN[1] | RCVR_ GAIN[0] | 0xDD |
| LVDS_ REC_ CNT3 | 12 | SMP_DEL[9] | SMP_ DEL[8] | SMP_ DEL[7] | SMP_ DEL[6] | SMP_ DEL[5] | SMP_ DEL[4] | SMP_ DEL[3] | SMP_ DEL[2] | 0x29 |
| LVDS_ REC_ CNT4 | 13 | DCI_DEL[3] | DCI_ DEL[2] | DCI_ DEL[1] | DCI_ DEL[0] | FINE_DEL_ SKW[3] | FINE_DEL_ SKW[2] | FINE_DEL_ SKW[1] | FINE_DEL_ SKW[0] | 0x71 |
| LVDS_ REC_ CNT5 | 14 | CLKDIVPH[1] | CLKDIVPH[0] | DCI_ DEL[9] | DCI_ DEL[8] | DCI_ DEL[7] | DCI_ DEL[6] | DCI_ DEL[5] | DCI_ DEL[4] | 0x0A |
| LVDS_ REC_ CNT6 | 15 | SYNC_ GAIN[1] | SYNC_ GAIN[0] | SYNCOUT_ PH[1] | SYNCOUT_ PH[0] | LCKTHR[3] | LCKTHR[2] | LCKTHR[1] | LCKTHR[0] | 0x42 |
| LVDS_ REC_ CNT7 | 16 | N/A | SYNCO_ DEL[6] | SYNCO_ DEL[5] | SYNCO_ DEL[4] | SYNCO_ DEL[3] | SYNCO_ DEL[2] | SYNCO_ DEL[1] | SYNCO_ DEL[0] | 0x00 |
| LVDS_ REC_ CNT8 | 17 | SYNCSH_ DEL[0] | N/A | N/A | N/A | N/A | N/A | N/A | N/A | 0x00 |
| LVDS_ REC_ CNT9 | 18 | SYNCSH_ DEL[8] | SYNCSH_ DEL[7] | SYNCSH_ DEL[6] | SYNCSH_ DEL[5] | SYNCSH_ DEL[4] | SYNCSH_ DEL[3] | SYNCSH_ DEL[2] | SYNCSH_ DEL[1] | 0x00 |
| LVDS_ REC_ STAT1 | 19 | SMP_DEL[1] | SMP_DEL[0] | N/A | N/A | SMP_ FINE_ DEL[3] | SMP_ FINE_ DEL[2] | SMP_ FINE_ DEL[1] | SMP_ FINE_ DEL[0] | 0xC7 |
| LVDS_ REC_ STAT2 | 1A | SMP_DEL[9] | SMP_ DEL[8] | SMP_ DEL[7] | SMP_ DEL[6] | SMP_ DEL[5] | SMP_ DEL[4] | SMP_ DEL[3] | SMP_ DEL[2] | 0x29 |

| Name | Hex Addr | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Default |
|---|---|---|---|---|---|---|---|---|---|---|
| LVDS_ REC_ STAT3 | 1B | DCI_DEL[1] | DCI_DEL[0] | N/A | N/A | SYNCOUT PH[1] | SYNCOUT PH[0] | CLKDIV PH[1] | CLKDIV PH[0] | 0xC0 |
| LVDS_ REC_ STAT4 | 1C | DCI_DEL[9] | DCI_ DEL[8] | DCI_ DEL[7] | DCI_ DEL[6] | DCI_ DEL[5] | DCI_ DEL[4] | DCI_ DEL[3] | DCI_ DEL[2] | 0x29 |
| LVDS_ REC_ STAT5 | 1D | FINE_DEL_ PST[3] | FINE_DEL_ PST[2] | FINE_DEL_ PST[1] | FINE_DEL_ PST[0] | FINE_DEL_ PRE[3] | FINE_DEL_ PRE[2] | FINE_DEL_ PRE[1] | FINE_DEL_ PRE[0] | 0x86 |
| LVDS_ REC_ STAT6 | 1E | N/A | SYNCO_ DEL[6] | SYNCO_ DEL[5] | SYNCO_ DEL[4] | SYNCO_ DEL[3] | SYNCO_ DEL[2] | SYNCO_ DEL[1] | SYNCO_ DEL[0] | 0x00 |
| LVDS_ REC_ STAT7 | 1F | SYNCSH_ DEL[0] | N/A | N/A | N/A | N/A | N/A | N/A | N/A | 0x00 |
| LVDS_ REC_ STAT8 | 20 | SYNCSH_ DEL[8] | SYNCSH_ DEL[7] | SYNCSH_ DEL[6] | SYNCSH_ DEL[5] | SYNCSH_ DEL[4] | SYNCSH_ DEL[3] | SYNCSH_ DEL[2] | SYNCSH_ DEL[1] | 0x00 |
| LVDS_ REC_ STAT9 | 21 | SYNC_ TRK_ON | SYNC_ INIT_ON | SYNC_ LST_LCK | SYNC_LCK | RCVR_ TRK_ON | RCVR_ FE_ON | RCVR_LST | RCVR_LCK | 0x00 |
| CROSS_ CNT1 | 22 | N/A | N/A | N/A | DIR_P | CLKP_ OFFSET[3] | CLKP_ OFFSET[2] | CLKP_ OFFSET[1] | CLKP_ OFFSET[0] | 0x00 |
| CROSS_ CNT2 | 23 | N/A | N/A | N/A | DIR_N | CLKN_ OFFSET[3] | CLKN_ OFFSET[2] | CLKN_ OFFSET[1] | CLKN_ OFFSET[0] | 0x00 |
| PHS_ DET | 24 | N/A | N/A | CMP_BST | PHS_DET AUTO_EN | Bias[3] | Bias[2] | Bias[1] | Bias[0] | 0x00 |
| MU_ DUTY | 25 | MU_ DUTYAUTO_EN | POS/NEG | ADJ[5] | ADJ[4] | ADJ[3] | ADJ[2] | ADJ[1] | ADJ[0] | 0x00 |
| MU_ CNT1 | 26 | N/A | Slope | Mode[1] | Mode[0] | Read | Gain[1] | Gain[0] | Enable | 0x42 |
| MU_ CNT2 | 27 | MUDEL[0] | SRCH_MODE [1] | SRCH_MODE [0] | SET_PHS[4] | SET_PHS[3] | SET_PHS[2] | SET_PHS[1] | SETPHS[0] | 0x40 |
| MU_ CNT3 | 28 | MUDEL[8] | MUDEL[7] | MUDEL[6] | MUDEL[5] | MUDEL[4] | MUDEL[3] | MUDEL[2] | MUDEL[1] | 0x00 |
| MU_ CNT4 | 29 | SEARCH_TOL | Retry | CONTRST | Guard[4] | Guard[3] | Guard[2] | Guard[1] | Guard[0] | 0x0B |
| MU_ STAT1 | 2A | N/A | N/A | N/A | N/A | N/A | N/A | MU_LOST | MU_LKD | 0x00 |
| RSVD | 2B | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A |
| RSVD | 2C | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A |
| ANA_ CNT1 | 32 | HDRM[7] | HDRM[6] | HDRM[5] | HDRM[4] | HDRM[3] | HDRM[2] | HDRM[1] | HDRM[0] | 0xCA |
| ANA_ CNT2 | 33 | N/A | N/A | N/A | N/A | N/A | N/A | MSEL[1] | MSEL[0] | 0x03 |
| RSVD | 34 | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A |
| PART ID | 35 | ID[7] | ID[6] | ID[5] | ID[4] | ID[3] | ID[2] | ID[1] | ID[0] | 0x20 |

## SPI PORT CONFIGURATION AND SOFTWARE RESET

**Table 10. SPI Port Configuration and Software Reset Register**

| Address (Hex) | Name | Bit | R/W | Default Setting | Comments |
|---|---|---|---|---|---|
| 0x00 | SDIO_DIR | 7 | R/W | 0 | 0 = 4-wire SPI, 1 = 3-wire SPI. |
| | LSB/MSB | 6 | R/W | 0 | 0 = MSB first, 1 = LSB first. |
| | Reset | 5 | R/W | 0 | Software reset is recommended before modification of other SPI registers from the default setting. Setting the bit to 1 causes all registers (except 0x00) to be set to the default setting. Setting the bit to 0 corresponds to the inactive state, allowing the user to modify registers from the default setting. |

## POWER-DOWN LVDS INTERFACE AND TXDAC®

**Table 11. Power-Down LVDS Interface and TxDAC Register**

| Address (Hex) | Name | Bit | R/W | Default Setting | Comments |
|---|---|---|---|---|---|
| 0x01 | LVDS_DRVR_PD | 5 | R/W | 0 | Power-down of the LVDS drivers/receivers and TxDAC. 0 = enable, 1 = disable. |
| | LVDS_RCVR_PD | 4 | R/W | 0 | |
| | CLK_RCVR_PD | 1 | R/W | 0 | |
| | DAC_BIAS_PD | 0 | R/W | 0 | |

## CONTROLLER CLOCK DISABLE

**Table 12. Controller Clock Disable Register**

| Address (Hex) | Name | Bit | R/W | Default Setting | Comments |
|---|---|---|---|---|---|
| 0x02 | CLKGEN_PD | 3 | R/W | 0 | Internal CLK distribution enable: 0 = enable, 1 = disable. |
| | REC_CNT_CLK | 1 | R/W | 1 | LVDS receiver and Mu controller clock disable. 0 = disable, 1 = enable. |
| | MU_CNT_CLK | 0 | R/W | 1 | |

## INTERRUPT REQUEST (IRQ) ENABLE/STATUS

**Table 13. Interrupt Request (IRQ) Enable/Status Register**

| Address (Hex) | Name | Bit | R/W | Default Setting | Comments |
|---|---|---|---|---|---|
| 0x03 | SYNC_LST_EN | 5 | W | 0 | This register enables the sync, mu, and LVDS Rx controllers to update their corresponding IRQ status bits in Register 0x04, which defines whether the controller is locked (LCK) or unlocked (LST). 0 = disable (resets the status bit). 1 = enable. |
| | SYNC_LCK_EN | 4 | W | 0 | |
| | MU_LST_EN | 3 | W | 0 | |
| | MU_LCK_EN | 2 | W | 0 | |
| | RCV_LST_EN | 1 | W | 0 | |
| | RCV_LCK_EN | 0 | W | 0 | |
| 0x04 | SYNC_LST_IRQ | 5 | R | 0 | This register indicates the status of the controllers. For LCK_IQR bits: 0 = lost locked, 1 = locked. For LST_IQR bits: 0 = not lost locked, 1 = unlocked. Note that, if the controller IRQ is serviced, the relevant bits in Register 0x03 should be reset by writing 0, followed by another write of 1 to enable. |
| | SYNC_LCK_IRQ | 4 | R | 0 | |
| | MU_LST_IRQ | 3 | R | 0 | |
| | MU_LCK_IRQ | 2 | R | 0 | |
| | RCV_LST_IRQ | 1 | R | 0 | |
| | RCV_LCK_IRQ | 0 | R | 0 | |

## TxDAC FULL-SCALE CURRENT SETTING ($I_{OUTFS}$) AND SLEEP

**Table 14. TxDAC Full-Scale Current Setting ($I_{OUTFS}$) and Sleep Register**

| Address (Hex) | Name | Bit | R/W | Default Setting | Comments |
|---|---|---|---|---|---|
| 0x06 | FSC_1 | [7:0] | R/W | 0x00 | Sets the TxDAC $I_{OUTFS}$ current between 8 mA and 31 mA (default = 20 mA). $I_{OUTFS}$ = 0.0226 × FSC[9:0] + 8.58, where FSC = 0 to 1023. |
| 0x07 | FSC_2 | [1:0] | R/W | 0x02 | |
| | Sleep | 7 | R/W | | 0 = enable DAC output, 1 = disable DAC output (sleep). |

## TxDAC QUAD-SWITCH MODE OF OPERATION

**Table 15. TxDAC Quad-Switch Mode of Operation Register**

| Address (Hex) | Name | Bit | R/W | Default Setting | Comments |
|---|---|---|---|---|---|
| 0x08 | DAC-DEC | [1:0] | R/W | 0x00 | 0x00 = normal baseband mode.<br>0x01 = return-to-zero mode.<br>0x02 = mix mode. |

## DCI PHASE ALIGNMENT STATUS

**Table 16. DCI Phase Alignment Status Register**

| Address (Hex) | Name | Bit | R/W | Default Setting | Comments |
|---|---|---|---|---|---|
| 0x0C | DCI_PRE_PH0 | 2 | R | 0 | 0 = DCI rising edge is after the PRE delayed version of the Phase 0 sampling edge.<br>1 = DCI rising edge is before the PRE delayed version of the Phase 0 sampling edge. |
| | DCI_PST_PH0 | 0 | R | 0 | 0 = DCI rising edge is after the POST delayed version of the Phase 0 sampling edge.<br>1 = DCI rising edge is before the POST delayed version of the Phase 0 sampling edge. |

## SYNC_IN PHASE ALIGNMENT STATUS

**Table 17. SYNC_IN Phase Alignment Status Register**

| Address (Hex) | Name | Bit | R/W | Default Setting | Comments |
|---|---|---|---|---|---|
| 0x0D | SYNC_IN_PH90 | 5 | R | 0 | 0 = SYNCIN rising edge is after Phase 90 sampling edge.<br>1 = SYNCIN rising edge is before Phase 90 sampling edge. |
| | SYNC_IN_PH0 | 4 | R | 0 | 0 = SYNCIN rising edge is after Phase 0 sampling edge.<br>1 = SYNCIN rising edge is before Phase 0 sampling edge. |

## DATA RECEIVER CONTROLLER CONFIGURATION

**Table 18. Data Receiver Controller Configuration Register**

| Address (Hex) | Name | Bit | R/W | Default Setting | Comments |
|---|---|---|---|---|---|
| 0x10 | SYNC_FLG_RST | 7 | W | 0 | Sync controller flag reset. Write 1 followed by 0 to reset flags. |
| | SYNC_LOOP_ON | 6 | R/W | 1 | 0 = disable, 1 = enable. Enable for master only. When enabled, sync controller generates an IRQ when master falls out of lock and automatically begins search/track routine. |
| | SYNC_MST/SLV | 5 | R/W | 0 | Sync controller configuration. 0 = slave, 1 = master. |
| | SYNC_CNT_ENA | 4 | R/W | 0 | Sync controller enable. 0 = disable, 1 = enable |
| | RCVR_FLG_RST | 2 | W | 0 | Data receiver controller flag reset. Write 1 followed by 0 to reset flags. |
| | RCVR_LOOP_ON | 1 | R/W | 1 | 0 = disable, 1 = enable. When enabled, the data receiver controller generates an IRQ; it falls out of lock and automatically begins a search/track routine. |
| | RCVR_CNT_ENA | 0 | R/W | 0 | Data receiver controller enabled. 0 = disable, 1 = enable. |

## DATA RECEIVER CONTROLLER_DATA SAMPLE DELAY VALUE

**Table 19. Data Receiver Controller_Data Sample Delay Value Register**

| Address (Hex) | Name | Bit | R/W | Default Setting | Comments |
|---|---|---|---|---|---|
| 0x11 | SMP_DEL[1:0] | [7:6] | R/W | 0xDD | Controller enabled: the 10-bit value (with a maximum of 332) represents the start value for the delay line used by the state machine to sample data. Leave at the default setting of 167, which represents the midpoint of the delay line. Controller disabled: the value sets the actual value of the delay line. |
| 0x12 | SMP_DEL[9:2] | [7:0] | R/W | 0x29 | |

## DATA AND SYNC RECEIVER CONTROLLER_DCI DELAY VALUE/WINDOW AND PHASE ROTATION

**Table 20. Data and Sync Receiver Controller_DCI Delay Value/Window and Phase Rotation Register**

| Address (Hex) | Name | Bit | R/W | Default Setting | Comments |
|---|---|---|---|---|---|
| 0x13 | DCI_DEL[3:0] | [7:4] | R/W | 0111 | Refer to the DCI_DEL description in Register 0x14. |
| | FINE_DEL_SKEW | [3:0] | R/W | 0001 | A 4-bit value sets the difference (that is, window) for the DCI PRE and POST sampling clocks. Leave at the default value of 1 for a narrow window. |
| 0x14 | CLKDIVPH[1:0] | [7:6] | R/W | 00 | Relative phase of internal div-by-4 circuit. This feature allows phase rotation in 90° increments (that is, 1 count) to extend Rx controllers locking range for clock rates between 0.8 GSPS to 1.6 GSPS (only valid with sync controller disabled). |
| | DCI_DEL[9:4] | [5:0] | R/W | 001010 | Controller enabled: the 10-bit value (with a maximum of 332) represents the start value for the delay line used by the state machine to sample the DCI input. Leave at the default setting of 167, which represents the midpoint of the delay line. Controller disabled: the value sets the actual value of the delay line. |
| 0x15 | SYNC GAIN[1:0] | [7:6] | R/W | 00 | Sets the sync tracking gain (optimal value is 1). |
| | SYNCOUT_PH[1:0] | [5:4] | R/W | 00 | Readback of the present SYNC_OUT phase selection. |
| | LCKTHR[3:0] | [3:0] | R/W | 0000 | Sets the difference between the sample and DCI delays to lock (optimal value is 2). |
| 0x16 | SYNCO_DEL[6:0] | [6:0] | R/W | 0x00 | Sets the sync output delay value when the synch controller is disabled; otherwise, is the read status of the sync output delay value when sync is enabled. |
| 0x17 | SYNCSH_DEL[0] | [7] | R/W | 0x00 | Sets the sync setup and hold delay value when the synch controller is disabled; otherwise, is the read status of sync setup and hold value when sync is enabled. |
| 0x18 | SYNCSH_DEL[8:1] | [7:0] | R/W | 0x00 | Sets the sync setup and hold delay value when the synch controller is disabled; otherwise, is the read status of sync setup and hold value when sync is enabled. |

## DATA RECEIVER CONTROLLER_DELAY LINE STATUS AND SYNC CONTROLLER SYNC_OUT STATUS

**Table 21. Data Receiver Controller_Delay Line Status and Sync Controller SYNC_OUT Status Register**

| Address (Hex) | Name | Bit | R/W | Default Setting | Comments |
|---|---|---|---|---|---|
| 0x19 | SMP_DEL[1:0] | [7:6] | R | 00 | The actual value of the DCI and data delay lines determined by the data receiver controller (when enabled) after the state machine completes its search and enters track mode. Note that these values should be equal. SYNCOUT_PH provides phase status (0/90/180/270) of phase select mux, while CLKDIVPH provides phase status of data receiver controller (Register 0x14). |
| 0x1A | SMP_DEL[9:2] | [7:0] | R | 0x00 | |
| 0x1B | SYNCOUT_PH[1:0] | 3:2 | R | 00 | |
| | CLKDIV PH[1:0] | 1:0 | R | 00 | |
| | DCI_DEL[1:0] | [7:6] | R | 00 | |
| 0x1C | DCI_DEL[9:2] | [7:0] | R | 0x00 | |

## SYNC AND DATA RECEIVER CONTROLLER LOCK/TRACKING STATUS

**Table 22. Sync and Data Receiver Controller Lock/Tracking Status Register**

| Address (Hex) | Name | Bit | R/W | Default Setting | Comments |
|---|---|---|---|---|---|
| 0x21 | SYNC_TRK_ON | 7 | R | 0 | SYNC_TRK_ON and RCVR_TRK_ON:<br>0 = tracking not established.<br>1 = tracking established.<br>SYNC_LCK and RCVR_LCK:<br>0 = controller is not locked.<br>1 = controller is locked.<br>SYNC_LST and RCVR_LST:<br>0 = lock has not been lost.<br>1 = lock has been lost at some point. |
| | SYNC_LST | 5 | R | 0 | |
| | SYNC_LCK | 4 | R | 0 | |
| | RCVR_TRK_ON | 3 | R | 0 | |
| | RCVR_LST | 1 | R | 0 | |
| | RCVR_LCK | 0 | R | 0 | |

## CLK INPUT COMMON MODE

**Table 23. CLK Input Common Mode Register**

| Address (Hex) | Name | Bit | R/W | Default Setting | Comments |
|---|---|---|---|---|---|
| 0x22 | DIR_P | 4 | R/W | 0 | DIR_P and DIR_N:<br>0 = VCM at the DACCLK_P input decreases with the offset value.<br>1 = VCM at the DACCLK_P input increases with the offset value.<br>CLKx_OFFSET sets the magnitude of the offset for the DACCLK_P and DACCLK_N inputs. For optimum performance, set to 1111. |
| | CLKP_OFFSET[3:0] | [3:0] | R/W | 0000 | |
| 0x23 | DIR_N | 4 | R/W | 0 | |
| | CLKN_OFFSET[3:0] | [3:0] | R/W | 0000 | |

## MU CONTROLLER CONFIGURATION AND STATUS

**Table 24. Mu Controller Configuration and Status Register**

| Address (Hex) | Name | Bit | R/W | Default Setting | Comments |
|---|---|---|---|---|---|
| 0x24 | CMP_BST | 5 | R/W | 0 | Phase detector enable and boost bias bits. Note that both bits should always be set to 1 to enable these functions. |
| | PHS_DET AUTO_EN | 4 | R/W | 0 | |
| 0x25 | MU_DUTY AUTO_EN | 7 | R/W | 0 | Mu controller duty cycle enable. Note that this bit should always be set to 1 to enable. |
| 0x26 | Slope | 6 | R/W | 1 | Mu controller phase slope lock. 0 = negative slope, 1 = positive slope.<br>Refer to Table 28 for optimum setting. |
| | Mode[1:0] | [5:4] | R/W | 00 | Sets the mu controller mode of operation.<br>00 = search and track (recommended).<br>01 = search only.<br>10 = track. |
| | Read | 3 | R/W | 0 | Set to 1 to read the current value of the Mu delay line in. |
| | Gain[1:0] | [2:1] | R/W | 01 | Sets the mu controller tracking gain. Recommended to leave at the default 01 setting. |
| | Enable | 0 | R/W | 0 | 1 = enable the mu controller.<br>0 = disable the mu controller. |
| 0x27 | MUDEL[0] | 7 | R/W | 0 | The LSB of the 9-bit MUDEL setting. |
| | SRCH_MODE[1:0] | [6:5] | R/W | 0 | Sets the direction in which the mu controller searches (from its initial MUDEL setting) for the optimum mu delay line setting that corresponds to the desired phase/slope setting (that is, SET_PHS and slope ).<br>00 = down.<br>01 = up.<br>10 = down/up (recommended). |
| | SET_PHS[4:0] | [4:0] | R/W | 0 | Sets the target phase that the mu controller locks to with a maximum setting of 16.<br>Refer to Table 28 for optimum setting. |

| Address (Hex) | Name | Bit | R/W | Default Setting | Comments |
|---|---|---|---|---|---|
| 0x28 | MUDEL[8:1] | [7:0] | W | 0x00 | With enable (Bit 0, Register 0x26) set to 0, this 9-bit value represents the value that the mu delay is set to. Note that the maximum value is 432.<br>With enable set to 1, this value represents the mu delay value at which the controller begins its search. Setting this value to the delay line midpoint of 216 is recommended. |
| | | | R | 0x00 | When read (Bit 3, Register 0x26) is set to 1, the value read back is equal to the value written into the register when enable = 0 or the value that the mu controller locks to when enable = 1. |
| 0x29 | SEARCH_TOL | 7 | R/W | 0 | 0 = not exact (can find a phase within two values of the desired phase).<br>1 = finds the exact phase that is targeted (optimal setting). |
| | Retry | 6 | R/W | 0 | 0 = stop the search if the correct value is not found.<br>1 = retry the search if the correct value is not found. |
| | CONTRST | 5 | R/W | 0 | Controls whether the controller resets or continues when it does not find the desired phase.<br>0 = continue (optimal setting).<br>1 = reset. |
| | Guard[4:0] | 4 | R/W | 01011 | Sets a guard band from the beginning and end of the mu delay line which the mu controller does not enter into unless it does not find a valid phase outside the guard band (optimal value is Decimal 11 or 0x0B). |
| 0x2A | MU_LST | 1 | R | 0 | 0 = mu controller has not lost lock.<br>1 = mu controller has lost lock. |
| | MU_LKD | 0 | R | 0 | 0 = mu controller is not locked.<br>1= mu controller is locked. |

## PART ID

**Table 25. Part ID Register**

| Address (Hex) | Name | Bit | R/W | Default Setting | Comments |
|---|---|---|---|---|---|
| 0x35 | PART_ID | [7:0] | R | 0x20 | Part ID number. |

# THEORY OF OPERATION

Figure 39 shows a top-level functional diagram of the AD9739. A high performance TxDAC core delivers a signal dependent, differential current (nominal ±10 mA) to a balanced load referenced to ground. The frequency of the clock signal appearing at the AD9739 differential clock receiver, DACCLK, sets the TxDAC's update rate. This clock signal, which serves as the master clock, is routed directly to the TxDAC as well as to a clock distribution block that generates all critical internal and external clocks.

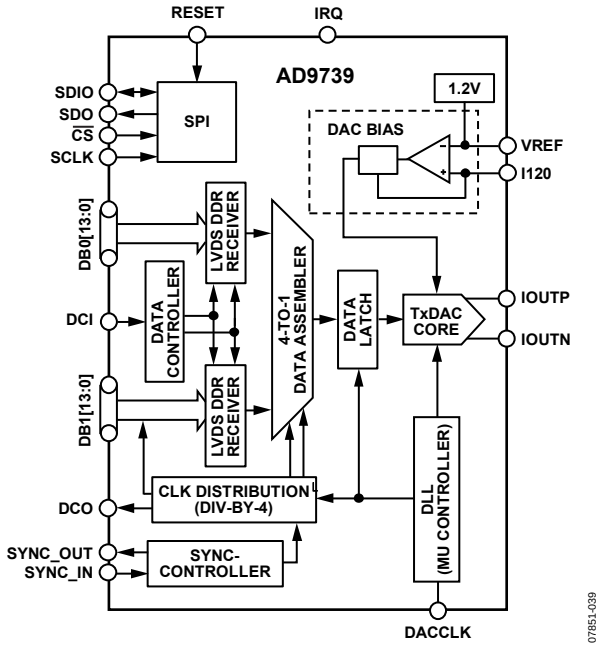



*Figure 39. Functional Block Diagram of the AD9739*

The AD9739 includes two 14-bit LVDS data ports (DB0 and DB1) to reduce the data interface rate to ½ the TxDAC update rate. The host processor drives deinterleaved data with offset binary format onto the DB0 and DB1 ports, along with an embedded DCI clock that is synchronous with the data. Because the interface is double data rate (DDR), the DCI clock is essentially an alternating 010101…….01010 bit pattern with a frequency equal to ¼ the TxDAC update rate ($f_{DAC}$). To simplify synchronization with the host processor, the AD9739 passes an LVDS clock output (DCO) that is also equal to the DCI frequency.

The AD9739 data receiver controller generates an internal sampling clock offset by 90° from the DCI to sample the input data on the DB0 and DB1 ports. When enabled and configured properly for track mode, it ensures proper data recovery between the host and the AD9739 clock domains. The data receiver controller has the ability to track several hundreds of ps of drift between these clock domains, typically caused by supply and temperature variation.

As mentioned, the host processor provides the AD9739 with a deinterleaved data stream such that the DB0 and DB1 data ports receive alternating samples (that is, odd/even data streams). The AD9739 data assembler is used to reassemble (that is, multiplex) the odd/even data streams into their original order before delivery into the TxDAC for signal reconstruction. The pipeline delay from a sample being latched into the data port to when it appears at the DAC output is on the order of 78 (±2) DACCLK cycles. Applications that require matching pipeline delays (that is, synchronization) between multiple AD9739 devices can use the SYNC controller. The SYNC controller phase aligns the outputs of one or more AD9739 devices (that is, slaves) to a master AD9739 device.

The AD9739 includes a delay lock loop (DLL) circuit controlled via a mu controller to optimize the timing hand-off between the AD9739 digital clock domain and TxDAC core. Besides ensuring proper data reconstruction, the TxDAC's ac performance is also dependent on this critical hand-off between these clock domains with speeds of up to 2.5 GSPS. Once properly initialized and configured for track mode, the DLL maintains optimum timing alignment over temperature, time, and power supply variation.

A SPI interface is used to configure the various functional blocks as well as monitor their status for debug purposes. Proper operation of the AD9739 requires that controller blocks be initialized upon power-up. A simple SPI initialization routine is used to configure the controller blocks (see Figure 51 and Figure 52). An IRQ output signal is available to alert the host should any of the controllers fall out of lock during normal operation.

The following sections discuss the various functional blocks in more detail as well as their implications when interfacing to external ICs and circuitry. While a detailed description of the various controllers (and associated SPI registers used to configure and monitor) is also included for completeness, the recommended SPI boot procedure can be used to ensure reliable operation.

## LVDS DATA PORT INTERFACE

The AD9739 supports input data rates from 1.6 GSPS to 2.5 GSPS using dual LVDS data ports. The interface is source synchronous and double data rate (DDR) where the host provides an embedded data clock input (DCI) at $f_{DAC}/4$ with its rising and falling edges aligned with the data transitions. The data format is offset binary; however, twos complement format can be realized by reversing the polarity of the MSB differential trace. As shown in Figure 40, the host feeds the AD9739 with deinterleaved input data into two 14-bit LVDS data ports (DB0 and DB1) at ½ the DAC clock rate (that is, $f_{DAC}/2$). The AD9739 internal data receiver controller then generates a phase shifted version of DCI to register the input data on both the rising and falling edges.

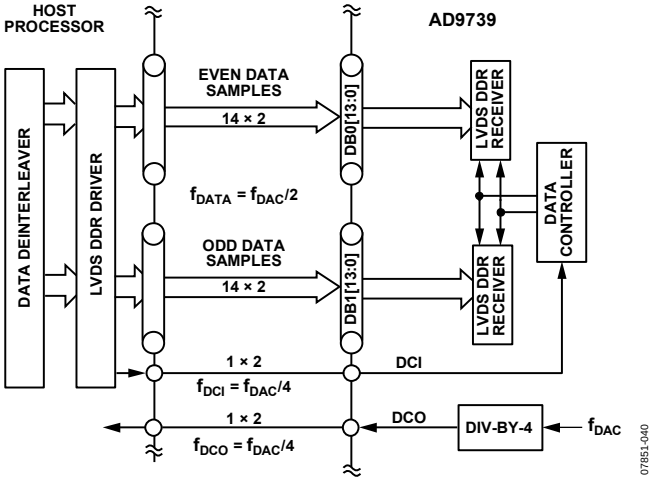



*Figure 40. Recommended Digital Interface Between the AD9739 and Host Processor*

As shown in Figure 41, the DCI clocks edges must be coincident with the data bit transitions with minimum skew, jitter, and intersymbol interference. To ensure coincident transitions with the data bits, the DCI signal should be implemented as an additional data line with an alternating (010101…) bit sequence from the same output drivers used for the data. Maximizing the opening of the eye in both the DCI and data signals improves the reliability of the data port interface. Differential controlled impedance traces of equal length (that is, delay) should also be used between the host processor and AD9739 input to limit bit-to-bit skew.

The maximum allowable skew and jitter out of the host processor with respect to the DCI clock edge on each LVDS port is calculated as

$$MaxSkew + Jitter = Period(\text{ns}) - ValidWindow(\text{ps}) - Guard$$
$$= 800\text{ ps} - 344\text{ ps} - 100\text{ ps}$$
$$= 356\text{ ps}$$

where *ValidWindow*(ps) is represented by $t_{VALID}$ and *Guard* is represented by $t_{GUARD}$ in Figure 41.

The minimum specified LVDS valid window is 344 ps, and a guard band of 100 ps is recommended. Therefore, at the maximum operating frequency of 2.5 GSPS, the maximum allowable FPGA and PCB bit skew plus jitter is equal to 356 ps.

For synchronous operation, the AD9739 provides a data clock output, DCO, to the host at the same rate as DCI (that is, $f_{DAC}/4$) to maintain the lowest skew variation between these clock domains. Since the DCO signal is generated from a separate clock divider, its phase relationship relative to the $f_{DAC}/4$ clocks used by the data receiver controller will vary upon each power-up. Applications sensitive to this phase ambiguity (resulting in a ±2 DACCLK pipeline variation) should consider using the sync controller.

The host processor has a worst-case skew between DCO and DCI that is both implementation and process dependent. This worst-case skew can also vary an additional 30% over temperature and supply corners. The delay line within the data receiver controller can track a ±1.5 ns skew variation after initial lock. While it is possible for the host to have an internal PLL that generates a synchronous $f_{DAC}/4$ from which the DCI signal is derived, digital implementations that result in the shortest propagation delays result in the lowest skew variation.

The data receiver controller is used to ensure proper data hand-off between the host and AD9739 internal digital clock domains. The circuit shown in Figure 42 functions as a delay lock loop in which a 90° phase shifted version of the DCI clock input is used to sample the input data into the DDR receiver registers. This ensures that the sampling instance occurs in the middle of the data pattern eyes (assuming matched DCI and DBx[13:0] delays). Note that, because the DCI delay and sample delay clocks are derived from the div-by-4 circuitry, this 90° phase relationship holds as long as the delay settings (that is, DCI_DEL, SMP_DEL) are also matched.

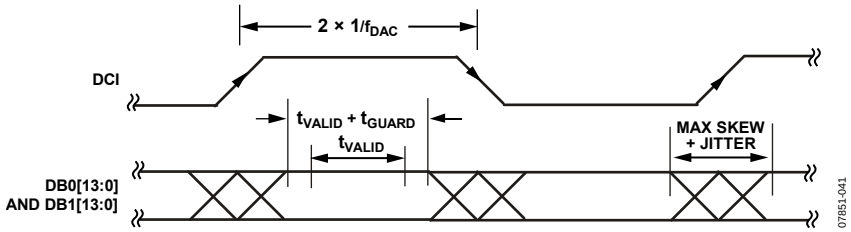



*Figure 41. LVDS Data Port Timing Requirements*

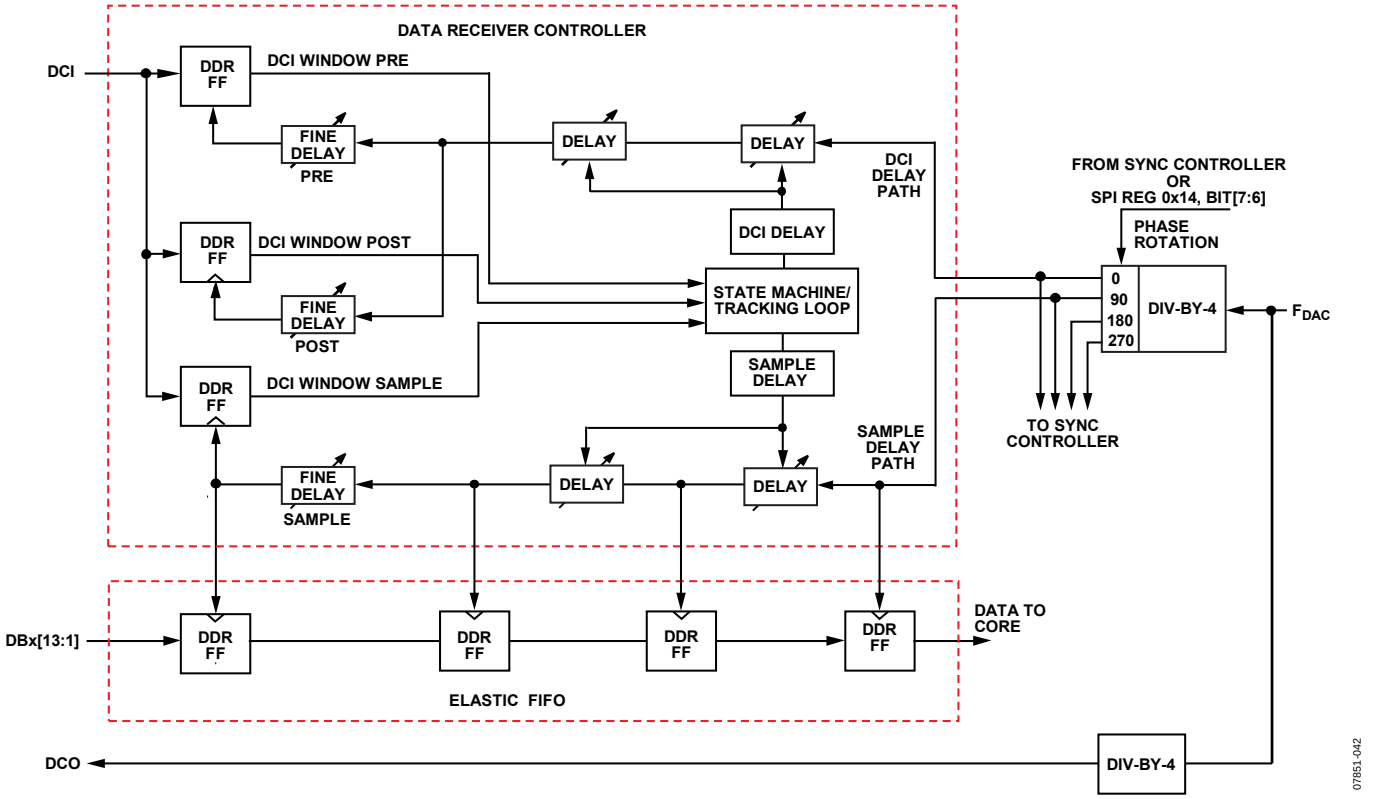



*Figure 42. Top Level Diagram of the Data Receiver Controller*

The div-by-4 circuit generates four clock phases that serve as inputs to the data receiver controller. All of the DDR registers in the data and DCI paths operate on both clock edges; however, for clarity purposes, only the phases (that is, 0° and 90°) corresponding to the positive edge of each path are shown. One of the div-by-4 phases is used to generate the DCO signal; therefore, the phase relationship between DCO and clocks fed into the controller remains fixed. Note that it is this attribute that allows possible factory calibration of images and clock spurs attributed to $f_{DAC}/4$ modulation of the critical DAC clock.

Once this data has been successively sampled into the first set of registers, an elastic FIFO is used to transfer the data into the AD9739 clock domain. To continuously track any phase variation between the two clock domains, the data receiver controller should always be enabled and placed into track mode (Register 0x10, Bit 1 and Bit 0). Tracking mode operates continuously in the background to track delay variations between the host and AD9739 clock domains. It does so by ensuring that the DCI signal is sampled within a very narrow window defined by two internally generated clocks (that is, PRE and PST), as shown in Figure 43.

Proper sampling of the DCI signal can also be confirmed by monitoring the status of DCI_PRE_PH0 (Register 0x0C, Bit 2) and DCI_PST_PH0 (Register 0x0C, Bit 0). If the delay settings are correct, the state of DCI_ PRE_PH0 should be 0, and the state of DCI_PST_PH0 should be 1. Note that the states of these status bits may toggle occasionally due to cycle-to cycle jitter exceeding the window width. However, the controller averages these status bits over multiple clock cycles to ensure that the DCI signal falls within a programmable window.

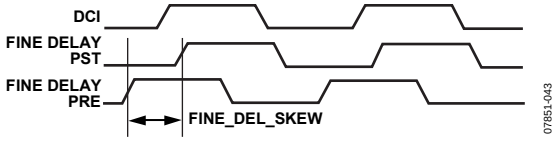



*Figure 43. Pre- and Post-Delay Sampling Diagram*

The skew or window width (FINE_DEL_SKEW) is set via Register 0x13, Bits[3:0], with a maximum skew of approximately 180 ps and resolution of 12 ps. It is recommended that the skew be set to 36 ps (that is, Register 0x13 = 0x72) during initialization. The skew setting also affects the speed of the controller loop, with tighter skew settings corresponding to longer response time.

### Data Receiver Controller Initialization Description

The data controller should be initialized and placed into track mode as the second step in the SPI boot sequence. The following steps are recommended for the initialization of the data receiver controller:

1. Set FINE_DEL_SKEW to 2 for a larger DCI sampling window (Register 0x13 = 0x72). Note that the default DCI_DEL and SMP_DEL settings of 167 are optimum.
2. Disable the controller before enabling (that is, Register 0x10 = 0x00).
3. Enable the Rx controller in two steps: Register 0x10 = 0x02 followed by Register 0x10 = 0x03.
4. Wait 135K clock cycles.
5. Read back Register 0x21 and confirm that it is equal to 0x05 to ensure that the DLL loop is locked and tracking.
6. Include this step for operation <1.6 GSPS. Read back the DCI_DEL value to determine whether the value falls within a user-defined tracking guard band. If it does not, rotate CLKDIVPH by 1 (Register 0x14, Bits[7:6] and go back to Step 2.

Once the controller is enabled during the initial SPI boot process (see Table 31 and Table 32), the controller enters a search mode where it seeks to find the closest rising edge of the DCI clock (relative to a delayed version of an internal $f_{DAC}/4$ clock) by simultaneously adjusting the delays in the clocks used to register the DCI and data inputs. A state machine searches above and below the initial DCI_DEL value. The state machine first searches for the first rising edge above the DCI_DEL and then searches for the first rising edge below the DCI_DEL value. The state machine selects the closest rising edge and then enters track mode. It is recommended that the default midscale delay setting (that is, Decimal 167) for the DCI_DEL and SMP_DEL bits be kept to ensure that the selected edge remains closest to the delay line midpoint, thus providing the greatest range for tracking timing variations and preventing the controller from falling out of lock.

The adjustable delay span for these internal clocks (that is, DCI and sample delay) is nominally 4 ns. The 10-bit delay value is user programmable from the decimal equivalent code (0 to 384) with approximately 12 ps/LSB resolution via the DCI_DEL and SMP_DEL registers (via Register 0x11 thru Register 0x14). When the controller is enabled, it overwrites these registers with the delay value it converges upon. The minimum difference between this delay value and the minimum/maximum values (that is, 0 and 334) represents the guard band for tracking. Therefore, if the controller initially converges upon a DCI_DEL and SMP_DEL value between 80 and 304, the controller has a guard band of at least 80 code (approximately 1 ns) to track phase variations between the clock domains.

Upon initialization of the AD9739, a certain period of time is required for the data receiver controller to lock onto the DCI clock signal. Note that, due to its dependency on the mu controller and synchronization controller (optional), the data receiver controller should be enabled only after these other controllers have been enabled and established locked. All of the internal controllers operate at submultiples of the DAC update rate. The number of $f_{DAC}$ clock cycles required to lock onto the DCI clock is dependent on whether the synchronization controller is enabled as shown in Table 26.

**Table 26. Typical/Worst-Case Lock Times for LVDS Controller (Relative to $1/f_{DAC}$)**

| Synchronization Controller | Typical | Worst Case |
|---|---|---|
| Off | 70K | 135K |
| Slave | 70K | 135K |
| Master | 300K | 560K |

During the SPI initialization process, the user has the option of polling Register 0x21 (Bit 0, Bit 1, and Bit 3) to determine if the data receiver controller is locked, has lost lock, or has entered into track mode before completing the boot sequence. Alternatively, the appropriate IRQ bit (Register 0x03 and Register 0x04) can be enabled such that an IRQ output signal is generated upon the controller establishing lock (see the Interrupt Requests section).

The data receiver controller can also be configured to generate an interrupt request (IRQ) upon losing lock. Losing lock can be caused by excessive jitter on the DCI input signal, disruption of the main DAC clock input, or loss of a power supply rail. To service the interrupt, the host can poll the RCVR_LCK bit (Register 0x21, Bit 0) to determine the current state of the controller. If this bit is cleared, the search/track procedure can be restarted by setting the RCVR_LOOP_ON bit in Register 0x10, Bit 1. After waiting the required lock time, the host can poll the RCVR_LCK bit to see if it has been set. Before leaving the interrupt routine, the RCVR_FLG_RST bit (Register 0x10, Bit 2) should be reset by writing a high followed by a low.

### Data Receiver Operation at Lower Clock Rates

At clock rates below 1.6 GSPS, it is recommended to include provisions to rotate the CLKDIVPH setting in the SPI boot process. As previously mentioned, the delay line can be varied over a nominal 4 ns window. If the minimum specified clock rate of 800 MSPS is considered, a DCI clock rate of 200 MSPS corresponds to a 5 ns period, thus exceeding the delay line length. Therefore, it becomes possible that the initial startup phase from the div-by-4 circuit (and DCO output) is such that the data receiver controller can never establish initial lock upon power up.

If the clock rate is increased to 1600 MSPS (that is, DCI clock period of 2.5 ns), the controller will always be able to find at least two DCI clock edges, therefore, establish lock. However, should the DCI edges fall symmetrically (equal distance) from the initial DCI_DEL midscale setting, a guard band of ±0.75 ns (that is, (4.0 – 2.5)/2) results. Rotating the CLKDIVPH can result in an improvement in this case by skewing one of the DCI edges toward the DCI_DEL midscale value.

Rotating the CLKDIVPH phase provides a means of adjusting the delay in course steps of $f_{DAC}/4$. For example, in the 800 MSPS and 1600 MSPS cases described above, rotating the CLKDIVPH setting by 1 corresponds to a delay shift of 5 ns and 2.5 ns, respectively. By adding an additional step in the SPI initialization routine for the data receiver controller, it becomes possible to increase the effective range of the delay line to ensure a DCI_DEL value that falls within a reasonable guard band.

### LVDS Driver and Receiver Input

The AD9739 features a LVDS-compatible driver and receivers. The LVDS driver output used for the DCO and SYNC_OUT signal includes an equivalent 200 Ω source resistor that limits its nominal output voltage swing to ±200 mV when driving a 100 Ω load. The DCO output driver can be powered down via Register 0x01, Bit 5. An equivalent circuit is shown in Figure 44

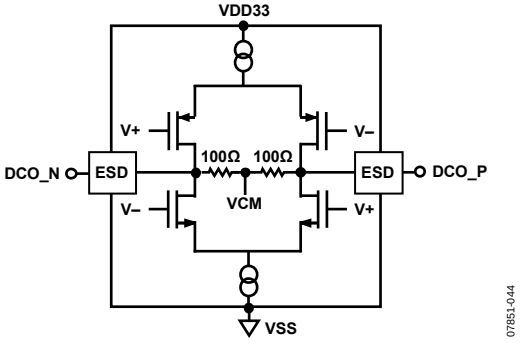



*Figure 44. Equivalent LVDS Output*

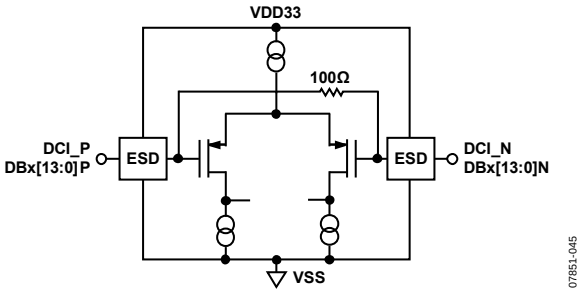



*Figure 45. Equivalent LVDS Input*

The LVDS receivers include 100 Ω termination resistors, as shown in Figure 45. These receivers meet the IEEE-1596.3-1996 reduced swing specification (with the exception of input hysteresis, which cannot be guaranteed over all process corners). Figure 46 and Figure 47 show an example of nominal LVDS voltage levels seen at the input of the differential receiver with resulting common-mode voltage and equivalent logic level. The LVDS receivers can be powered down via Register 0x01, Bit 4.

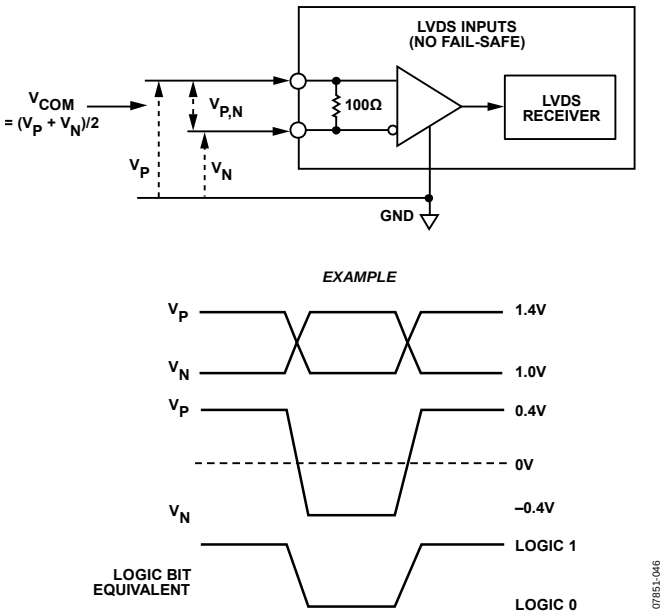



*Figure 46. LVDS Data Input Levels*

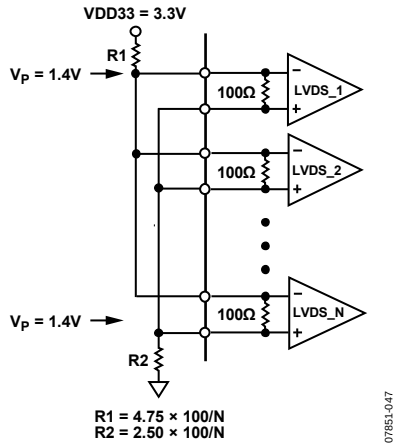



*Figure 47. Resistor Network to Bias Unused LVDS Data Inputs*

The AD9739 LVDS inputs do not include fail-safe capability. Any unused data input pins should be biased with an external network or static driver. Figure 47 shows an external biasing network that can be used to place unused data bits into a known state. The resistor values for R1 and R2 are selected to establish a $V_P$ and $V_N$ of 1.4 V and 1.0 V, respectively, depending on the number of unused digital inputs, N.

**Table 27. Example of LVDS Input Levels**

| Applied Voltages | | Resulting Differential Voltage | Resulting Common-Model Voltage | |
|---|---|---|---|---|
| $V_P$ (V) | $V_N$ (V) | $V_{P,N}$ | $V_{COM}$ | Logic Bit Binary Equivalent |
| 1.4 | 1.0 | +0.4 V | 1.2 V | 1 |
| 1.0 | 1.4 | −0.4 V | 1.2 V | 0 |
| 1.0 | 0.8 | +200 mV | 900 mV | 1 |
| 0.8 | 1.0 | −200 mV | 900 mV | 0 |

## MU CONTROLLER

A delay lock loop (DLL) is used to optimize the timing between the internal digital and analog domains of the AD9739 such that data is successfully transferred into the TxDAC core at rates of up to 2.5 GSPS. As shown in Figure 48, the DAC clock is split into an analog and a digital path with the critical analog path leading to the DAC core (for minimum jitter degradation) and the digital path leading to a programmable delay line. Note that the output of this delay line serves as the master internal digital clock from which all other internal and external digital clocks are derived. The amount of delay added to this path is under the control of the mu controller, which optimizes the timing between these two clock domains and continuously tracks any variation (once in track mode) to ensure proper data hand-off.

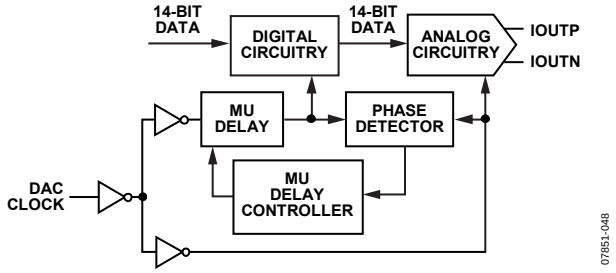



*Figure 48. Mu Delay Controller Block Diagram*

The mu controller adjusts the timing relationship between the digital and analog domains via a tapped digital delay line having a nominal total delay of 864 ps. The delay value is programmable to a 9-bit resolution (that is, 0 to 432 decimal) via the MUDEL register, resulting in a nominal resolution of 2 ps/LSB. Because a time delay maps to a phase offset for a fixed clock frequency, the control loop essentially compares the phase relationship between the two clock domains and adjusts the phase (that is, via a tapped delay line) of the digital clock such that it is at the desired fixed phase offset (SET_PHS) from the critical analog clock.

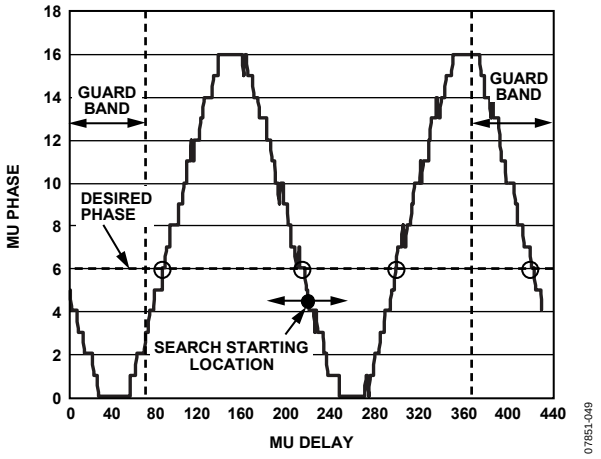



*Figure 49. Typical Mu Phase Characteristic Plot at 2.4 GSPS*

Figure 49 maps the typical mu phase characteristic at 2.4 GSPS vs. the 9-bit digital delay setting (MUDEL). The mu phase scaling is such that a value of 16 corresponds to 180 degrees. The critical keep-out window between the digital and analog domains occurs at a value of 0 (but can extend out to 2 depending on the clock rate). The target mu phase (and slope) is selected to provide optimum ac performance while ensuring that the mu controller for any device can establish and maintain lock. For example, while a slope and phase setting of −6 is considered optimum for operation between 1.6 GSPS and 2.5 GSPS, other values are required below 1.6 GSPS.

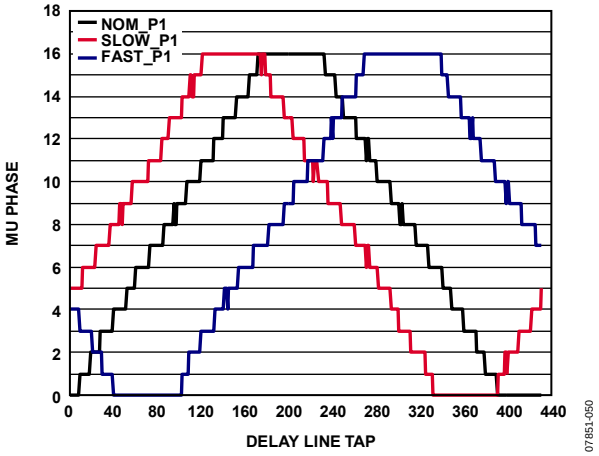



*Figure 50. Mu Phase Characteristics of Three Devices from Different Process Lots at 1.2 GSPS*

The mu phase characteristics can vary significantly among devices due to gm variations in the digital delay line that are sensitive to process skews (along with temperature and supply). As a result, careful selection of the target phase location is required such that the mu controller can converge upon this phase location for all devices. Figure 50 shows that mu phase characteristics of three devices at 25°C from slow, nominal, and fast skew lots at 1.2 GSPS. Note that a −6 mu phase setting does not map to any delay line tap setting for the fast process skew case; therefore, another target mu phase is recommended at this clock rate.

Table 28 provides a list of recommended mu phase/slope settings over the specified clock range of the AD9739 based on the considerations previously described. These values should be used to ensure robust operation of the mu controller.

**Table 28. Recommended Target Mu Phase Settings vs. Clock Rate**

| Clock Rate (GSPS) | Slope | MU Phase |
|---|---|---|
| 0.8 | − | 6 |
| 0.9 | − | 4 |
| 1.0 | + | 5 |
| 1.1 | + | 8 |
| 1.2 | + | 12 |
| 1.3 | − | 12 |
| 1.4 | − | 10 |
| 1.5 | − | 8 |
| 1.6 to 2.5 | − | 6 |

After the mu controller completes its search and establishes lock on the target mu phase, it attempts to maintain a constant timing relationship between the two clock domains over the specified temperature and supply range. If the mu controller requests a mu delay setting that exceeds the tapped delay line range (that is, <0 or >432), the mu controller can lose lock, causing possible system disruption (that is, can generate IRQ or restart the search). To avoid this scenario, symmetrical guard bands are recommended at each end of the mu delay range. The guard band scaling is such that one LSB of Guard[4:0] (Register 0x29) corresponds to eight LSBs of MUDEL (Register 0x28). The recommended guard band setting of 11 (that is, Register 0x29 = 0xCB) corresponds to 88 LSBs, thus providing sufficient margin.

### *Mu Controller Initialization Description*

The mu controller must be initialized and placed into track mode as a first step in the SPI boot sequence. The following steps are required for initialization of the mu controller. Note that the AD9739 data sheet specifications and characterization data are based on the following mu controller settings:

1. Turn on the phase detector with boost (Register 0x24 = 0x30).
2. Enable the mu delay controller duty-cycle correction circuitry and specify the recommended slope for phase. (that is, Register 0x25 = 0x80 corresponds to a negative slope).
3. Specify search/track mode with a recommended target phase, SET_PHS, of 6 (for example) and an initial MUDEL[8:0] setting of 216 (Register 0x27 = 0x46 and Register 0x28 = 0x6C).
4. Set search tolerance to exact and retry if the search fails its initial attempt. Also, set the guard band to the recommended setting of 11 (Register 0x29 = 0xCB).
5. Set the mu controller tracking gain to the recommended setting and enable the mu controller state machine (Register 0x26 = 0x03).

Upon completion of the last step, the mu controller begins a search algorithm that starts with an initial delay setting specified by the MUDEL register (that is, 216, which corresponds to the midpoint of the delay line). The initial search algorithm works by sweeping through different mu delay values in an alternating manner until the desired phase (that is, a SET_PHS of 4) is exactly measured. When the desired phase is measured, the slope of the phase measurement is then calculated and compared against the specified slope (slope = negative).

If everything matches, the search algorithm is finished. If not, the search continues in both directions until an exact match can be found or a programmable guard band is reached in one of the directions. When the guard band is reached, the search still continues but only in the opposite direction. If the desired phase is not found before the guard band is reached in the second direction, the search changes back to the alternating mode and continues looking within the guard band. The typical locking time for the mu controller is approximately 180K DAC cycles (at 2 GSPS ~ 75 µs).

The search fails if the mu delay controller reaches the endpoints. The mu controller can be configured to retry (Register 0x29, Bit 6) the search or stop. For applications that have a microcontroller, the preferred approach is to poll the MU_LKD status bit (Register 0x2A, Bit 0) after the typical locking time has expired. This method allows the system controller to check the status of other system parameters (that is, power supplies and clock source) before reattempting the search (by writing 0x03 to Register 0x26). For applications that do not have polling capabilities, the mu controller state machine should be reconfigured to restart the search in hopes that the system's condition that did not cause locking on the first attempt has disappeared.

Once the mu delay value is found that exactly matches the desired mu phase setting and slope (for example, 6 with a negative. slope), the mu controller goes into track mode. In this mode, the mu controller makes slight adjustments to the delay value to track any variations between the two clock paths due to temperature, time, and supply variations. Two status bits, MU_LKD (Register 0x2A, Bit 0) and MU_LST (Register 0x2A, Bit 1) are available to the user to signal the existing status control loop. If the current phase is more than four steps away from the desired phase, the MU_LKD bit is cleared, and if the lock acquired was previously set, the MU_LST bit is set. Should the phase deviation return to within three steps, the MU_LKD bit is set again while the MU_LST is cleared. Note that this sort of event may occur if the main clock input (that is, DACCLK) is disrupted or the mu controller exceeds the tapped delay line range (that is, <0 or >432).

If lock is lost, the mu controller has the option of remaining in the tracking loop or resetting and starting the search again via the CONTRST bit (Register 0x29, Bit 5). Continued tracking is the preferred state because it is the least disruptive to a system in which the AD9739 temporarily loses lock. The user can poll the mu delay and phase value by first setting the read bit high (Register 0x26, Bit 3). Once the read bit is set, the MUDEL[8:0] bits and the SET_PHS[4:0] bits (Register 0x27 and Register 0x28) that the controller is currently using can be read.

## INTERRUPT REQUESTS

The AD9739 can provide the host processor with an interrupt request output signal (IRQ) that indicates that one or more of the AD9739 internal controllers have achieved lock or lost lock. These controllers include the mu, data receiver, and synchronization controllers. The host can then poll the IRQ status register (Register 0x04) to determine which controller has lost lock. The IRQ output signal is an active high output signal available on Pin F13. If used, its output should be connected via a 10 kΩ pull-up resistor to VDD33.

Each IRQ is enabled by setting the enable bits in Register 0x03, which purposely has the same bit mapping as the IRQ status bits in Register 0x04. Note that these IRQ status bits are set only when the controller transitions from a false to true state. Therefore, it is possible for the x_LCK_IRQ and x_LST_IRQ status bits to be set when a controller temporarily loses lock but is able to reestablish lock before the IRQ is serviced by the host. In this case, the host should validate the present status of the suspect controller by reading back its current status bits, which are available in Register 0x21 and/or Register 0x2A. Based on the status of these bits, the host can take appropriate action, if required, to reestablish lock. To clear an IRQ after servicing, it is necessary to reset relevant bits in Register 0x03 by writing 0 followed by another write of 1 to reenable. A detailed diagram of the interrupt circuitry is shown in Figure 51.

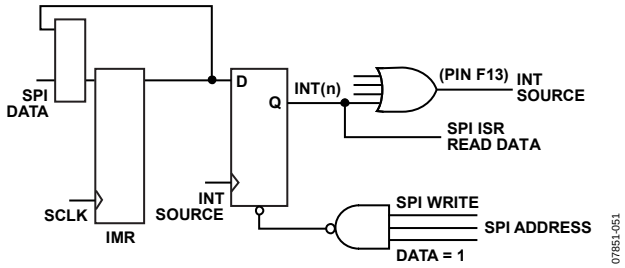



*Figure 51. Interrupt Request Circuitry*

It is also possible to use the IRQ during the AD9739 initialization phase after power-up to determine when the mu and data receiver controllers have achieved lock. For example, before enabling the mu controller, the MU_LCK_EN bit (Register 0x03, Bit 2) can be set and the IRQ output signal monitored to determine when lock has been established before continuing in a similar manner with the data receiver controllers. Note that the relevant LCK bit should be cleared before continuing to the next controller. After all controllers are locked, the lost lock enable bits (that is, x_LST_EN) should be set.

**Table 29. Interrupt Request Registers**

| Address (Hex) | Bit | Description |
|---|---|---|
| 0x03 | 5 | SYNC_LST_EN |
| | 4 | SYNC_LCK_EN |
| | 3 | MU_LST_EN |
| | 2 | MU_LCK_EN |
| | 1 | RCV_LST_EN |
| | 0 | RCV_LCK_EN |
| 0x04 | 5 | SYNC_LST_IRQ |
| | 4 | SYNC_LCK_IRQ |
| | 3 | MU_LST_IRQ |
| | 2 | MU_LCK_IRQ |
| | 1 | RCV_LST_IRQ |
| | 0 | RCV_LCK_IRQ |
| 0x21 | 7 | SYNC_TRK_ON |
| | 5 | SYNC_LST |
| | 4 | SYNC_LCK |
| | 3 | RCVR_TRK_ON |
| | 1 | RCVR_LST |
| | 0 | RCVR_LCK |
| 0x2A | 1 | MU_LST |
| | 0 | MU_LKD |

## MULTIPLE DEVICE SYNCHRONIZATION

Synchronization of multiple AD9739devices requires all of the devices to have matching pipeline delays. This implies the DAC outputs are time aligned to the same phase when all devices are fed with the same data pattern at the same instance of time. The main contributor to phase ambiguity between devices is from the div-by-4 circuitry that drives the Rx data path and data controller (see Figure 53). This phase ambiguity can result in a ±2 sample offset between any two devices. Because the state of this internal divider is unknown at power-up, a synchronization method that phase aligns the digital paths of multiple AD9739 devices is required to ensure matching pipeline delays.

Figure 52 shows a top-level diagram of multiple AD9739 devices synchronized to each other with sample alignment of the different data streams within the FPGA (or among multiple FPGAs) being assumed. A common RF clock source is distributed to each of the AD9739 devices via a dual clock buffer (such as the ADCLK946) with matched PCB trace lengths to each device to ensure matched propagation delays.

One AD9739 is designated as the master providing a SYNC_OUT reference clock (equal to $f_{DAC}/4$) to itself as well as the other AD9739 slave device's SYNC_IN input. LVDS fanout buffers with matched output delays are again used to distribute the SYNC_OUT and DCO signals of the master to the slave devices and FPGAs, respectively, thus ensuring tight time alignment. Note, in the case of a single FPGA implementation (that is, I/Q application), the DCO of the master can drive the FPGA directly.

After synchronization, the internal div-by-4 circuitry will have equal phases that drive their respective LVDS controllers. Note, the mu and data receiver controller of both devices must be configured for the same SPI register settings (that is, SET_PHS and DCI_DEL) upon SPI initialization such that controllers converge to similar delays. To validate that delays are roughly matched, the user can read back the delays of both devices (that is, MUDEL and DCI_DEL) to determine if they are in an acceptable window that accounts for slight mismatches between different devices' delay lines.

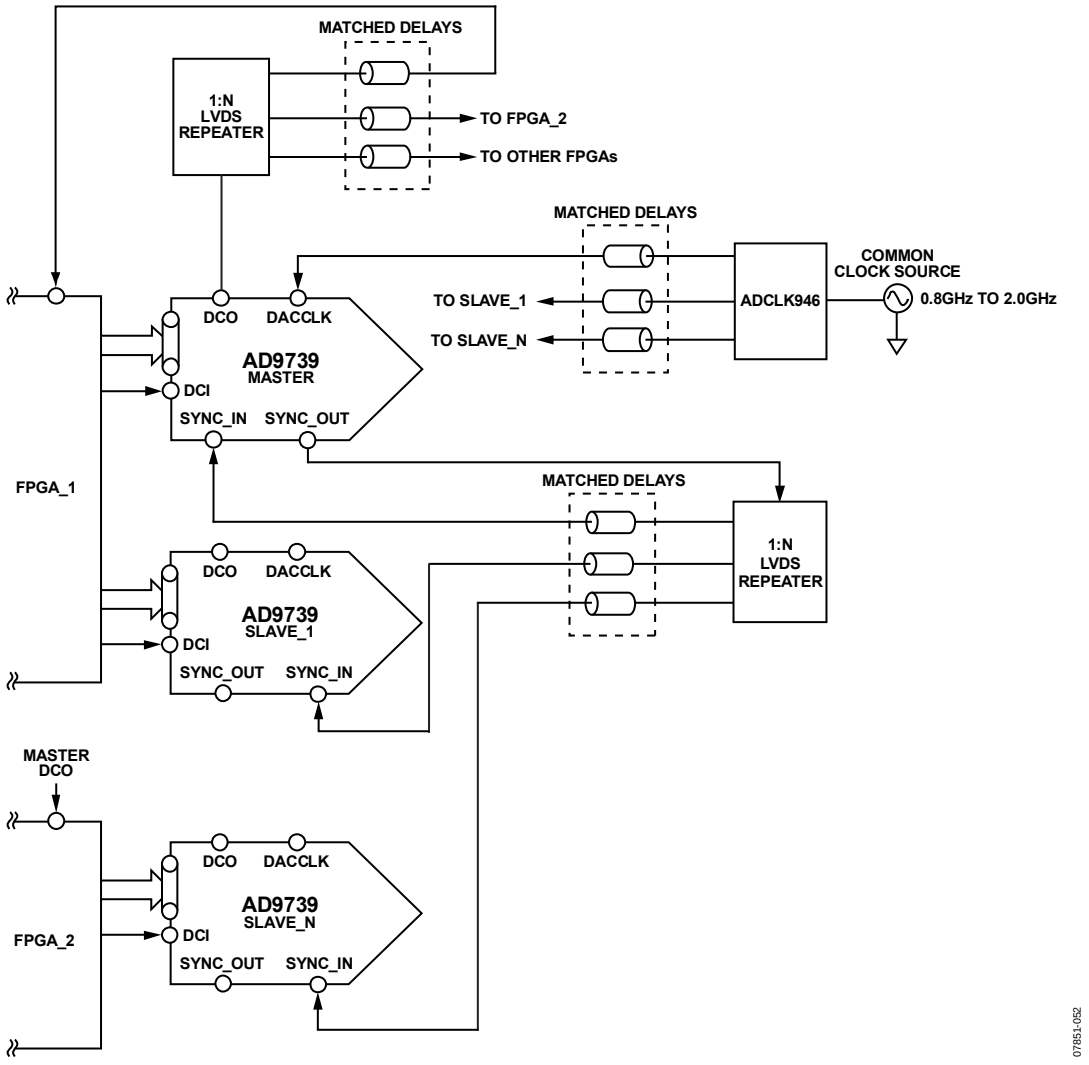



*Figure 52. Functional Block Diagram of Two AD9739 Devices Synchronized*

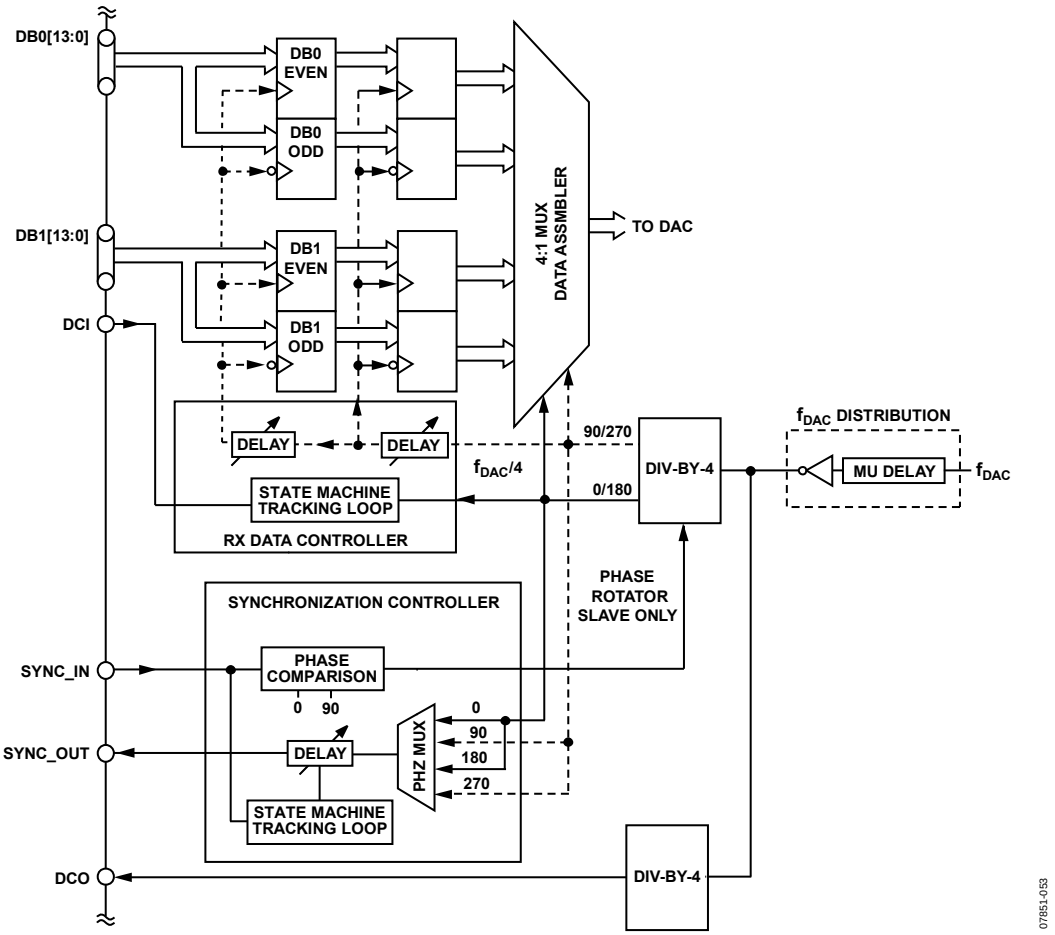



*Figure 53. Top Level Block Diagram of Synchronization Circuitry and Controller*

Figure 53 shows a top-level diagram of the synchronization controller (bottom) and how it interfaces to other digital functional blocks within the AD9739. Note the following observations of this top level diagram:

- Synchronization between multiple devices is achieved by rotating the div-by-4 phases of the slave devices such that they align with the master.
- For the slave devices, the sync controller compares the phase alignment of the master's SYNC_IN reference signal with the initial 0°/90° outputs of the div-by-4 and then rotates the div-by-4 phase until the SYNC_IN signal falls between these phases.
- A reference signal common to all devices is required for synchronization. The master device generates this signal by providing a SYNC_OUT signal which is then distributed to all the devices (including itself with tight time alignment) as a SYNC_IN signal.
- Because the SYNC_IN signal has a defined relationship between the div-by-4 phase of the master, the slave devices can now align their respective div-by-4 phases to the SYNC_IN phase thus ensuring phase alignment among all devices.
- It is not possible to manually rotate the div-by-4 phases of the data path with the sync controller enabled. This can be a problem at lower clock rates were one may desire to rotate the div-by-4 phase to ensure locking of the data receiver controller and/or achieve a more optimum DCI_DEL value.
- The DCO output signal is generated from a separate div-by-4 circuit, and therefore, has a random phase upon each startup. For this reason, the DCO of the master should be distributed to all the FPGAs.

### *SYNC Controller Initialization Description*

The sync controller of the master is enabled by writing 0x70 to Register 0x10. Once enabled, a state machine automatically adjusts the output delay of its SYNC_OUT signal such that the fed back reference SYNC_IN signal is centered between the 0° and 90° output phases associated with its div-by-4 circuit. Note that the coarse delay is performed by shifting phases via PHZ MUX while the fine delay that centers (and tracks) variation is done by a variable delay line. The variable delay line tap size is 12 ps. Once SYNC_IN is centered, the controller enters tracking mode such that SYNC_IN remains centered despite possible system variations in temperature and/or supply. Centering the SYNC_IN signal on the master device ensures that the SYNC_IN signals of the slave devices also remain centered between their respective div-by-4 phases; therefore, providing the greatest margin to absorb nonideal timing skews. The following status bits are available in Register 0x21 indicating lock, lost-lock, and tracking: SYNC_LCK, SYNC_LST and SYNC_TRK_ON.

The sync controller of the slave is enabled by writing 0x50 to Register 0x10. Once enabled, the state machine compares the reference SYNC_IN signal to the 0°/90° phase outputs of the div-by-4 phase settings. If the SYNC_IN signal does not fall between these phases, the state machine of the slave rotates the div-by-4 phase setting until it does. To validate that phase alignment has been achieved, the SYNC_IN_PH90 and SYNC_IN_PH0 status bits should read 1 and 0, respectively (that is, Register 0x0D, Bits[5:4]). Note that the DCO and SYNC_OUT outputs of the slave can be disabled via Register 0x01, Bit 5.

### *Synchronization Limitations*

Ensuring consistent synchronization over production lots in systems containing two or more AD9739 devices becomes increasingly more challenging at the higher update rates because the timing offset between adjacent phases of the div-by-4 output clock is equal to $1/f_{DAC}$ . For example, a DAC update of 2 GSPS corresponds to a 500 ps period. If the SYNC_IN signal of an ideal master device is positioned in the center of its div-by-4 0° and 90° phase outputs, only ±250 ps of timing margin exists for the slave devices. This ideal margin is actually reduced by quadrature phase errors in the div-by-4 circuit of the master as well as its ability to position the SYNC_IN exactly in the center of the 0° and 90° output phases.

The timing margin is further eroded by the following sources:

- Master-to-slave device(s) mismatch in the propagation delays in the mu delay clock path and SYNC_IN. Note that these mismatches can be up to 100 ps between devices that are at opposite extremes of the process corners.
- Quadrature phase errors in the div-by-4 outputs of the slave.

These sources of timing skews become more significant as the DACCLK period is decreased (that is, clock rate is increased), leaving less margin for timing skews external to the master-to-slave device(s). Special consideration to PCB layout and selection of clock distribution ICs are required to ensure minimum skew between the distributed DACCLK and SYNC_IN signals. Note that timing skews can quickly accumulate considering that the propagation delay on an FR4 PCB is on the order of 170 ps/inch, and that output-to-output skews on each clock distribution IC can be as high as 25 ps.

The problem becomes more pronounced in multiboard synchronization where clock signals (that is, DACCLK, SYNC_OUT, and DCO) are distributed over a back plane to multiple PCBs. Data alignment among the various data sources is required when driven by phase aligned DCO signals that are a buffered version of the master's DCO. However, these data sources (FPGAs) also have process, supply voltage, and temperature sensitivities (PVTs) that can cause misalignment among their respective DCI outputs.

Adding to this dilemma is that it also possible for the data receiver controller of different AD9739 devices to converge on different delay settings due to PVT variations of the delay line (even if DCI inputs are exactly aligned). This can result in a four sample pipeline mismatch between devices if the difference in absolute delays exceeds a period of $4/f_{DAC}$. Recall that the controller searches up/down for its first valid edge from its initial start value (that is, DCI_DEL and SMP_DEL). While the initial start values between devices should be made the same, different absolute time delays due to PVT can cause devices to converge on different edges of DCI above or below this initial start value. As a result, confirm that DCI_DEL values between multiple devices are matched sufficiently such that the absolute differences between the readback DCI_DEL values do not exceed a data period (that is, $4/f_{DAC}$). If the difference exceeds a data period, modify the DCI_DEL (and SMP_DEL) setting of the slave device so that its start point is roughly ½ the difference between the master and slave readback values.

# ANALOG INTERFACE CONSIDERATIONS

## ANALOG MODES OF OPERATION

The AD9739 uses the quad-switch architecture shown in Figure 54. The quad-switch architecture masks the code-dependent glitches that occur in a conventional two-switch DAC. Figure 55 compares the waveforms for a conventional DAC and the quad-switch DAC. In the two-switch architecture, a code-dependent glitch occurs each time the DAC switches to a different state (that is, D1 to D2). This code-dependent glitching causes an increased amount of distortion in the DAC. In a quad-switch architecture (no matter what the codes are), there are always two switches transitioning at each half clock cycle, thus eliminating the code-dependent glitches. However, a constant glitch occurs at 2 × DACCLK because half of the internal switches change state on the rising DACCLK edge, while the other half change state on the falling DACCLK edge.

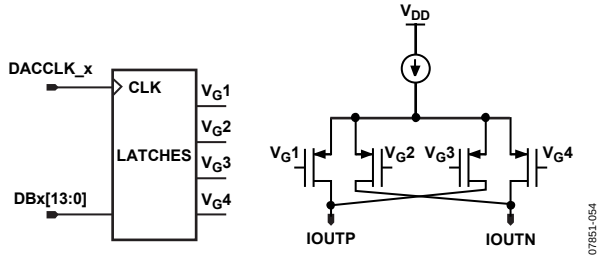



*Figure 54. AD9739 Quad-Switch Architecture*

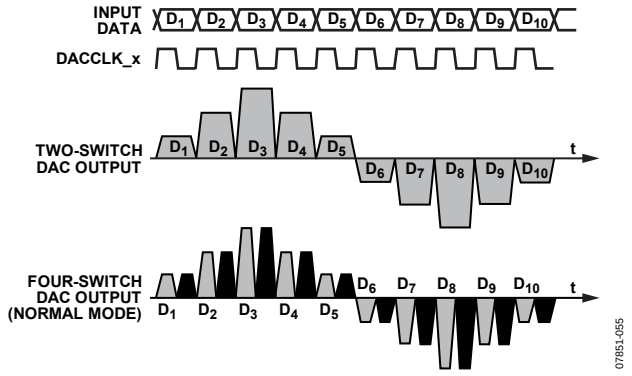



*Figure 55. Two-Switch and Quad-Switch DAC Waveforms*

Another attribute of the quad-switch architecture is that it also enables the DAC core to operate in one of the following three modes: normal mode, mix mode, and return-to-zero (RZ) mode. The mode is selected via SPI Register 0x08, Bits[1:0] with normal mode being the default value. In the mix mode, the output is effectively chopped at the DAC sample rate. This has the effect of reducing the power of the fundamental signal while increasing the power of the images centered around the DAC sample rate, thus improving the output power of these images. The RZ mode is similar to the analog mix mode, except that the intermediate data samples are replaced with midscale values.

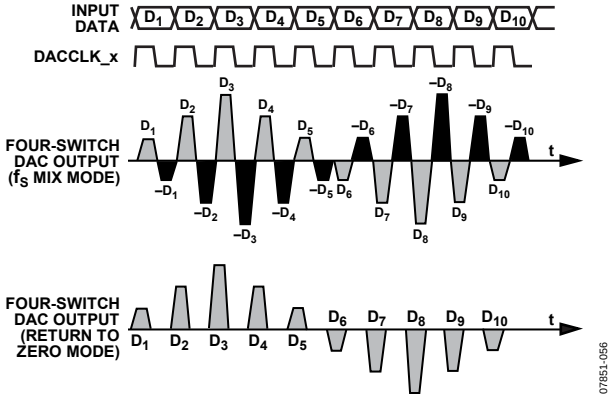



*Figure 56. Mix-Mode and RZ DAC Waveforms*

Figure 56 shows the DAC waveforms for both the mix mode and the RZ mode. Note that the disadvantage of the RZ mode is the 6 dB loss of power to the load because the DAC is only functioning for ½ the DAC update period. This ability to change modes provides the user the flexibility to place a carrier anywhere in the first three Nyquist zones, depending on the operating mode selected. Switching between the analog modes reshapes the sinc roll-off inherent at the DAC output. The maximum amplitude in all three Nyquist zones is impacted by this sinc roll-off, depending on where the carrier is placed (see Figure 57). As a practical matter, the usable bandwidth in the third Nyquist zone becomes limited at higher DAC clock rates (that is, >2 GSPS) when the output bandwidth of DAC core and the interface network (that is, balun) contributes to additional roll-off.

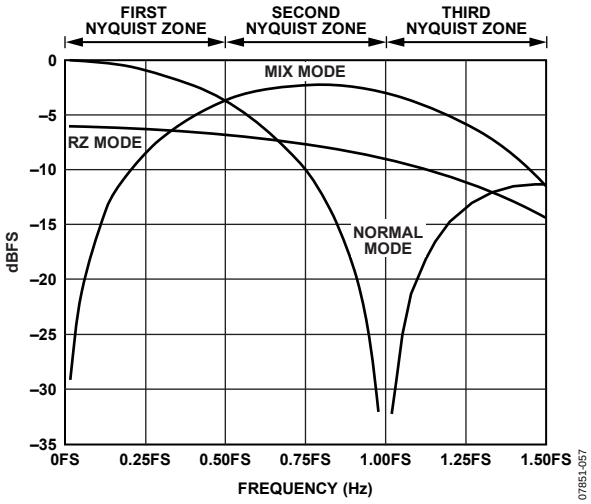



*Figure 57. Sinc Roll-Off for Each Analog Operating Mode*

## CLOCK INPUT CONSIDERATIONS

The quality of the clock source and its drive strength are important considerations in maintaining the specified ac performance. The phase noise and spur characteristics of the clock source should be selected to meet the target application requirements. Phase noise and spurs at a given frequency offset on the clock source are directly translated to the output signal. It can be shown that the phase noise characteristics of a reconstructed output sine wave are related to the clock source by 20 × log10 ($f_{OUT}/f_{CLK}$) when the DAC clock path contribution, along with thermal and quantization effects, are negligible.

The AD9739 clock receiver provides optimum jitter performance when driven by a fast slew rate originating from the LVPECL or CML output drivers. For optimal ac performance out of the AD9739, the recommended minimum differential peak-to-peak voltage is approximately 1.4 V p-p. For a low jitter sinusoidal clock source, the ADCLK914 can be used to square-up the signal and provide a CML input signal for the AD9739 clock receiver. Note that all specifications and characterization presented in the data sheet are with the ADCLK914 driven by a high quality RF signal generator with the clock receiver biased at an 800 mV level. A dc blocking capacitor is used between the clock driver output and clock receiver input to allow for different dc bias levels. To minimize signal loss for high clock rates, a high quality, dc blocking RF capacitor is recommended.

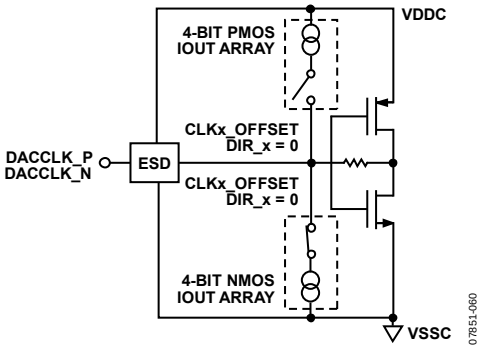



*Figure 58. Clock Input and Common-Mode Control*

The AD9739 clock receiver features the ability to independently adjust the common-mode level of its inputs over a span of ±100 mV centered about its midsupply point (that is, VDDC/2) as well as an offset for hysteresis purposes. Figure 58 shows the equivalent input circuit of one of the inputs. ESD diodes are not shown for clarity purposes. It has been found through characterization that the optimum setting is for both inputs to be biased at approximately 0.8 V. This can be achieved by writing a 0x0F (corresponding to a −15) setting to both cross controller registers (that is, Register 0x22 and Register 0x23).

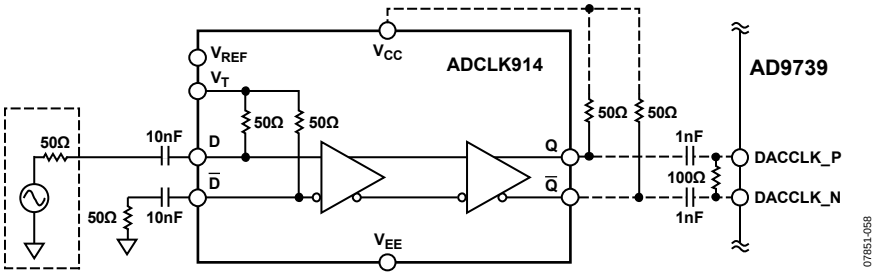



*Figure 59. ADCLK914 Interface to the AD9739 CLK Input*

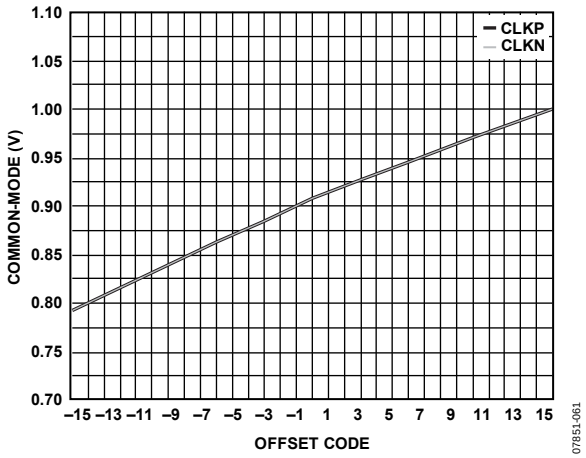



*Figure 60. Common-Mode Voltage with Respect to CLKP_OFFSET/CLKN_OFFSET and DIR_P/DIR_N*

## VOLTAGE REFERENCE

The AD9739 output current is set by a combination of digital control bits and the I120 reference current, as shown in Figure 61.

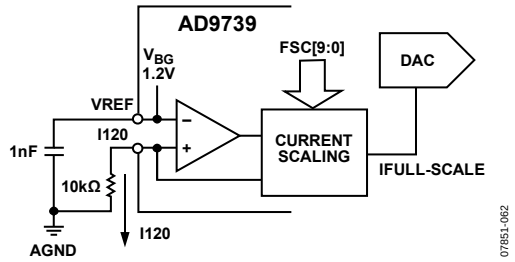



*Figure 61. Voltage Reference Circuit*

The reference current is obtained by forcing the band gap voltage across an external 10 kΩ resistor from I120 (Pin B14) to ground. The 1.2 V nominal band gap voltage (VREF) generates a 120 μA reference current in the 10 kΩ resistor. Note the following constraints when configuring the voltage reference circuit:

- Both the 10 kΩ resistor and 1 nF bypass capacitor are required for proper operation.
- Adjusting the output full-scale current, $I_{OUTFS}$, of the DAC from its default setting of 20 mA should be performed digitally.
- The AD9739 is not a multiplying DAC. Modulating the reference current, I120, with an ac signal is not supported.
- The band gap voltage appearing at VREF (Pin C14) must be buffered for use with external circuitry because its output impedance is approximately 5 kΩ.
- An external reference can be used to overdrive the internal reference by connecting it to VREF (Pin C14).

$I_{OUTFS}$ can be adjusted digitally over 8.7 mA to 31.7 mA by using FSC[9:0] (Register 0x06 and Register 0x07).

The following equation relates $I_{OUTFS}$ to the FSC[9:0] register, which can be set from 0 to 1023:

$$I_{OUTFS} = 22.6 \times FSC[9:0]/1000 + 8.7 \quad (1)$$

Note that a default value of 0x200 generates 20 mA full scale, which is used for most of the characterization presented in this data sheet (unless noted otherwise).

## ANALOG OUTPUTS

### *Equivalent DAC Output and Transfer Function*

The AD9739 provides complementary current outputs, IOUTP and IOUTN, that source current into an external ground reference load. Figure 62 shows an equivalent output circuit for the DAC. Note that, compared to most current output DACs of this type, the AD9739 outputs exhibit a slight offset current (that is, $I_{OUTFS}/16$), and the peak differential ac current is slightly below $I_{OUTFS}/2$ (that is, 15/32 × $I_{OUTFS}$).

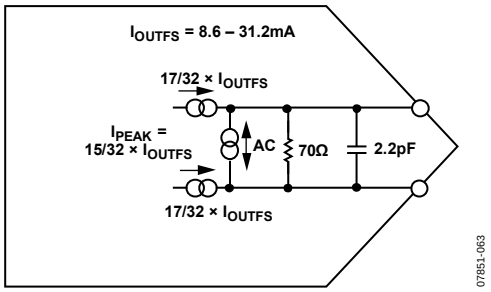



*Figure 62. Equivalent DAC Output Circuit*

As shown in Figure 62, the DAC output can be modeled as a pair of dc current sources that source a current of 17/32 × $I_{OUTFS}$ to each output. A differential ac current source, $I_{PEAK}$, is used to model the signal-dependent nature of the DAC output. The polarity and signal dependency of this ac current source are related to the digital code by the following equations:

$$F(Code) = (DACCODE - 8192)/8192 \quad (2)$$

$$-1 \le F(Code) < 1 \quad (3)$$

where *DACCODE* = 0 to 16,383 (decimal).

Because $I_{PEAK}$ can swing ±(15/32) × $I_{OUTFS}$, the output currents measured at IOUTP and IOUTN can span from $I_{OUTFS}/16$ to $I_{OUTFS}$. However, because the ac signal-dependent current component is complementary, the sum of the two outputs is always constant (that is, IOUTP + IOUTN = (34/32) × $I_{OUTFS}$).

The code-dependent current measured at the IOUTP (and IOUTN) output is as follows:

$$IOUTP = 17/32 \times I_{OUTFS} + 15/32 \times I_{OUTFS} \times F(Code) \quad (4)$$

$$IOUTN = 17/32 \times I_{OUTFS} - 15/32 \times I_{OUTFS} \times F(Code) \quad (5)$$

Figure 63 shows the IOUTP vs. DACCODE transfer function when $I_{OUTFS}$ is set to 19.65 mA.

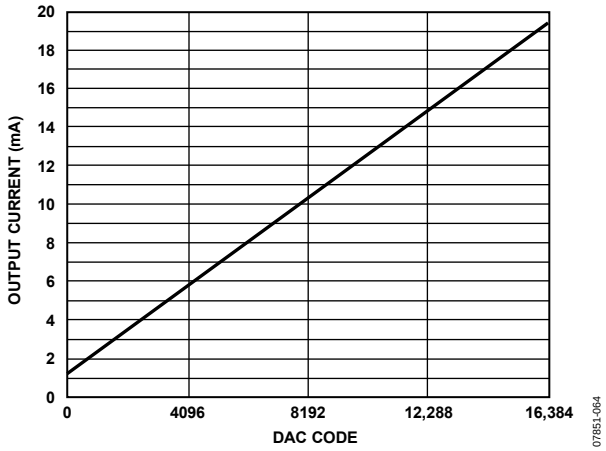



*Figure 63. Gain Curve for FSC[9:0] = 512, DAC Offset = 1.228 mA*

### Peak DAC Output Power Capability

The maximum peak power capability of a differential current output DAC is dependent on its peak differential ac current, $I_{PEAK}$, and the equivalent load resistance it sees. Because the AD9739 includes a differential 70 Ω resistance, it is best to use a doubly terminated external output network similar to what is shown in Figure 64. In this case, the equivalent load seen by the ac current source of the DAC is 25 Ω.

If the AD9739 is programmed for $I_{OUTFS}$ = 20 mA, its peak ac current is 9.375 mA and its peak power delivered to the equivalent load is 2.2 mW (that is, $P = I^2R$). Because the source and load resistance seen by the 1:1 balun are equal, this power is shared equally; therefore, the output load receives 1.1 mW or 0.4 dBm.

To calculate the rms power delivered to the load, the following must be considered:

- Peak-to-rms of the digital waveform
- Any digital backoff from digital full scale
- The DAC's sinc response and nonideal losses in external network

For example, a reconstructed sine wave with no digital backoff ideally measures −2.6 dBm because it has a peak-to-rms ratio of 3 dB. If a typical balun loss of 0.4 dBm is included, −3 dBm of actual power can be expected in the region where the sinc response of the DAC has negligible influence. Increasing the output power is best accomplished by increasing $I_{OUTFS}$, although any degradation in linearity performance must be considered acceptable for the target application.

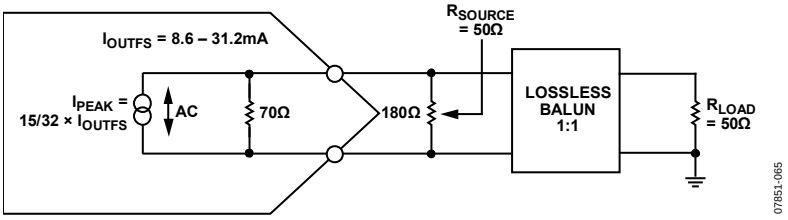



*Figure 64. Equivalent Circuit for Determining Maximum Peak Power to a 50 Ω Load*

### Output Stage Configuration

The AD9739 is intended to serve high dynamic range applications that require wide signal reconstruction bandwidth (that is, DOCSIS CMTS) and/or high IF/RF signal generation. Optimum ac performance can only be realized if the DAC output is configured for differential (that is, balanced) operation with its output common-mode voltage biased to analog ground. The output network used to interface to the DAC should provide a near 0 Ω dc bias path to analog ground. Any imbalance in the output impedance between the IOUTP and IOUTN pins results in asymmetrical signal swings that degrade the distortion performance (mostly even order) and noise performance. Component selection and layout are critical in realizing the performance potential of the AD9739.

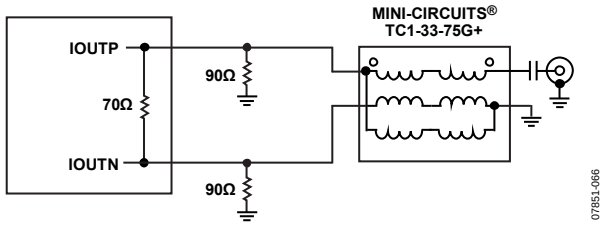



*Figure 65. Recommended Balun for Wideband Applications with Upper Bandwidths of up to 2.2 GHz*

Most applications requiring balanced-to-unbalanced conversion can take advantage of the Ruthroff 1:1 balun configuration shown in Figure 65. This configuration provides excellent amplitude/phase balance over a wide frequency range while providing a 0 Ω dc bias path to each DAC output. Also, its design provides exceptional bandwidth and can be considered for applications requiring signal reconstruction of up to 2.2 GHz. The characterization plots shown in this data sheet are based on the AD9739 evaluation board, which uses this configuration. Figure 66 compares the measured frequency response for normal and mix mode using the AD9739 evaluation board vs. the ideal frequency response.

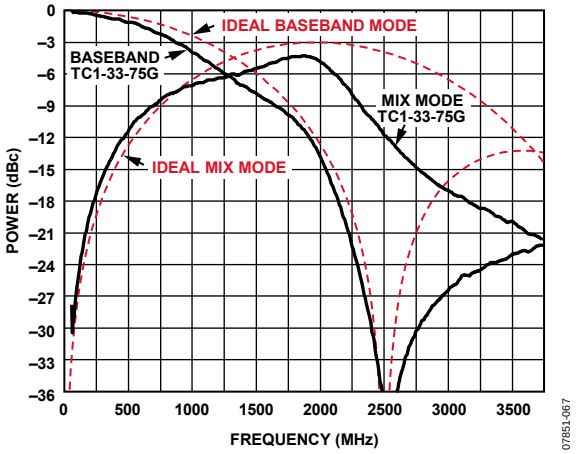



*Figure 66. Measured vs. Ideal Frequency Response for Normal (Baseband) and Mix Mode Operation Using a TC1-33-75G Transformer on the AD9739 Evaluation Board*

Figure 67 shows an interface that can be considered when interfacing the DAC output to a self-biased differential gain block. The inductors shown serve as RF chokes (L) that provide the dc bias path to analog ground. The value of the inductor, along with the dc blocking capacitors (C), determines the lower cutoff frequency of the composite pass-band response. An RF balun should also be considered before the RF differential gain stage and any filtering to ensure symmetrical common-mode impedance seen by the DAC output while suppressing any common-mode noise, harmonics, and clock spurs prior to amplification.

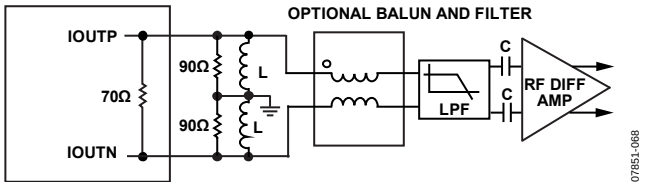



*Figure 67. Interfacing the DAC Output to the Self-Biased Differential Gain Stage*

For applications operating the AD9739 in mix mode with output frequencies extending beyond 2.2 GHz, the circuits shown in Figure 68 should be considered. The circuit in Figure 68 uses a wideband balun with a configuration similar to the one shown in Figure 67 to provide a dc bias path for the DAC outputs. The circuit in Figure 69 takes advantage of ceramic chip baluns to provide a dc bias path for the DAC outputs while providing excellent amplitude/phase balance over a narrower RF band. These low cost, low insertion loss baluns are available for different popular RF bands and provide excellent amplitude/phase balance over their specified frequency range.

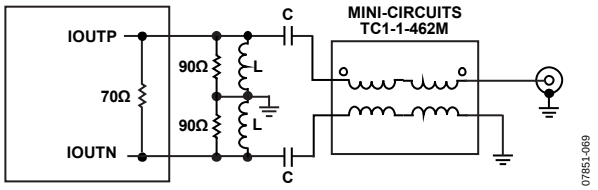



*Figure 68. Recommended Mix Mode Configuration Offering Extended RF Bandwidth Using a TC1-1-43A+ Balun*

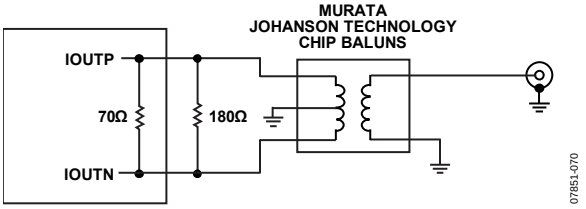



*Figure 69. Lowest Cost and Size Configuration for Narrow RF Band Operation*

## NONIDEAL SPECTRAL ARTIFACTS

The AD9739 output spectrum contains spectral artifacts that are not part of the original digital input waveform. These nonideal artifacts included harmonics (including alias harmonics), images, and clock spurs. Figure 70 shows a spectral plot of the AD9739 within the first Nyquist zone (that is, dc to $f_{DAC}/2$) reconstructing a 650 MHz, 0 dBFS sine wave at 2.4 GSPS. Besides the desired fundamental tone at the −7.8 dBm level, the spectrum also reveals these nonideal artifacts that also appear as spurs above the measurement noise floor. Because these nonideal artifacts are also evident in the second and third Nyquist zones during mix mode operation, the effects of these artifacts should also be considered when selecting the DAC clock rate for a target RF band.

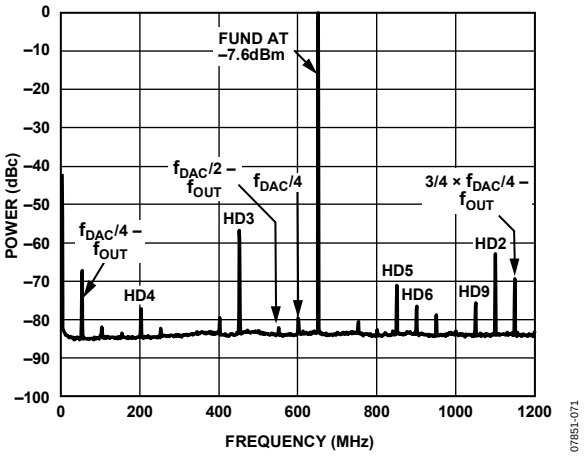



*Figure 70. Spectral Plot*

Note the following important observations pertaining to these nonideal spectral artifacts:

1. A full-scale sine wave (that is, single-tone) typically represents the worst-case condition because it has a peak-to-rms ratio of 3 dB and is unmodulated. Harmonics and aliased harmonics of a sine wave are easy to identify because they also appear as discrete spurs. Significant characterization of a high speed DAC is performed using single (or multitone) signals for this reason.
2. Modulated signals (that is, AM, PM, or FM) do not appear as spurs but rather as signals whose power spectral density is spread over a defined bandwidth determined by the modulation parameters of the signals. Any harmonics from the DAC spread over a wider bandwidth determined by the order of the harmonic and bandwidth of the modulated signal. For this reason, harmonics often appear as slight bumps in the measurement noise floor and can be difficult to discern.
3. Images appear as replicas of the original signal, therefore, can be easier to identify. In the case of the AD9739, internal modulation of the sampling clock at intervals related to $f_{DAC}/4$ generate image pairs at ¼ × $f_{DAC}$, ½ × $f_{DAC}$, and ¾ × $f_{DAC}$. Both upper and lower sideband images associated with ¼ × $f_{DAC}$ fall within the first Nyquist zone, while only the lower image of ½ × $f_{DAC}$ and ¾ × $f_{DAC}$ fall back. Note that the lower images appear frequency inverted. The difference in dBc between the fundamental and various images remains mostly signal independent because the mechanism causing these images is related to corruption of the sampling clock.
4. The magnitude of these images for a given device is dependent on several factors including DAC clock rate, output frequency, mu controller phase setting, and div-by-4 clock divider phase (Register 0x14, bit [7:6]. Table 30 shows how the magnitude of these images vary as the phase is varied for the case represented in Figure 70. Because the phase varies at power up, the image magnitude varies making it difficult to compensate digitally through a one-time factory calibration procedure. Also, the image magnitude can vary a few decibels over temperature and between devices due to process dependencies. (Note that the AD9739A is a viable option if factory calibration is considered acceptable for nonmultichip synchronization applications operating with clock rates in the 1.6 GSPS to 2.5 GSPS range).

**Table 30. Image Magnitude vs. Phase (PHZ) Setting**

| Image location | PHZ0 | PHZ1 | PHZ2 | PHZ3 |
|---|---|---|---|---|
| $f_{DAC}/4 - f_{OUT}$ | −70.2 | −71.4 | −72.2 | −77.1 |
| $f_{DAC}/2 - f_{OUT}$ | −80.2 | −71.3 | −69.9 | −74.9 |
| ¾ × $f_{DAC} - f_{OUT}$ | −69.9 | −72.5 | −73.4 | −73.7 |

5. A clock spur appears at $f_{DAC}/4$ and integer multiples of this frequency. Similar to images, the spur magnitude is also dependent on the same factors that cause variations in image levels. However, unlike images and harmonics, clock spurs always appear as discrete spurs, albeit their magnitude shows a slight dependency on the digital waveform and output frequency. Note that the clock spur appearing at $f_{DAC}/4$ can also be factory calibrated.
6. A large clock spur also appears at 2 × $f_{DAC}$ in either normal or mix mode operation. This clock spur is due to the quad switch DAC architecture causing switching events to occur on both edges of $f_{DAC}$.

## LAB EVALUATION OF THE AD9739

Figure 71 shows a recommended lab setup that was used to characterize the performance of the AD9739. The DPG2 is a dual port LVDS/CMOS data pattern generator available from Analog Devices, Inc., with an up to 1.25 GSPS data rate. The DPG2 directly interfaces to the AD9739 evaluation board via Tyco Z-PACK HM-Zd connectors. A low phase noise/jitter RF source, such as an R&S SMA100A signal generator, is used for the DAC clock. A 5 V power supply is used to power up the AD9739 evaluation board, and SMA cabling is used to interface to the supply, clock source, and spectrum analyzer. A USB 2.0 interface to a host PC is used to communicate to both the AD9739 evaluation board and the DPG2.

A high dynamic range spectrum analyzer is required to evaluate the AD9739 reconstructed waveform's ac performance. This is especially the case when measuring ACLR performance for high dynamic range applications, such as multicarrier DOCSIS CMTS applications. Harmonic, SFDR, and IMD measurements pertaining to unmodulated carriers can benefit by using a sufficiently high RF attenuation setting because these artifacts are easy to identify above the spectrum analyzer noise floor. However, reconstructed waveforms having modulated carrier(s) often benefit from the use of a high dynamic range RF amplifier and/or passive filters to measure close-in and wideband ACLR performance when using spectrum analyzers of limited dynamic range.

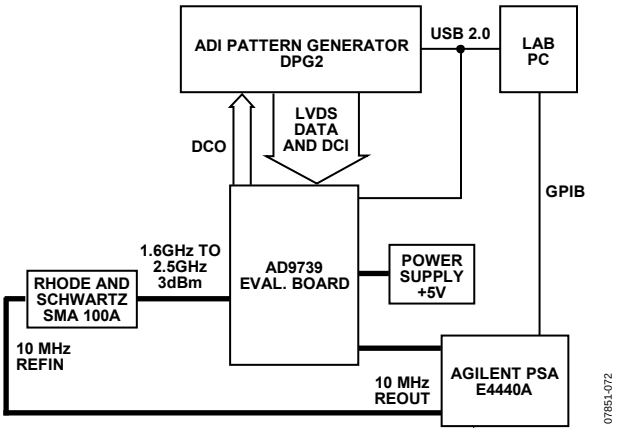



*Figure 71. Lab Test Setup Used to Characterize the AD9739*

## POWER DISSIPATION AND SUPPLY DOMAINS

The power dissipation of the AD9739 is dependent on the DAC clock rate as shown in Figure 72 and Figure 73. The current consumption from the 3.3 V supply remains relatively constant because it is used for biasing the DAC core (that is, VDDA) and differential input receivers (that is, VDD33). However, the current consumption from the 1.8 V supply is clock rate dependent and increases linearly with frequency because this supply is used by the digital path (that is, VDD) as well as the clock distribution circuitry (that is, VDDC).

Treat the VDDC supply as an analog supply because the clock distribution circuitry has poor power supply rejection; therefore, noise on this supply can induce clock jitter. To ensure low noise on this sensitive supply, use a separate 1.8 V regulator powered from the 3.3 V analog supply rail that is also used to power VDDA. This supply rail can also be used to power-up VDD33 via an LC filter network. The digital 1.8 V supply, VDD, can be supplied via a well-filtered switching regulator.

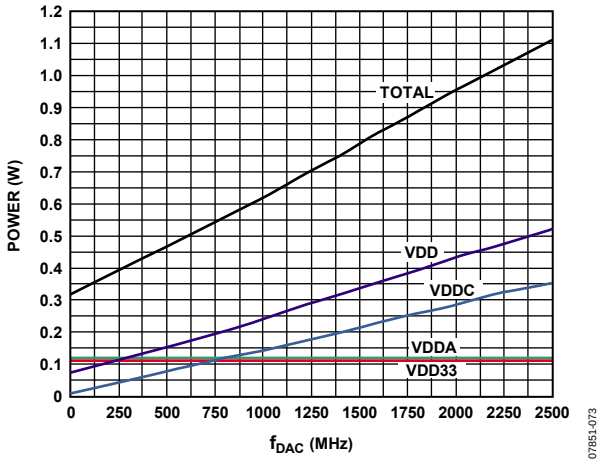



*Figure 72. Power Consumption vs. $f_{DAC}$ @ 25°C*

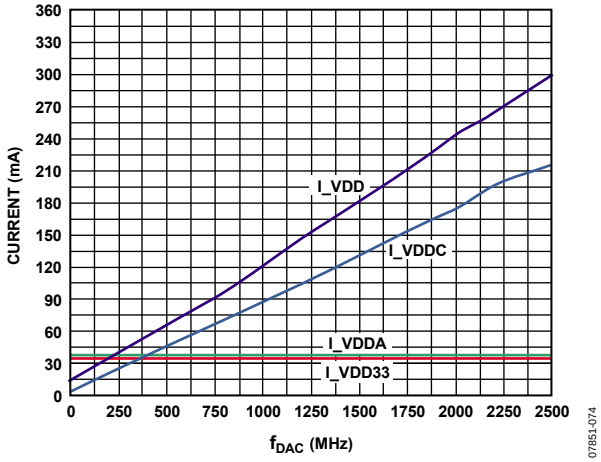



*Figure 73. Current Consumption vs. $f_{DAC}$ @ 25°C*

## RECOMMENDED START-UP SEQUENCE

Upon power-up of the AD9739, a host processor is required to initialize and configure the AD9739 via its SPI port. Figure 74 shows a flowchart of the sequential steps required, while Table 31 and Table 32 provides more detail on the SPI register write/read operations required to implement the flowchart steps. Note the following:

- A software reset is optional because the AD9739 has both an internal POR circuit and a RESET pin.
- The SYNC controller is optional because it is only required to synchronize two or more devices. If synchronization is required, validate that DCI_DEL values between devices are sufficiently matched.
- The mu controller must be first enabled (and in track mode) before the data receiver controller is enabled because the DCO output signal is derived from this circuitry.
- A wait period is related to $f_{DATA}$ periods.
- Limit the number of attempts to lock the controllers to three; locks typically occur on the first attempt.
- Hardware or software interrupts can be used to monitor the status of the controllers.

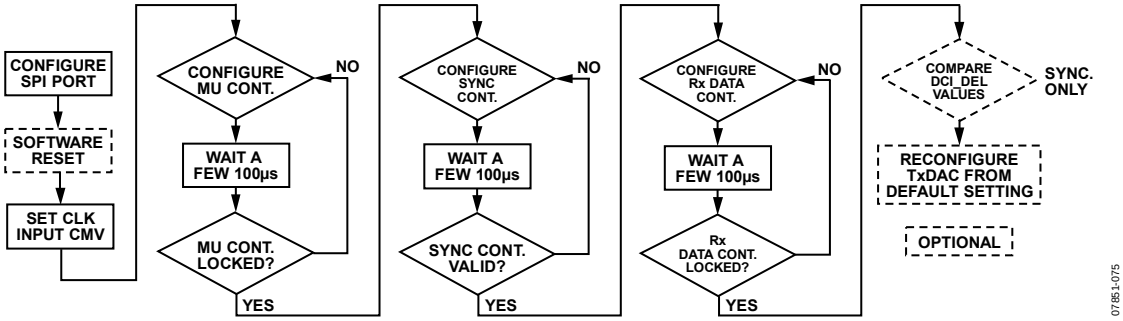



*Figure 74. Flowchart for Initialization and Configuration of the AD9739*

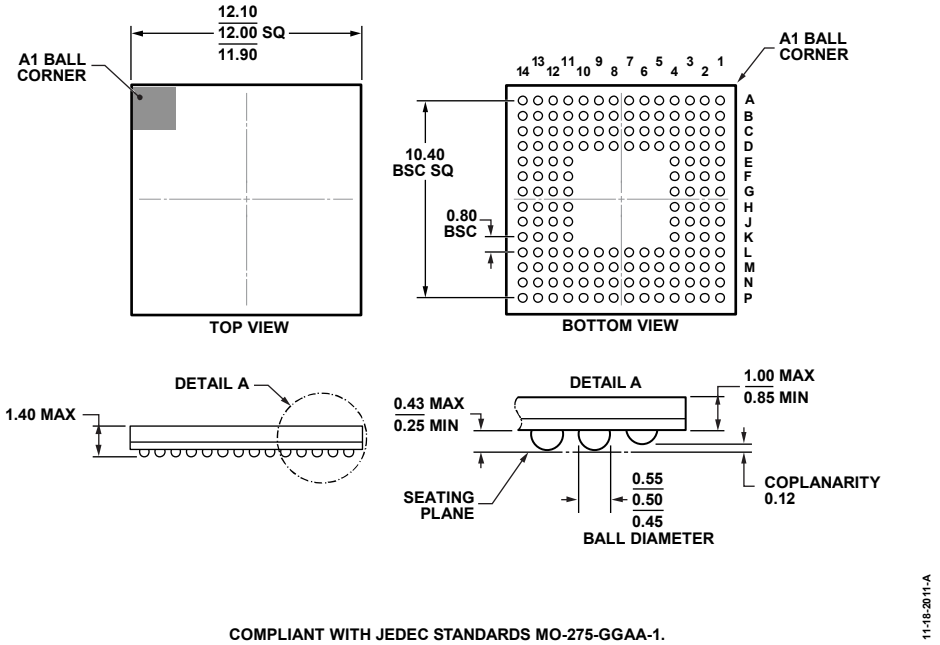
**Table 31. Recommended SPI Initialization with SYNC Controller Disabled**

| Step | Address (Hex) | Write Value | Comments |
|---|---|---|---|
| 1 | 0x00 | 0x00 | Configure for the 4-wire SPI mode with MSB. Note that Bits[7:5] must be mirrored onto Bits[2:0] because the MSB/LSB format can be unknown at power-up. |
| 2 | 0x00 | 0x20 | Software reset to default SPI values. |
| 3 | 0x00 | 0x00 | Clear the reset bit. |
| 4 | 0x22 | 0x0F | Set the common-mode voltage of DACCLK_P and DACCLK_N inputs. |
| 5 | 0x23 | 0x0F | |
| 6 | 0x24 | 0x30 | Configure the mu controller. Refer to Table 28 for recommended target mu slope and phase settings vs. clock rate. |
| 7 | 0x25 | 0x80 | |
| 8 | 0x27 | 0x44 | |
| 9 | 0x28 | 0x6C | |
| 10 | 0x29 | 0xCB | |
| 11 | 0x26 | 0x02 | |
| 12 | 0x26 | 0x03 | Enable the mu controller search and track mode. |
| 13 | Not applicable | Not applicable | Wait for 160 K × 1/$f_{DATA}$ cycles. |
| 14 | 0x2A | | Read back Register 0x2A and confirm that it is equal to 0x01 to ensure that the DLL loop is locked. If it is not locked, proceed to Step 10 and repeat. Limit attempts to three before breaking out of the loop and reporting a mu lock failure. |
| 15 | Not applicable | Not applicable | Ensure that the AD9739 is fed with DCI clock input from the data source. |
| 16 | 0x13 | 0x72 | Set FINE_DEL_SKEW to 2. |
| 17 | 0x10 | 0x00 | Disable the data Rx controller before enabling it. |
| 18 | 0x10 | 0x02 | Enable the data Rx controller for loop and IRQ. |
| 19 | 0x10 | 0x03 | Enable the data Rx controller for search and track mode. |
| 20 | Not applicable | Not applicable | Wait for 135 K × 1/$f_{DATA}$ cycles. |
| 21 | 0x21 | | Read back Register 0x21 and confirm that it is equal to 0x09 to ensure that the DLL loop is locked and tracking. If it is not locked and tracking, advance the CLKDIVPH[1:0] phase in Register 0x14, Bit[7:6] before proceeding to Step 17 to repeat attempt. Limit attempts to three before breaking out of the loop and reporting an Rx data lock failure. |
| 22 | 0x06, 0x07 | 0x00, 0x02 | Optional: modify the TxDAC $I_{OUTFS}$ setting (the default is 20 mA). |
| 23 | 0x08 | 0x00 | Optional: modify the TxDAC operation mode (the default is normal mode). |

**Table 32. Recommended SPI Initialization with SYNC Controller Enabled**

| Step | Address (Hex) | Write Value | Comments |
|---|---|---|---|
| 1 | 0x00 | 0x00 | Configure for the 4-wire SPI mode with MSB. Note that Bits[7:5] must be mirrored onto Bits[2:0] because the MSB/LSB format can be unknown at power-up. |
| 2 | 0x00 | 0x20 | Software reset to default SPI values. |
| 3 | 0x00 | 0x00 | Clear the reset bit. |
| 4 | 0x22 | 0x0F | Set the common-mode voltage of DACCLK_P and DACCLK_N inputs. |
| 5 | 0x23 | 0x0F | |
| 6 | 0x24 | 0x30 | Configure the mu controller. Refer to Table 28 for recommended target Mu slope and phase settings vs. clock rate. |
| 7 | 0x25 | 0x80 | |
| 8 | 0x27 | 0x44 | |
| 9 | 0x28 | 0x6C | |
| 10 | 0x29 | 0xCB | |
| 11 | 0x26 | 0x02 | |
| 12 | 0x26 | 0x03 | Enable the mu controller search and track mode. |
| 13 | Not applicable | Not applicable | Wait for 160 K × 1/$f_{DATA}$ cycles. |
| 14 | 0x2A | | Read back Register 0x2A and confirm that it is equal to 0x01 to ensure that the DLL loop is locked. If it is not locked, proceed to Step 10 and repeat. Limit attempts to three before breaking out of the loop and reporting a mu lock failure. |
| 15 | 0x15 | 0x42 | Configure sync controller. |
| 16 | 0x10 | 0x00 | Disable sync controller before enabling it. |
| 17 | 0x10 | 0x60 or 0x40 | Enable sync controller for loop and IRQ.<br>0x60 = master mode.<br>0x40 = slave mode. |
| 18 | 0x10 | 0x70 or 0x50 | Enable sync controller:<br>0x70 = master mode.<br>0x50 = slave mode. |
| 19 | Not applicable | Not applicable | Wait for 160 K × 1/$f_{DATA}$ for DLL to lock. |
| 20 | 0x21 | | Read back Register 0x21 to confirm proper operation:<br>0x90 = master mode.<br>0x00 = slave mode.<br>If not, proceed to Step 15 and repeat. Limit to three attempts before breaking out of loop and reporting sync lock failure. |
| 21 | 0x0D | | Read back Register 0x0D and confirm Bits[5:4] = 10. If not, proceed to Step 2 and repeat. Limit to three attempts before breaking out of loop and reporting sync lock failure. |
| 22 | Not applicable | Not applicable | Ensure that the AD9739 is fed with DCI clock input from the data source. |
| 23 | 0x13 | 0x72 | Set FINE_DEL_SKEW to 2. |
| 24 | 0x10 | 0x70 or 0x50 | Disable the data Rx controller before enabling it.<br>0x70 = master mode.<br>0x50 = slave mode. |
| 25 | 0x10 | 0x72 or 0x52 | Enable the data Rx controller for loop and IRQ.<br>0x72 = master mode.<br>0x52 = slave mode. |
| 26 | 0x10 | 0x73 or 0x53 | Enable the data Rx controller for search and track mode.<br>0x73 = master mode.<br>0x53 = slave mode. |
| 27 | | | Wait for 135 K × 1/$f_{DATA}$ cycles. |
| 28 | 0x21 | | Read back Register 0x21 and confirm that it is equal to 0x09 to ensure that the DLL loop is locked and tracking. If it is not locked and tracking, proceed to Step 16 and repeat. Limit attempts to three before breaking out of the loop and reporting an Rx data lock failure. |
| 29 | Not applicable | Not applicable | Readback DCI_DEL value in Register 0x13 and Register 0x14 for master and slave. If slave devices are not within 40 codes of each other, re-specify target DCI_DEL value to be average between master and readback DCI_DEL value. |
| 30 | 0x06, 0x07 | 0x00, 0x02 | Optional: modify the TxDAC $I_{OUTFS}$ setting (the default is 20 mA). |
| 31 | 0x08 | 0x00 | Optional: modify the TxDAC operation mode (the default is normal mode). |

# OUTLINE DIMENSIONS




COMPLIANT WITH JEDEC STANDARDS MO-275-GGAA-1.

*Figure 75. 160-Ball Chip Scale Package Ball Grid Array [CSP_BGA]*
*(BC-160-1)*
*Dimensions shown in millimeters*

## ORDERING GUIDE

| Model[1] | Temperature Range | Package Description | Package Option |
|---|---|---|---|
| AD9739BBCZ | −40°C to +85°C | 160-Ball Chip Scale Package Ball Grid Array [CSP_BGA] | BC-160-1 |
| AD9739BBCZRL | −40°C to +85°C | 160-Ball Chip Scale Package Ball Grid Array [CSP_BGA] | BC-160-1 |
| AD9739BBC | −40°C to +85°C | 160-Ball Chip Scale Package Ball Grid Array [CSP_BGA] | BC-160-1 |
| AD9739BBCRL | −40°C to +85°C | 160-Ball Chip Scale Package Ball Grid Array [CSP_BGA] | BC-160-1 |
| AD9739-R2-EBZ | | Evaluation Board | |

[1] Z = RoHS Compliant Part.