

FEATURES

Dual DPLL synchronizes 2 kHz to 750 MHz physical layer clocks providing frequency translation with jitter cleaning of noisy references

Complies with ITU-T G.8262 and Telcordia GR-253

Supports Telcordia GR-1244, ITU-T G.812, G.813, G.823, G.824, and G.825

Continuous frequency monitoring and reference validation for frequency deviation as low as 50 ppb

Both DPLLs feature a 24-bit fractional divider with 24-bit programmable modulus

Programmable digital loop filter bandwidth: 10^{-4} Hz to 1850 Hz

Automatic and manual holdover and reference switchover, providing zero delay, hitless, or phase buildout operation

Programmable priority-based reference switching with manual, automatic revertive, and automatic nonrevertive modes supported

5 pairs of clock output pins with each pair useable as differential LVDS/HCSL/CML or as 2 single-ended outputs (1 Hz to 500 MHz)

2 differential or 4 single-ended input references

Crosspoint mux interconnects reference inputs to PLLs

Supports embedded (modulated) input/output clock signals

Fast DPLL locking modes

Provides internal capability to combine the low phase noise of a crystal resonator or crystal oscillator with the frequency stability and accuracy of a TCXO or OCXO

External EEPROM support for autonomous initialization

Single 1.8 V power supply operation with internal regulation

Built in temperature monitor/alarm and temperature compensation for enhanced zero delay performance

APPLICATIONS

SyncE jitter cleanup and synchronization

Optical transport networks (OTN), SDH, and macro and small cell base stations

OTN mapping/demapping with jitter cleaning

Small base station clocking, including baseband and radio

Stratum 2, Stratum 3e, and Stratum 3 holdover, jitter cleanup, and phase transient control

JESD204B support for analog-to-digital converter (ADC) and digital-to-analog converter (DAC) clocking

Cable infrastructures

Carrier Ethernet

GENERAL DESCRIPTION

The 10 clock outputs of the AD9542 are synchronized to any one of up to four input references. The digital phase-locked loops (DPLLs) reduce timing jitter associated with the external references. The digitally controlled loop and holdover circuitry continuously generate a low jitter output signal, even when all reference inputs fail.

The AD9542 is available in a 48-lead LFCSP (7 mm × 7 mm) package and operates over the -40°C to $+85^{\circ}\text{C}$ temperature range.

Note that throughout this data sheet, multifunction pins, such as SDO/M5, are referred to either by the entire pin name or by a single function of the pin, for example, M5, when only that function is relevant.

TABLE OF CONTENTS

| | | | |
|---|----|--|----|
| Features | 1 | System Clock PLL..... | 29 |
| Applications..... | 1 | System Clock Input Frequency Declaration | 29 |
| General Description | 1 | System Clock Source..... | 29 |
| Revision History | 3 | 2× Frequency Multiplier..... | 29 |
| Functional Block Diagram | 4 | Prescale Divider..... | 29 |
| Specifications..... | 5 | Feedback Divider..... | 30 |
| Supply Voltage..... | 5 | System Clock PLL Output Frequency | 30 |
| Supply Current..... | 5 | System Clock PLL Lock Detector..... | 30 |
| Power Dissipation..... | 5 | System Clock Stability Timer..... | 30 |
| System Clock Inputs, XOA and XOB..... | 6 | System Clock Input Termination Recommendations | 30 |
| Reference Inputs | 7 | Digital PLL (DPLL)..... | 31 |
| Reference Monitors | 8 | Overview | 31 |
| DPLL Phase Characteristics..... | 8 | DPLL Phase/Frequency Lock Detectors | 31 |
| Distribution Clock Outputs | 9 | DPLL Loop Controller..... | 31 |
| Time Duration of Digital Functions | 10 | Applications Information | 32 |
| Digital PLL (DPLL0, DPLL1) Specifications | 10 | Optical Networking Line Card | 32 |
| Digital PLL Lock Detection Specifications | 11 | Small Cell Base Station | 33 |
| Holdover Specifications..... | 11 | Initialization Sequence..... | 34 |
| Analog PLL (APLL0, APLL1) Specifications..... | 11 | Status and Control Pins | 37 |
| Output Channel Divider Specifications | 11 | Multifunction Pins at Reset/Power-Up | 37 |
| System Clock Compensation Specifications..... | 12 | Status Functionality..... | 38 |
| Temperature Sensor Specifications | 12 | Control Functionality | 38 |
| Serial Port Specifications | 12 | Interrupt Request (IRQ)..... | 43 |
| Logic Input Specifications (RESETB, M0 to M6) | 14 | IRQ Monitor | 43 |
| Logic Output Specifications (M0 to M6) | 14 | IRQ Mask..... | 43 |
| Jitter Generation (Random Jitter) | 14 | IRQ Clear..... | 43 |
| Phase Noise | 15 | Watchdog Timer | 45 |
| Absolute Maximum Ratings..... | 18 | Lock Detectors | 46 |
| Thermal Resistance | 18 | DPLL Lock Detectors | 46 |
| ESD Caution..... | 18 | Phase Step Detector..... | 48 |
| Pin Configuration and Function Descriptions..... | 19 | Phase Step Limit | 48 |
| Typical Performance Characteristics | 21 | Skew Adjustment..... | 49 |
| Terminology | 25 | EEPROM Usage..... | 50 |
| Theory of Operation | 26 | Overview | 50 |
| Overview..... | 26 | EEPROM Controller General Operation..... | 50 |
| Reference Input Physical Connections..... | 26 | EEPROM Instruction Set | 51 |
| Input/Output Termination Recommendations | 27 | Multidevice Support..... | 53 |
| System Clock Inputs..... | 27 | Serial Control Port | 55 |
| Reference Clock Inputs..... | 27 | SPI/I ² C Port Selection..... | 55 |
| Clock Outputs | 28 | SPI Serial Port Operation..... | 55 |

I²C Serial Port Operation.....58 Ordering Guide61
Outline Dimensions.....61

REVISION HISTORY

9/2017—Revision 0: Initial Version

FUNCTIONAL BLOCK DIAGRAM

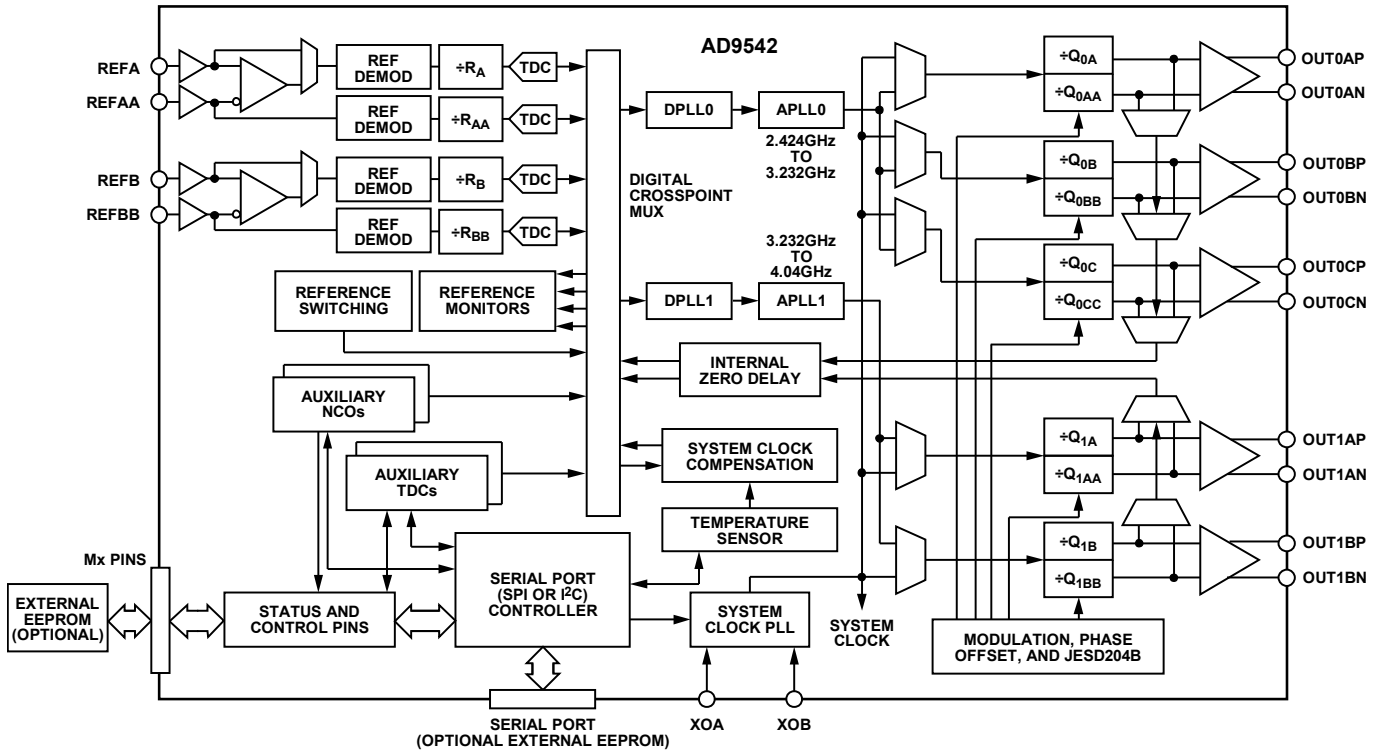


Figure 1.

1592E-001

SPECIFICATIONS

The minimum and maximum values apply for the full range of supply voltage and operating temperature variations. The typical values apply for VDD = 1.8 V and T_A = 25°C, unless otherwise noted.

SUPPLY VOLTAGE

Table 1.

| Parameter | Min | Typ | Max | Unit | Test Conditions/Comments |
|----------------|------|-----|-------|------|---|
| SUPPLY VOLTAGE | | | | | |
| VDDIOA, VDDIOB | 1.71 | 1.8 | 3.465 | V | 1.8 V, 2.5 V, and 3.3 V operation supported |
| VDD | 1.71 | 1.8 | 1.89 | V | |

SUPPLY CURRENT

The maximum supply voltage values given in Table 1 are the basis for the maximum supply current specifications. The typical supply voltage values given in Table 1 are the basis for the typical supply current specifications. The minimum supply voltage values given in Table 1 are the basis for the minimum supply current specifications.

Table 2.

| Parameter | Min | Typ | Max | Unit | Test Conditions/Comments |
|---|-----|-----|-----|------|--|
| SUPPLY CURRENT FOR TYPICAL CONFIGURATION | | | | | The Typical Configuration specification in Table 3 is the basis for the values shown in this section |
| I _{VDDIOx} | | 5 | 8 | mA | Aggregate current for all VDDIOx pins (where x = A or B) |
| I _{VDD} | 260 | 310 | 355 | mA | Aggregate current for all VDD pins |
| SUPPLY CURRENT FOR ALL BLOCKS RUNNING CONFIGURATION | | | | | The All Blocks Running condition in Table 3 is the basis for the values shown in this section |
| I _{VDDIOx} | | 5 | 8 | mA | Aggregate current for all VDDIOx pins (where x = A or B) |
| I _{VDD} | 321 | 390 | 430 | mA | Aggregate current for all VDD pins |

POWER DISSIPATION

The typical values apply for VDD = 1.8 V, and the maximum values apply for VDD = 1.89 V.

Table 3.

| Parameter | Min | Typ | Max | Unit | Test Conditions/Comments |
|---|-----|-----|-----|------|--|
| POWER DISSIPATION | | | | | |
| Typical Configuration | 445 | 560 | 671 | mW | System clock = 49.152 MHz crystal; two DPLLs active; two 19.44 MHz input references in differential mode; two ac-coupled PLL0 CML output drivers at 245.76 MHz; and 2 PLL1 CML output drivers at 156.25 MHz |
| All Blocks Running | 548 | 700 | 813 | mW | System clock = 49.152 MHz crystal; two DPLLs active; two 19.44 MHz input references in differential mode; 3 ac-coupled PLL0 HCSL output drivers at 400 MHz; and two PLL1 HCSL output drivers at 400 MHz |
| Full Power-Down | | 125 | | mW | Based on the Typical Configuration specification with the power down all bit set to Logic 1 |
| Incremental Power Dissipation | | | | | Based on the Typical Configuration specification; the values in this section indicate the change in power due to the indicated operation relative to the Typical Configuration specification |
| Complete DPLL/APLL On/Off | | 200 | | mW | Change in dissipated power relative to the Typical Configuration specification; the blocks, powered down, consist of 1 reference input, 1 DPLL, 1 APLL, 2 channel dividers, and 2 output drivers |
| Incremental Power Dissipation Complete DPLL/APLL On/Off | | 200 | | mW | Based on the Typical Configuration specification; the values in this section indicate the change in power due to the indicated operation relative to the Typical Configuration specification; the blocks, powered down, consist of one reference input, one DPLL, one APLL, two channel dividers, and two output drivers |

| Parameter | Min | Typ | Max | Unit | Test Conditions/Comments |
|-----------------------------------|-----|-----|-----|------|--------------------------|
| Input Reference On/Off | | | | | |
| Differential (Normal Mode) | | 20 | | mW | $f_{REF} = 19.44$ MHz |
| Differential (DC-Coupled LVDS) | | 21 | | mW | $f_{REF} = 19.44$ MHz |
| Single-Ended | | 13 | | mW | $f_{REF} = 19.44$ MHz |
| Output Distribution Driver On/Off | | | | | At 156.25 MHz |
| 15 mA Mode | | 30 | | mW | |
| 12 mA Mode | | 23 | | mW | |
| 7.5 mA Mode | | 15 | | mW | |
| Auxiliary DPLL On/Off | | 1 | | mW | |

SYSTEM CLOCK INPUTS, XOA AND XOB

Table 4.

| Parameter | Min | Typ | Max | Unit | Test Conditions/Comments |
|--|------|------|------|------------|--|
| SYSTEM CLOCK MULTIPLIER | | | | | |
| Output Frequency Range | 2250 | | 2415 | MHz | The frequency range of the internal voltage controlled oscillator (VCO) places limits on the choice of the system clock input frequency |
| Phase Frequency Detector (PFD) Rate | 20 | | 300 | MHz | |
| SYSTEM CLOCK REFERENCE INPUT PATH | | | | | System clock input must be ac-coupled |
| Input Frequency Range | | | | | |
| System Clock Input Doubler Disabled | 20 | | 300 | MHz | Support of oven controlled crystal oscillators (OCXOs) < 20 MHz is possible using the auxiliary DPLL for system clock frequency compensation |
| Enabled | 16 | | 150 | MHz | |
| Self Biased Common-Mode Voltage Input Voltage | | 0.75 | | V | Internally generated |
| High | 0.9 | | | V | For dc-coupled, single-ended operation |
| Low | | | 0.5 | V | For dc-coupled, single-ended operation |
| Differential Input Voltage Sensitivity | 250 | | | mV p-p | Minimum voltage swing required (as measured with a differential probe) across the XOA/XOB pins to ensure switching between logic states; the instantaneous voltage on either pin must not exceed 1.2V; accommodate the single-ended input by ac grounding the complementary input; 800 mV p-p recommended for optimal jitter performance |
| Slew Rate for Sinusoidal Input | 50 | | | V/ μ s | Minimum input slew rate for device operation; oscillators with square wave outputs are recommended if not using a crystal |
| System Clock Input Divider (J Divider) Frequency | 100 | | | MHz | |
| System Clock Input Doubler Duty Cycle | | | | | Tolerable duty cycle variation on the system clock input when using the frequency doubler |
| 20 MHz to 150 MHz | 43 | 50 | 57 | % | |
| 16 MHz to 20 MHz | 47 | 50 | 53 | % | |
| Input Resistance | | 5 | | k Ω | |
| QUARTZ CRYSTAL RESONATOR PATH | | | | | |
| Resonator Frequency Range | 25 | | 60 | MHz | Fundamental mode, AT cut crystal |
| Crystal Motional Resistance | | | 100 | Ω | A maximum motional resistance of 50 Ω , and maximum C_{LOAD} of 8 pF is strongly recommended for crystals >52 MHz |

REFERENCE INPUTS

Table 5.

| Parameter | Min | Typ | Max | Unit | Test Conditions/Comments |
|---|-----------------------|------|-----------------------|------------|--|
| DIFFERENTIAL MODE | | | | | |
| Differential mode specifications assume ac coupling of the input signal to the reference input pins | | | | | |
| Frequency Range | | | 750 | MHz | Lower limit dependent on input slew rate |
| Sinusoidal Input | | | 750 | Hz | Lower limit dependent on ac coupling |
| LVPECL Input | 2000 | | 750×10^6 | Hz | Assumes an LVDS minimum of 494 mV p-p differential amplitude; lower limit dependent on ac coupling |
| LVDS Input | 2000 | | 500×10^6 | Hz | Assumes an LVDS minimum of 494 mV p-p differential amplitude; lower limit dependent on ac coupling |
| Slew Rate for Sinusoidal input | 20 | | | V/ μ s | Minimum input slew rate for device operation; jitter degradation may occur for slew rates < 35 V/ μ s |
| Common-Mode Input Voltage | | 0.64 | | V | Internally generated self bias voltage |
| Differential Input Amplitude | | | | | Peak-to-peak differential voltage swing across pins required to ensure switching between logic levels as measured with a differential probe; instantaneous voltage on either pin must not exceed 1.3 V |
| $f_{IN} < 500$ MHz | 350 | | 2100 | mV p-p | |
| $f_{IN} = 500$ MHz to 750 MHz | 500 | | 2100 | mV p-p | |
| Differential Input Voltage Hysteresis | | 55 | 100 | mV | |
| Input Resistance | | 16 | | k Ω | Equivalent differential input resistance |
| Input Pulse Width | | | | | |
| LVPECL | 600 | | | ps | |
| LVDS | 900 | | | ps | |
| DC-COUPLED, LVDS-COMPATIBLE MODE | | | | | |
| Applies for dc-coupling to an LVDS source | | | | | |
| Frequency Range | 2000 | | 450×10^6 | Hz | |
| Common-Mode Input Voltage | 1.125 | | 1.375 | V | |
| Differential Input Amplitude | 400 | | 1200 | mV p-p | Differential voltage across pins required to ensure switching between logic levels; instantaneous voltage on either pin must not exceed the supply rails |
| Differential Input Voltage Hysteresis | | 55 | 100 | mV | |
| Input Resistance | | 16 | | k Ω | |
| Input Pulse Width | 1 | | | ns | |
| SINGLE-ENDED MODE | | | | | |
| Single-ended mode specifications assume dc coupling of the input signal to the reference input pins | | | | | |
| Frequency Range | | | | | |
| 1.2 V AC-Coupled | 2000 | | 500×10^6 | Hz | Lower limit dependent on ac-coupling |
| 1.2 V and 1.8 V CMOS | 2000 | | 500×10^6 | Hz | CMOS specifications assume dc coupling of the input signal to the reference input pins |
| 1.2 V AC-Coupled Common-Mode Voltage | | 610 | | mV | Internally generated self-bias voltage |
| Input Amplitude (Single-Ended, AC-Coupled Mode) | 360 | | 1200 | mV p-p | Peak-to-peak single-ended voltage swing; instantaneous voltage must not exceed 1.3 V |
| 1.2 V and 1.8 V CMOS Input Voltage | | | | | |
| High, V_{IH} | $0.65 \times V_{REF}$ | | $1.15 \times V_{REF}$ | V | V_{REF} is determined by operating mode of the CMOS input receiver, 1.2 V or 1.8 V |
| Low, V_{IL} | | | $0.35 \times V_{REF}$ | V | |
| Input Resistance | | | | | |
| DC-Coupled Single-Ended Mode | | 30 | | k Ω | |
| AC-Coupled Single-Ended Mode | | 15 | | k Ω | |
| Input Pulse Width | 900 | | | ps | |

REFERENCE MONITORS

Table 6.

| Parameter | Min | Typ | Max | Unit | Test Conditions/Comments |
|----------------------------------|--------------------|------------------------------------|-------|---------------|--|
| REFERENCE MONITORS | | | | | |
| Reference Monitor | | | | | |
| Loss of Reference Detection Time | | $4.9 + 0.13 \times t_{\text{PFD}}$ | | μs | t_{PFD} is the nominal phase detector period, R/f_{REF} , where R is the frequency division factor determined by the R divider, and f_{REF} is the frequency of the active reference |
| Frequency Out of Range Limits | 5×10^{-8} | | 0.015 | ppb | Parts per billion (ppb) is defined as $\Delta f/f_{\text{REF}}$, where Δf is the frequency deviation, and f_{REF} is the reference input frequency; programmable with the lower bound, subject to quality of the system clock (or the source of system clock compensation) |
| Validation Timer | 0.001 | | 1048 | sec | Programmable in 1 ms increments |
| Excess Jitter Alarm Threshold | 1 | | 65535 | ns | Programmable in 1 ns increments |

DPLL PHASE CHARACTERISTICS

Table 7.

| Parameter | Min | Typ | Max | Unit | Test Conditions/Comments |
|-----------------------------------|-------|----------|-----------|--------------------------|---|
| MAXIMUM OUTPUT PHASE PERTURBATION | | | | | |
| Phase Refinement Disabled | | | | | Assumes a jitter free reference; satisfies Telcordia GR-1244-CORE requirements; 0 ppm frequency difference between references; reference switch initiated via register map (see the AD9542 Register Map Reference Manual) by faulting the active reference input |
| Peak | | ± 20 | ± 140 | ps | 50 Hz DPLL loop bandwidth; normal phase margin mode; frequency translation = 19.44 MHz to 155.52 MHz; 49.152 MHz signal generator used for system clock source |
| Steady State | | | | | |
| Phase Buildout Operation | | ± 18 | ± 125 | ps | |
| Hitless Operation | | 0 | | ps | |
| Phase Refinement Enabled | | | | | 50 Hz DPLL loop bandwidth; high phase margin mode; phase refinement iterations = 4; frequency translation = 19.44 MHz to 155.52 MHz; 49.152 MHz signal generator used for system clock source |
| Peak | | ± 5 | ± 40 | ps | |
| Steady State | | | | | |
| Phase Buildout Operation | | ± 4 | ± 35 | ps | |
| Hitless Operation | | 0 | | ps | |
| PHASE SLEW LIMITER | 0.001 | | 250 | $\mu\text{s}/\text{sec}$ | See the AN-1420 Application Note, Phase Buildout and Hitless Switchover with Digital Phase-Locked Loops (DPLLs) |

DISTRIBUTION CLOCK OUTPUTS

Table 8.

| Parameter | Min | Typ | Max | Unit | Test Conditions/Comments |
|-----------------------------------|-----------|-----------|-------------------|------|--|
| DIFFERENTIAL MODE | | | | | |
| Output Frequency | | | | | All testing is both ac-coupled and dc-coupled Frequency range determined by driver functionality; actual frequency synthesis may be limited by the APLL VCO frequency range |
| CML | 1 | | 500×10^6 | Hz | Terminated per Figure 33 |
| HCSL | 1 | | 500×10^6 | Hz | Terminated per Figure 32 |
| Differential Output Voltage Swing | | | | | Voltage between output pins measured with output driver static; peak-to-peak differential output amplitude is twice that shown when driver is toggling and measured using a differential probe |
| Output Current = 7.5 mA | | | | | |
| HCSL | 312 | 368 | 402 | mV | Terminated per Figure 32 |
| CML | 257 | 348 | 408 | mV | Terminated to VDD (nominal 1.8 V) per Figure 33 |
| Output Current = 15 mA | | | | | |
| HCSL | 631 | 745 | 809 | mV | Terminated per Figure 32 |
| CML | 578 | 729 | 818 | mV | Terminated to VDD (nominal 1.8 V) per Figure 33 |
| Common-Mode Output Voltage | | | | | |
| Output Current = 7.5 mA | | | | | |
| HCSL | 155 | 184 | 201 | mV | Terminated per Figure 32 |
| CML | VDD – 208 | VDD – 188 | VDD – 169 | mV | Terminated to VDD (nominal 1.8 V) per Figure 33 (maximum common-mode voltage case occurs at the minimum amplitude) |
| Output Current = 15 mA | | | | | |
| HCSL | 316 | 372 | 405 | mV | Terminated per Figure 32 |
| CML | VDD – 416 | VDD – 371 | VDD – 327 | mV | Terminated to VDD (nominal 1.8 V) per Figure 33 (maximum common-mode voltage case occurs at the minimum amplitude) |
| SINGLE-ENDED MODE | | | | | |
| Output Frequency | 1 | | 500×10^6 | Hz | Frequency range determined by driver functionality; actual frequency synthesis may be limited by the APLL VCO frequency range |
| Output Current = 12 mA | | | | | |
| Voltage Swing (Peak-to-Peak) | | | | | |
| HCSL Driver Mode | 509 | 584 | 634 | mV | Each output terminated per Figure 37 with $R_L = 50 \Omega$ |
| CML Driver Mode | 456 | 565 | 644 | mV | Each output terminated per Figure 37 with $R_L = 50 \Omega$ connected to VDD (nominal 1.8 V) instead of GND |
| Voltage Swing Midpoint | | | | | |
| HCSL Driver Mode | 255 | 292 | 317 | mV | Each output terminated per Figure 37 with $R_L = 50 \Omega$ |
| CML Driver Mode | VDD – 325 | VDD – 291 | VDD – 266 | mV | Each output terminated per Figure 37 with $R_L = 50 \Omega$ connected to VDD (nominal 1.8 V) instead of GND |
| Output Current = 15 mA | | | | | |
| Voltage Swing (Peak-to-Peak) | | | | | |
| HCSL Driver Mode | 645 | 734 | 796 | mV | Each output terminated per Figure 37 with $R_L = 50 \Omega$ |
| CML Driver Mode | 589 | 721 | 815 | mV | Each output terminated per Figure 37 with $R_L = 50 \Omega$ connected to VDD (nominal 1.8 V) instead of GND |
| Voltage Swing Midpoint | | | | | |
| HCSL Driver Mode | 322 | 367 | 398 | mV | Each output terminated per Figure 37 with $R_L = 50 \Omega$ |
| CML Driver Mode | VDD – 411 | VDD – 367 | VDD – 334 | mV | Each output terminated per Figure 37 with $R_L = 50 \Omega$ connected to VDD (nominal 1.8 V) instead of GND |

TIME DURATION OF DIGITAL FUNCTIONS

Table 9.

| Parameter | Min | Typ | Max | Unit | Test Conditions/Comments |
|--|-----|-----|-----|------------------|--|
| TIME DURATION OF DIGITAL FUNCTIONS | | | | | |
| EEPROM to Register Download Time | | 10 | | ms | Using the Typical Configuration from Table 3 |
| Power-On Reset (POR) | | | 25 | ms | Time from power supplies > 80% to release of internal reset |
| Mx Pin to RESETB Rising Edge Setup Time | | | 1 | ns | Mx refers to Pin M0 through Pin M6 |
| Mx Pin to RESETB Rising Edge Hold Time | | | 2 | ns | |
| Multiple Mx Pin Timing Skew | | | 39 | ns | Applies only to multibit Mx pin functions |
| RESETB Falling Edge to Mx Pin High-Z Time | | | 14 | ns | |
| TIME FROM START OF DPLL ACTIVATION TO ACTIVE PHASE DETECTOR OUTPUT | | | | | |
| Untagged Operation | | | 10 | t_{PFD} | t_{PFD} is the nominal phase detector period given by R/f_{REF} , where R is the frequency division factor determined by the R divider, and f_{REF} is the frequency of the active reference |
| Tagged Operation | | | 10 | Tag period | Tag period = (tag ratio/ f_{TAG}), where f_{TAG} is either f_{REF} (for tagged reference mode) or f_{FEEDBACK} (for all other tagged modes); the tag ratio corresponds to the selection of f_{TAG} |

DIGITAL PLL (DPLL0, DPLL1) SPECIFICATIONS

Table 10.

| Parameter | Min | Typ | Max | Unit | Test Conditions/Comments |
|--|--------|------|-----------------|---------|---|
| DIGITAL PLL | | | | | |
| Digital Phase Detector (DPD) Input Frequency Range | 1 | | 2×10^5 | Hz | |
| Loop Filter | | | | | |
| Profile 0 | | | | | |
| Bandwidth | 0.0001 | | 1850 | Hz | Programmable design parameter; (f_{PFD} /bandwidth) \geq 20 |
| Phase Margin | | 70 | | Degrees | |
| Closed-Loop Peaking | | 1.1 | | dB | |
| Profile 1 | | | | | |
| Bandwidth | 0.0001 | | 305 | Hz | Programmable design parameter; (f_{PFD} /bandwidth) \geq 20 |
| Phase Margin | | 88.5 | | Degrees | |
| Closed-Loop Peaking | | | 0.1 | dB | |
| DIGITAL PLL NCO Division Ratio | | | | | |
| These specifications cover limitations on the DPLLx frequency tuning word (FTW0); the AD9542 evaluation software frequency planning wizard sets these values automatically for the user, and the AD9542 evaluation software is available for download from the AD9542 product page at www.analog.com/AD9542 ; NCO division = $2^{48}/\text{FTW0}$, which takes the form INT.FRAC, where INT is the integer portion, and FRAC is the fractional portion | | | | | |
| NCO Integer | 7 | | 13 | | This is the integer portion of NCO division |
| NCO Fraction | 0.05 | | 0.95 | | This is the fractional portion of NCO division |

DIGITAL PLL LOCK DETECTION SPECIFICATIONS

Table 11.

| Parameter | Min | Typ | Max | Unit | Test Conditions/Comments |
|-----------------------------|-----|-----|--------------|------|--|
| PHASE LOCK DETECTOR | | | | | |
| Threshold Programming Range | 10 | | $2^{24} - 1$ | ps | |
| Threshold Resolution | | 1 | | ps | |
| FREQUENCY LOCK DETECTOR | | | | | |
| Threshold Programming Range | 10 | | $2^{24} - 1$ | ps | |
| Threshold Resolution | | 1 | | ps | |
| PHASE STEP DETECTOR | | | | | |
| Threshold Programming Range | 100 | | $2^{32} - 1$ | ps | Setting this value too low causes false triggers |
| Threshold Resolution | | 1 | | ps | |

HOLDOVER SPECIFICATIONS

Table 12.

| Parameter | Min | Typ | Max | Unit | Test Conditions/Comments |
|--|-------|------------|-----------|------|--|
| HOLDOVER SPECIFICATIONS | | | | | |
| Initial Frequency Accuracy | | ± 0.01 | ± 0.1 | ppb | AD9542 is configured using Configuration 1 from Table 21; excludes frequency drift of system clock (SYSCLK) source; excludes frequency drift of input reference prior to entering holdover; 160 ms history timer; history holdoff setting of 8; three holdover history features (bits) are enabled: delay history until frequency lock bit, delay history until phase lock bit, and delay holdover history accumulation until not phase slew limited bit |
| Relative Frequency Accuracy Between Channels | | 0 | | ppb | |
| Cascaded Operation | | | | sec | |
| History Averaging Window | 0.001 | | 268435 | | |

ANALOG PLL (APLLO, APLL1) SPECIFICATIONS

Table 13.

| Parameter | Min | Typ | Max | Unit |
|--|------|-----|------|---------|
| VCO FREQUENCY RANGE | | | | |
| Analog PLL0 (APLLO) | 2424 | | 3232 | MHz |
| Analog PLL1 (APLL1) | 3232 | | 4040 | MHz |
| PHASE FREQUENCY DETECTOR (PFD) INPUT FREQUENCY RANGE | 162 | | 350 | MHz |
| LOOP BANDWIDTH | | 260 | | kHz |
| PHASE MARGIN | | 68 | | Degrees |

OUTPUT CHANNEL DIVIDER SPECIFICATIONS

Table 14.

| Parameter | Min | Typ | Max | Unit | Test Conditions/Comments |
|-------------------------------|-----|-----|-----|-----------|---|
| OUTPUT PHASE ADJUST STEP SIZE | 1 | | | t_{VCO} | $t_{VCO} = 1 / (\text{APLLx VCO frequency})$, where x = 0, 1 |

SYSTEM CLOCK COMPENSATION SPECIFICATIONS

Table 15.

| Parameter | Min | Typ | Max | Unit | Test Conditions/Comments |
|---|-----|-------|-----------------|------|--|
| DIRECT COMPENSATION | | | | | |
| Resolution | | 0.028 | | ppt | ppt is parts per trillion (10^{-12}) |
| CLOSED-LOOP COMPENSATION (AUXILIARY DPLL) | | | | | |
| Phase Detector Frequency | 2 | | 200 | kHz | |
| Loop Bandwidth | 0.1 | | 2×10^3 | Hz | |
| Reference Monitor Threshold | | 5 | | % | |

TEMPERATURE SENSOR SPECIFICATIONS

Table 16.

| Parameter | Min | Typ | Max | Unit | Test Conditions/Comments |
|-----------------|-----|------------|-----|------|---|
| TEMPERATURE | | | | | |
| Accuracy | | | | | |
| Absolute | | 5 | | °C | $T_A = -50^\circ\text{C}$ to $+110^\circ\text{C}$ |
| Relative | | 1.7 | | % | $T_A = -50^\circ\text{C}$ to $+110^\circ\text{C}$ |
| Resolution | | 0.0078 | | °C | 16-bit (signed) resolution |
| Conversion time | | 0.18 | | ms | |
| REPEATABILITY | | ± 0.02 | | °C | $T_A = 25^\circ\text{C}$ |
| DRIFT | | 0.1 | | °C | 500 hour stress test at 100°C |

SERIAL PORT SPECIFICATIONS

Serial Port Interface (SPI) Mode

Table 17.

| Parameter | Min | Typ | Max | Unit | Test Conditions/Comments |
|------------------------|--------------|-----|---------|---------------|--|
| CSB | | | | | Valid for VDDIOA = 3.3 V, 1.8 V, and 2.5 V |
| Input Logic 1 Voltage | VDDIOA – 0.4 | | | V | |
| Input Logic 0 Voltage | | | 0.4 | V | |
| Input Logic 1 Current | | 1 | | μA | |
| Input Logic 0 Current | | 1 | | μA | |
| SCLK | | | | | |
| Input Logic 1 Voltage | VDDIOA – 0.4 | | | V | |
| Input Logic 0 Voltage | | | 0.4 | V | |
| Input Logic 1 Current | | 1 | | μA | |
| Input Logic 0 Current | | 1 | | μA | |
| SDIO | | | | | |
| As an Input | | | | | |
| Input Logic 1 Voltage | VDDIOA – 0.4 | | | V | |
| Input Logic 0 Voltage | | | 0.4 | V | |
| Input Logic 1 Current | | 1 | | μA | |
| Input Logic 0 Current | | 1 | | μA | |
| As an Output | | | | | |
| Output Logic 1 Voltage | VDDIOA – 0.2 | | | V | 1 mA load current |
| Output Logic 0 Voltage | | | 0.2 | V | 1 mA load current |
| SDO | | | | | |
| Output Logic 1 Voltage | VDDIOA – 0.2 | | | V | 1 mA load current |
| Output Logic 0 Voltage | | | 0.2 | V | 1 mA load current |
| Leakage Current | | | ± 1 | μA | SDO inactive (high impedance) |
| TIMING | | | | | Valid for VDDIOA = 3.3 V, 1.8 V, and 2.5 V |

| Parameter | Min | Typ | Max | Unit | Test Conditions/Comments |
|--------------------------------------|-----|-----|-----|-----------|--------------------------|
| SCLK | | | | | |
| Clock Rate, $1/t_{CLK}$ | | | 50 | MHz | |
| Pulse Width High, t_{HIGH} | 5 | | | ns | |
| Pulse Width Low, t_{LOW} | 9 | | | ns | |
| SDIO to SCLK Setup, t_{DS} | 2.2 | | | ns | |
| SCLK to SDIO Hold, t_{DH} | 0 | | | ns | |
| SCLK to Valid SDIO and SDO, t_{DV} | | | 9 | ns | |
| CSB to SCLK Setup, t_S | 1.5 | | | ns | |
| CSB to SCLK Hold, t_C | 0 | | | ns | |
| CSB Minimum Pulse Width High | 1 | | | t_{CLK} | |

I²C Mode

Table 18.

| Parameter | Min | Typ | Max | Unit | Test Conditions/Comments |
|---|-----------------------|-----|------------------------|-------------|---|
| SDA, SCL (AS INPUTS) | | | | | Valid for VDDIOA = 3.3 V, 1.8 V, and 2.5 V |
| Input Logic 1 Voltage | 70 | | | % of VDDIOA | |
| Input Logic 0 Voltage | | | $0.3 \times V_{DDIOA}$ | V | |
| Input Current | -10 | | +10 | μA | For $V_{IN} = 10\%$ to 90% of VDDIOA |
| Hysteresis of Schmitt Trigger Inputs | 1.5 | | | % of VDDIOA | |
| SDA (AS OUTPUT) | | | | | |
| Output Logic 0 Voltage | | | 0.2 | V | $I_{OUT} = 3 \text{ mA}$ |
| Output Fall Time from V_{IH} Minimum to V_{IL} Maximum | $20 + 0.1 \times C_B$ | | 250 | ns | $10 \text{ pF} \leq C_B \leq 400 \text{ pF}$ |
| TIMING | | | | | |
| SCL Clock Rate | | | 400 | kHz | |
| Bus Free Time Between a Stop and Start Condition, t_{BUF} | 1.3 | | | μs | |
| Repeated Start Condition Setup Time, $t_{SU; STA}$ | 0.6 | | | μs | |
| Repeated Hold Time Start Condition, $t_{HD; STA}$ | 0.6 | | | μs | After this period, the first clock pulse is generated |
| Stop Condition Setup Time, $t_{SU; STO}$ | 0.6 | | | μs | |
| Low Period of the SCL Clock, t_{LOW} | 1.3 | | | μs | |
| High Period of the SCL Clock, t_{HIGH} | 0.6 | | | μs | |
| SCL/SDA Rise Time, t_R | $20 + 0.1 \times C_B$ | | 300 | ns | |
| SCL/SDA Fall Time, t_F | $20 + 0.1 \times C_B$ | | 300 | ns | |
| Data Setup Time, $t_{SU; DAT}$ | 100 | | | ns | |
| Data Hold Time, $t_{HD; DAT}$ | 100 | | | ns | |
| Capacitive Load for Each Bus Line, C_B | | | 400 | pF | |

LOGIC INPUT SPECIFICATIONS (RESETB, M0 TO M6)

Table 19.

| Parameter | Min | Typ | Max | Unit | Test Conditions/Comments |
|---|-----------------------|----------|-----------|---------------|---|
| RESETB | | | | | Valid for $3.3\text{ V} \geq \text{VDDIOA} \geq 1.8\text{ V}$; internal 100 k Ω pull-up resistor |
| Input High Voltage (V_{IH}) | $\text{VDDIOA} - 0.4$ | | | V | |
| Input Low Voltage (V_{IL}) | | | 0.4 | V | |
| Input Current High (I_{INH}) | | 1 | | μA | |
| Input Current Low (I_{INL}) | | ± 15 | ± 125 | μA | |
| LOGIC INPUTS (M0 to M6) | | | | | Valid for $3.3\text{ V} \geq \text{VDDIOx} \geq 1.8\text{ V}$; VDDIOA applies to the M5 pin and the M6 pin; VDDIOB applies to the M0, M1, M2, M3, and M4 pins; the M3 and M4 pins have internal 100 k Ω pull-down resistors |
| Frequency Range | | | 51 | MHz | |
| Input High Voltage (V_{IH}) | $\text{VDDIOx} - 0.4$ | | | V | |
| Input Low Voltage (V_{IL}) | | | 0.4 | V | |
| Input Current (I_{INH} , I_{INL}) | | ± 15 | ± 125 | μA | |

LOGIC OUTPUT SPECIFICATIONS (M0 TO M6)

Table 20.

| Parameter | Min | Typ | Max | Unit | Test Conditions/Comments |
|----------------------------------|-----------------------|-----|-----|------|---|
| LOGIC OUTPUTS (M0 to M6) | | | | | Valid for $3.3\text{ V} \geq \text{VDDIOx} \geq 1.8\text{ V}$; VDDIOA applies for the M5 and M6 pins; VDDIOB applies for M0 to M4; normal (default) output drive current setting for M0 through M6 |
| Frequency Range | | | 26 | MHz | |
| Output High Voltage (V_{OH}) | $\text{VDDIOx} - 0.6$ | | | V | Load current = 10 mA |
| | $\text{VDDIOx} - 0.2$ | | | V | Load current = 1 mA |
| Output Low Voltage (V_{OL}) | | | 0.6 | V | Load current = 10 mA |
| | | | 0.2 | V | Load current = 1 mA |

JITTER GENERATION (RANDOM JITTER)

Table 21.

| Parameter | Min | Typ | Max | Unit | Test Conditions/Comments |
|---|-----|-----|-----|------|---|
| JITTER GENERATION | | | | | System clock doubler enabled; high phase margin mode enabled; there is not a significant jitter difference between driver modes Channel 1 powered down |
| Channel 0—DPLL0, APLL0 RMS Jitter (12 kHz to 20 MHz) Configuration 1—155.52 MHz | | 223 | | fs | Device configuration: $f_{\text{SYSCLK}} = 52\text{ MHz XTAL}$, $f_{\text{REF}} = 38.88\text{ MHz}$, $f_{\text{VCO}} = 2488.32\text{ MHz}$, $f_{\text{OUT}} = 155.52\text{ MHz}$, $\text{BW}_{\text{DPLL}} = 50\text{ Hz}$, phase buildout operation |
| Configuration 2—245.76 MHz | | 220 | | fs | Device configuration: $f_{\text{SYSCLK}} = 52\text{ MHz XTAL}$, $f_{\text{REF}} = 30.72\text{ MHz}$, $f_{\text{VCO}} = 2457.6\text{ MHz}$, $f_{\text{OUT}} = 245.76\text{ MHz}$, $\text{BW}_{\text{DPLL}} = 50\text{ Hz}$, internal zero delay operation |
| Configuration 3—491.52 MHz | | 235 | | fs | Device configuration: $f_{\text{SYSCLK}} = 52\text{ MHz XTAL}$, $f_{\text{COMP}} = 19.2\text{ MHz}$ temperature compensated crystal oscillator (TCXO), $\text{BW}_{\text{COMP}} = 50\text{ Hz}$, $f_{\text{REF}} = 1\text{ Hz}$, $f_{\text{VCO}} = 2949.12\text{ MHz}$, $f_{\text{OUT}} = 491.52\text{ MHz}$, $\text{BW}_{\text{DPLL}} = 50\text{ mHz}$, phase buildout operation |
| Configuration 4—125 MHz | | 213 | | fs | Device configuration: $f_{\text{SYSCLK}} = 52\text{ MHz XTAL}$, $f_{\text{COMP}} = 19.2\text{ MHz TCXO}$, $\text{BW}_{\text{COMP}} = 50\text{ Hz}$, $f_{\text{REF}} = 125\text{ MHz}$, $f_{\text{VCO}} = 2500\text{ MHz}$, $f_{\text{OUT}} = 125\text{ MHz}$, $\text{BW}_{\text{DPLL}} = 0.1\text{ Hz}$, phase buildout operation |
| Configuration 5—312.5 MHz | | 217 | | fs | Device configuration: $f_{\text{SYSCLK}} = 52\text{ MHz XTAL}$, $f_{\text{REF}} = 25\text{ MHz}$, $f_{\text{VCO}} = 2500\text{ MHz}$, $f_{\text{OUT}} = 312.5\text{ MHz}$, $\text{BW}_{\text{DPLL}} = 50\text{ Hz}$, phase buildout operation |
| Configuration 6—174.7030837 MHz | | 230 | | fs | Device configuration: $f_{\text{SYSCLK}} = 52\text{ MHz XTAL}$, $f_{\text{REF}} = 155.52\text{ MHz}$, $f_{\text{VCO}} = 2620.5463\text{ MHz}$, $f_{\text{OUT}} = (155.52 \times 255/227)\text{ MHz}$, $\text{BW}_{\text{DPLL}} = 50\text{ Hz}$ |

| Parameter | Min | Typ | Max | Unit | Test Conditions/Comments |
|---|-----|-----|-----|------|--|
| Channel 1—DPLL1, APLL1 RMS Jitter (12 kHz to 20 MHz) | | | | | Channel 0 powered down |
| Configuration 1—155.52 MHz | | 247 | | fs | Device configuration: $f_{\text{SYSCLK}} = 52 \text{ MHz XTAL}$, $f_{\text{REF}} = 38.88 \text{ MHz}$, $f_{\text{VCO}} = 3265.92 \text{ MHz}$, $f_{\text{OUT}} = 155.52 \text{ MHz}$, $\text{BW}_{\text{DPLL}} = 50 \text{ Hz}$, phase buildout operation, half divide enabled |
| Configuration 2—245.76 MHz | | 280 | | fs | Device configuration: $f_{\text{SYSCLK}} = 52 \text{ MHz XTAL}$, $f_{\text{REF}} = 30.72 \text{ MHz}$, $f_{\text{VCO}} = 3686.4 \text{ MHz}$, $f_{\text{OUT}} = 245.76 \text{ MHz}$, $\text{BW}_{\text{DPLL}} = 50 \text{ Hz}$, half divide enabled, internal zero delay operation |
| Configuration 3—491.52 MHz | | 323 | | fs | Device configuration: $f_{\text{SYSCLK}} = 52 \text{ MHz XTAL}$, $f_{\text{COMP}} = 19.2 \text{ MHz TCXO}$, $\text{BW}_{\text{COMP}} = 50 \text{ Hz}$, $f_{\text{REF}} = 1 \text{ Hz}$, $f_{\text{VCO}} = 3932.16 \text{ MHz}$, $f_{\text{OUT}} = 491.52 \text{ MHz}$, $\text{BW}_{\text{DPLL}} = 50 \text{ mHz}$, phase buildout operation |
| Configuration 4—125 MHz | | 243 | | fs | Device configuration: $f_{\text{SYSCLK}} = 52 \text{ MHz XTAL}$, $f_{\text{COMP}} = 19.2 \text{ MHz TCXO}$, $\text{BW}_{\text{COMP}} = 50 \text{ Hz}$, $f_{\text{REF}} = 125 \text{ MHz}$, $f_{\text{VCO}} = 3250 \text{ MHz}$, $f_{\text{OUT}} = 125 \text{ MHz}$, $\text{BW}_{\text{DPLL}} = 0.1 \text{ Hz}$, phase buildout operation |
| Configuration 5—312.5 MHz | | 266 | | fs | Device configuration: $f_{\text{SYSCLK}} = 52 \text{ MHz XTAL}$, $f_{\text{REF}} = 25 \text{ MHz}$, $f_{\text{VCO}} = 3750 \text{ MHz}$, $f_{\text{OUT}} = 312.5 \text{ MHz}$, $\text{BW}_{\text{DPLL}} = 50 \text{ Hz}$, phase buildout operation |
| Configuration 6—174.7030837 MHz | | 264 | | fs | Device configuration: $f_{\text{SYSCLK}} = 52 \text{ MHz XTAL}$, $f_{\text{REF}} = 155.52 \text{ MHz}$, $f_{\text{VCO}} = 3319.3586 \text{ MHz}$, $f_{\text{OUT}} = (155.52 \times 255/227) \text{ MHz}$, $\text{BW}_{\text{DPLL}} = 50 \text{ Hz}$, phase buildout operation |

PHASE NOISE

Table 22.

| Parameter | Min | Typ | Max | Unit | Test Conditions/Comments |
|---|-----|------|-----|--------|--|
| PHASE NOISE | | | | | System clock doubler enabled; high phase margin mode enabled; there is not a significant jitter difference between driver modes |
| Channel 0—DPLL0, APLL0 RMS Jitter (12 kHz to 20 MHz) | | | | | Channel 1 powered down |
| Configuration 1—155.52 MHz | | | | | Device configuration: $f_{\text{SYSCLK}} = 52 \text{ MHz XTAL}$, $f_{\text{REF}} = 38.88 \text{ MHz}$, $f_{\text{VCO}} = 2488.32 \text{ MHz}$, $f_{\text{OUT}} = 155.52 \text{ MHz}$, $\text{BW}_{\text{DPLL}} = 50 \text{ Hz}$, phase buildout operation |
| 10 Hz Offset | | -81 | | dBc/Hz | |
| 100 Hz Offset | | -98 | | dBc/Hz | |
| 1 kHz Offset | | -118 | | dBc/Hz | |
| 10 kHz Offset | | -128 | | dBc/Hz | |
| 100 kHz Offset | | -134 | | dBc/Hz | |
| 1 MHz Offset | | -144 | | dBc/Hz | |
| 10 MHz Offset | | -158 | | dBc/Hz | |
| Floor | | -161 | | dBc/Hz | |
| Configuration 2—245.76 MHz | | | | | Device configuration: $f_{\text{SYSCLK}} = 52 \text{ MHz XTAL}$, $f_{\text{REF}} = 30.72 \text{ MHz}$, $f_{\text{VCO}} = 2457.6 \text{ MHz}$, $f_{\text{OUT}} = 245.76 \text{ MHz}$, $\text{BW}_{\text{DPLL}} = 50 \text{ Hz}$, internal zero delay operation |
| 10 Hz Offset | | -77 | | dBc/Hz | |
| 100 Hz Offset | | -93 | | dBc/Hz | |
| 1 kHz Offset | | -114 | | dBc/Hz | |
| 10 kHz Offset | | -125 | | dBc/Hz | |
| 100 kHz Offset | | -130 | | dBc/Hz | |
| 1 MHz Offset | | -140 | | dBc/Hz | |
| 10 MHz Offset | | -156 | | dBc/Hz | |
| Floor | | -161 | | dBc/Hz | |
| Configuration 3—491.52 MHz | | | | | Device configuration: $f_{\text{SYSCLK}} = 52 \text{ MHz XTAL}$, $f_{\text{COMP}} = 19.2 \text{ MHz TCXO}$, $\text{BW}_{\text{COMP}} = 50 \text{ Hz}$, $f_{\text{REF}} = 1 \text{ Hz}$, $f_{\text{VCO}} = 2949.12 \text{ MHz}$, $f_{\text{OUT}} = 491.52 \text{ MHz}$, $\text{BW}_{\text{DPLL}} = 50 \text{ mHz}$, phase buildout operation |
| 10 Hz Offset | | -74 | | dBc/Hz | |
| 100 Hz Offset | | -89 | | dBc/Hz | |
| 1 kHz Offset | | -108 | | dBc/Hz | |

| Parameter | Min | Typ | Max | Unit | Test Conditions/Comments |
|---------------------------------|-----|------|-----|--------|---|
| 10 kHz Offset | | -119 | | dBc/Hz | Device configuration: $f_{\text{SYSCLK}} = 52 \text{ MHz XTAL}$, $f_{\text{COMP}} = 19.2 \text{ MHz TCXO}$, $BW_{\text{COMP}} = 50 \text{ Hz}$, $f_{\text{REF}} = 125 \text{ MHz}$, $f_{\text{VCO}} = 2500 \text{ MHz}$, $f_{\text{OUT}} = 125 \text{ MHz}$, $BW_{\text{DPLL}} = 0.1 \text{ Hz}$, phase buildout operation |
| 100 kHz Offset | | -123 | | dBc/Hz | |
| 1 MHz Offset | | -134 | | dBc/Hz | |
| 10 MHz Offset | | -152 | | dBc/Hz | |
| Floor | | -159 | | | |
| Configuration 4—125 MHz | | | | | |
| 10 Hz Offset | | -84 | | dBc/Hz | |
| 100 Hz Offset | | -106 | | dBc/Hz | |
| 1 kHz Offset | | -120 | | dBc/Hz | |
| 10 kHz Offset | | -131 | | dBc/Hz | |
| 100 kHz Offset | | -136 | | dBc/Hz | |
| 1 MHz Offset | | -147 | | dBc/Hz | |
| 10 MHz Offset | | -160 | | dBc/Hz | |
| Floor | | -163 | | dBc/Hz | |
| Configuration 5—312.5 MHz | | | | | Device configuration: $f_{\text{SYSCLK}} = 52 \text{ MHz XTAL}$, $f_{\text{REF}} = 25 \text{ MHz}$, $f_{\text{VCO}} = 2500 \text{ MHz}$, $f_{\text{OUT}} = 312.5 \text{ MHz}$, $BW_{\text{DPLL}} = 50 \text{ Hz}$, phase buildout operation |
| 10 Hz Offset | | -74 | | dBc/Hz | |
| 100 Hz Offset | | -91 | | dBc/Hz | |
| 1 kHz Offset | | -112 | | dBc/Hz | |
| 10 kHz Offset | | -123 | | dBc/Hz | |
| 100 kHz Offset | | -128 | | dBc/Hz | |
| 1 MHz Offset | | -138 | | dBc/Hz | |
| 10 MHz Offset | | -154 | | dBc/Hz | |
| Floor | | -161 | | dBc/Hz | |
| Configuration 6—174.7030837 MHz | | | | | Device configuration: $f_{\text{SYSCLK}} = 52 \text{ MHz XTAL}$, $f_{\text{REF}} = 155.52 \text{ MHz}$, $f_{\text{VCO}} = 2620.5463 \text{ MHz}$, $f_{\text{OUT}} = (155.52 \times 255/227) \text{ MHz}$, $BW_{\text{DPLL}} = 50 \text{ Hz}$ |
| 10 Hz Offset | | -82 | | dBc/Hz | |
| 100 Hz Offset | | -99 | | dBc/Hz | |
| 1 kHz Offset | | -117 | | dBc/Hz | |
| 10 kHz Offset | | -127 | | dBc/Hz | |
| 100 kHz Offset | | -133 | | dBc/Hz | |
| 1 MHz Offset | | -143 | | dBc/Hz | |
| 10 MHz Offset | | -157 | | dBc/Hz | |
| Floor | | -160 | | dBc/Hz | |
| Channel 1—DPLL1, APLL1 | | | | | Channel 0 powered down |
| RMS Jitter (12 kHz to 20 MHz) | | | | | |
| Configuration 1—155.52 MHz | | | | | Device configuration: $f_{\text{SYSCLK}} = 52 \text{ MHz XTAL}$, $f_{\text{REF}} = 38.88 \text{ MHz}$, $f_{\text{VCO}} = 3265.92 \text{ MHz}$, $f_{\text{OUT}} = 155.52 \text{ MHz}$, $BW_{\text{DPLL}} = 50 \text{ Hz}$, phase buildout operation, half divide enabled |
| 10 Hz Offset | | -81 | | dBc/Hz | |
| 100 Hz Offset | | -98 | | dBc/Hz | |
| 1 kHz Offset | | -118 | | dBc/Hz | |
| 10 kHz Offset | | -128 | | dBc/Hz | |
| 100 kHz Offset | | -132 | | dBc/Hz | |
| 1 MHz Offset | | -144 | | dBc/Hz | |
| 10 MHz Offset | | -158 | | dBc/Hz | |
| Floor | | -162 | | dBc/Hz | |
| Configuration 2—245.76 MHz | | | | | Device configuration: $f_{\text{SYSCLK}} = 52 \text{ MHz XTAL}$, $f_{\text{REF}} = 30.72 \text{ MHz}$, $f_{\text{VCO}} = 3686.4 \text{ MHz}$, $f_{\text{OUT}} = 245.76 \text{ MHz}$, $BW_{\text{DPLL}} = 50 \text{ Hz}$, half divide enabled; internal zero delay operation |
| 10 Hz Offset | | -76 | | dBc/Hz | |
| 100 Hz Offset | | -93 | | dBc/Hz | |
| 1 kHz Offset | | -114 | | dBc/Hz | |
| 10 kHz Offset | | -124 | | dBc/Hz | |

| Parameter | Min | Typ | Max | Unit | Test Conditions/Comments |
|---------------------------------|-----|------|-----|--------|--|
| 100 kHz Offset | | -127 | | dBc/Hz | Device configuration: $f_{\text{SYSCLK}} = 52 \text{ MHz XTAL}$, $f_{\text{COMP}} = 19.2 \text{ MHz TCXO}$, $BW_{\text{COMP}} = 50 \text{ Hz}$, $f_{\text{REF}} = 1 \text{ Hz}$, $f_{\text{VCO}} = 3932.16 \text{ MHz}$, $f_{\text{OUT}} = 491.52 \text{ MHz}$, $BW_{\text{DPLL}} = 50 \text{ mHz}$, phase buildout operation |
| 1 MHz Offset | | -138 | | dBc/Hz | |
| 10 MHz Offset | | -156 | | dBc/Hz | |
| Floor | | -161 | | dBc/Hz | |
| Configuration 3—491.52 MHz | | | | | |
| 10 Hz Offset | | -74 | | dBc/Hz | |
| 100 Hz Offset | | -90 | | dBc/Hz | |
| 1 kHz Offset | | -108 | | dBc/Hz | |
| 10 kHz Offset | | -118 | | dBc/Hz | |
| 100 kHz Offset | | -120 | | dBc/Hz | |
| 1 MHz Offset | | -131 | | dBc/Hz | Device configuration: $f_{\text{SYSCLK}} = 52 \text{ MHz XTAL}$, $f_{\text{COMP}} = 19.2 \text{ MHz TCXO}$, $BW_{\text{COMP}} = 50 \text{ Hz}$, $f_{\text{REF}} = 125 \text{ MHz}$, $f_{\text{VCO}} = 3250 \text{ MHz}$, $f_{\text{OUT}} = 125 \text{ MHz}$, $BW_{\text{DPLL}} = 0.1 \text{ Hz}$, phase buildout operation |
| 10 MHz Offset | | -150 | | dBc/Hz | |
| Floor | | -160 | | dBc/Hz | |
| Configuration 4—125 MHz | | | | | |
| 10 Hz Offset | | -83 | | dBc/Hz | |
| 100 Hz Offset | | -106 | | dBc/Hz | |
| 1 kHz Offset | | -120 | | dBc/Hz | |
| 10 kHz Offset | | -131 | | dBc/Hz | |
| 100 kHz Offset | | -135 | | dBc/Hz | |
| 1 MHz Offset | | -145 | | dBc/Hz | |
| 10 MHz Offset | | -160 | | dBc/Hz | Device configuration: $f_{\text{SYSCLK}} = 52 \text{ MHz XTAL}$, $f_{\text{REF}} = 25 \text{ MHz}$, $f_{\text{VCO}} = 3750 \text{ MHz}$, $f_{\text{OUT}} = 312.5 \text{ MHz}$, $BW_{\text{DPLL}} = 50 \text{ Hz}$, phase buildout operation |
| Floor | | -163 | | dBc/Hz | |
| Configuration 5—312.5 MHz | | | | | |
| 10 Hz Offset | | -73 | | dBc/Hz | |
| 100 Hz Offset | | -91 | | dBc/Hz | |
| 1 kHz Offset | | -112 | | dBc/Hz | |
| 10 kHz Offset | | -122 | | dBc/Hz | |
| 100 kHz Offset | | -125 | | dBc/Hz | |
| 1 MHz Offset | | -137 | | dBc/Hz | |
| 10 MHz Offset | | -154 | | dBc/Hz | |
| Floor | | -161 | | dBc/Hz | Device configuration: $f_{\text{SYSCLK}} = 52 \text{ MHz XTAL}$, $f_{\text{REF}} = 155.52 \text{ MHz}$, $f_{\text{VCO}} = 3319.3586 \text{ MHz}$, $f_{\text{OUT}} = (155.52 \times 255/227) \text{ MHz}$, $BW_{\text{DPLL}} = 50 \text{ Hz}$ |
| Configuration 6—174.7030837 MHz | | | | | |
| 10 Hz Offset | | -77 | | dBc/Hz | |
| 100 Hz Offset | | -99 | | dBc/Hz | |
| 1 kHz Offset | | -117 | | dBc/Hz | |
| 10 kHz Offset | | -127 | | dBc/Hz | |
| 100 kHz Offset | | -131 | | dBc/Hz | |
| 1 MHz Offset | | -142 | | dBc/Hz | |
| 10 MHz Offset | | -158 | | dBc/Hz | |
| Floor | | -161 | | dBc/Hz | |

ABSOLUTE MAXIMUM RATINGS

Table 23.

| Parameter | Rating |
|---|--------------------------|
| 1.8 V Supply Voltage (VDD) | 2 V |
| Input/Output Supply Voltage (VDDIOA, VDDIOB) | 3.6 V |
| Input Voltage Range (XOA, XOB, REFA, REFAA, REFB, REFBB Pins) | −0.5 V to VDD + 0.5 V |
| Digital Input Voltage Range SDO/M5, SCLK/SCL, SDIO/SDA, CSB/M6 Pins | −0.5 V to VDDIOA + 0.5 V |
| M0, M1, M2, M3, M4 Pins | −0.5 V to VDDIOB + 0.5 V |
| Storage Temperature Range | −65°C to +150°C |
| Operating Temperature Range ¹ | −40°C to +85°C |
| Lead Temperature (Soldering 10 sec) | 300°C |

¹ See the Thermal Resistance section for additional information.

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

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

Table 24. Thermal Resistance

| Symbol | Thermal Characteristic Using a JEDEC51-7 Plus JEDEC51-5 2S2P Test Board ¹ | Value | Unit |
|----------------|---|-------------------|------|
| θ_{JA} | Junction to ambient thermal resistance, 0.0 m/sec airflow per JEDEC JESD51-2 (still air) | 23.9 ² | °C/W |
| θ_{JMA} | Junction to ambient thermal resistance, 1.0 m/sec airflow per JEDEC JESD51-6 (moving air) | 19.4 ² | °C/W |
| θ_{JMA} | Junction to ambient thermal resistance, 2.5 m/sec airflow per JEDEC JESD51-6 (moving air) | 18.2 ² | °C/W |
| θ_{JC} | Junction to case thermal resistance (die to heat sink) per MIL-STD 883, Method 1012.1 | 1.5 ² | °C/W |

¹ The exposed pad on the bottom of the package must be soldered to ground to achieve the specified thermal performance.

Values of θ_{JA} are for package comparison and PCB design considerations. θ_{JA} provides for a first-order approximation of T_J per the following equation:

$$T_J = T_A + (\theta_{JA} \times PD)$$

where T_A is the ambient temperature (°C).

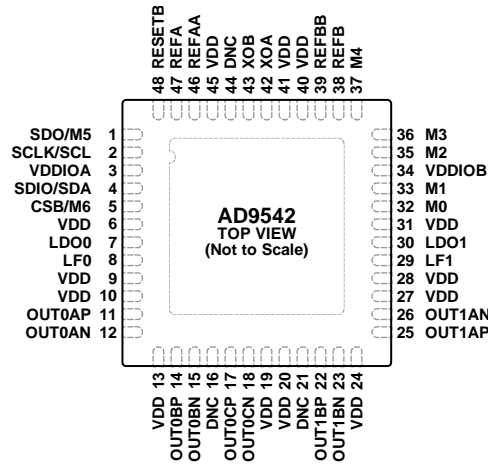
Values of θ_{JC} are for package comparison and PCB design considerations when an external heat sink is required.

ESD CAUTION



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

PIN CONFIGURATION AND FUNCTION DESCRIPTIONS



- NOTES
1. EXPOSED PAD. THE EXPOSED PAD IS THE GROUND CONNECTION ON THE CHIP. THE EXPOSED PAD MUST BE SOLDERED TO THE ANALOG GROUND OF THE PCB TO ENSURE PROPER FUNCTIONALITY AND FOR HEAT DISSIPATION, NOISE, AND MECHANICAL STRENGTH BENEFITS.
 2. DNC = DO NOT CONNECT. LEAVE THESE PINS FLOATING.

Figure 2. Pin Configuration

Table 25. Pin Function Descriptions

| Pin No. | Mnemonic | Input/Output | Pin Type | Description |
|--|----------|--------------|-----------------------|---|
| 1 | SDO/M5 | Output | CMOS | Serial Data Output (SDO). This pin is for reading serial data in 4-wire SPI mode. Changes to the VDDIOA supply voltage affect the V_{IH} and V_{OH} values for this pin. Configurable Input/Output (M5). This pin is a status and control pin when the device is not in 4-wire SPI mode. |
| 2 | SCLK/SCL | Input | CMOS | Serial Programming Clock (SCLK) Pin in SPI Mode. Changes to the VDDIOA supply voltage affect the V_{IH} and V_{OH} values for this pin. Serial Clock Pin (SCL) in I ² C Mode. Changes to the VDDIOA supply voltage affect the V_{IH} and V_{OH} values for this pin. |
| 3 | VDDIOA | Input | Power | Serial Port Power Supply. The valid supply voltage is 1.8 V, 2.5 V, or 3.3 V. The VDDIOA pin can be connected to the VDD supply bus if 1.8 V operation is desired. |
| 4 | SDIO/SDA | Input/output | CMOS | Serial Data Input/Output in SPI Mode (SDIO). Write data to this pin in 4-wire SPI mode. This pin has no internal pull-up or pull-down resistor. Changes to the VDDIOA supply voltage affect the V_{IH} and V_{OH} values for this pin. Serial Data Pin in I ² C Mode (SDA). |
| 5 | CSB/M6 | Input/output | CMOS | Chip Select in SPI Mode (CSB). Active low input. Maintain a Logic 0 level on this pin when programming the device in SPI mode. This pin has an internal 10 k Ω pull-up resistor. Changes to the VDDIOA supply voltage affect the V_{IH} and V_{OH} values for this pin. Configurable Input/Output (M6). This pin is a status and control pin when the device is not in SPI mode. |
| 6, 9, 10, 13, 19, 20, 24, 27, 28, 31, 40, 41, 45 | VDD | Input | Power | 1.8 V Power Supply. |
| 7 | LDO0 | Input | LDO bypass | APLL0 Loop Filter Voltage Regulator. Connect a 0.22 μ F capacitor from this pin to ground. This pin is the ac ground reference for the integrated APLL0 loop filter. |
| 8 | LF0 | Input/output | Loop filter for APLL0 | Loop Filter Node for APLL0. Connect a 3.9 nF capacitor from this pin to Pin 7 (LDO0). |
| 11 | OUT0AP | Output | HCSL, LVDS, CML, CMOS | PLL0 Output 0A. |
| 12 | OUT0AN | Output | HCSL, LVDS, CML, CMOS | PLL0 Complementary Output 0A. |

| Pin No. | Mnemonic | Input/ Output | Pin Type | Description |
|-----------------------|-----------------------|------------------|--|---|
| 14 | OUT0BP | Output | HCSL, LVDS, CML, CMOS | PLL0 Output 0B. |
| 15 | OUT0BN | Output | HCSL, LVDS, CML, CMOS | PLL0 Complementary Output 0B. |
| 16, 21, 44 | DNC | DNC | No Connect | Do Not Connect. Leave these pins floating. |
| 17 | OUT0CP | Output | HCSL, LVDS, CML, CMOS | PLL0 Output 0C. |
| 18 | OUT0CN | Output | HCSL, LVDS, CML, CMOS | PLL0 Complementary Output 0C. |
| 22 | OUT1BP | Output | HCSL, LVDS, CML, CMOS | PLL1 Output 1B. |
| 23 | OUT1BN | Output | HCSL, LVDS, CML, CMOS | PLL1 Complementary Output 1B. |
| 25 | OUT1AP | Output | HCSL, LVDS, CML, CMOS | PLL1 Output 1A. |
| 26 | OUT1AN | Input/ Output | HCSL, LVDS, CML, CMOS | PLL1 Complementary Output 1A. |
| 29 | LF1 | Input/ output | Loop filter for APLL1 | Loop Filter Node for APLL1. Connect a 3.9 nF capacitor from this pin to Pin 30 (LDO1). |
| 30 | LDO1 | Input | LDO bypass | APLL1 Loop Filter Voltage Regulator. Connect a 0.1 μ F capacitor from this pin to ground. This pin is the ac ground reference for the integrated APLL1 loop filter. |
| 32, 33, 35, 36, 37 | M0, M1, M2, M3, M4 | Input/ output | CMOS | Configurable Input/Output Pins. These are status and control pins. Changes to the VDDIOB supply voltage affect the V_{IH} and V_{OH} values for these pins. M3 and M4 have internal 100 k Ω pull-down resistors. M0, M1, and M2 do not have internal resistors. |
| 34 | VDDIOB | Input | Power | Mx Pin Power Supply. This power supply powers the digital section that controls the M0 to M4 pins. Valid supply voltages are 1.8 V, 2.5 V, or 3.3 V. The VDDIOB pin can be connected to the VDD supply bus if 1.8 V operation is desired. |
| 38 | REFB | Input | 1.8 V single- ended or differential input | Reference B Input. This internally biased input is typically ac-coupled; when configured in this manner, it can accept any differential signal with a single-ended swing up to the VDD power supply. If dc-coupled, the input can be LVDS or single-ended 1.8 V CMOS. |
| 39 | REFBB | Input | 1.8 V single- ended or differential input | Reference BB Input or Complementary Reference B Input. If REFB is in differential mode, the REFB complementary signal is on this pin. No connection is necessary to this pin if REFB is a single-ended input and REFBB is not used. |
| 42 | XOA | Input | Differential input | System Clock Input. XOA contains internal dc biasing and is ac-coupled with a 0.01 μ F capacitor except when using a crystal. When a crystal is used, connect the crystal across XOA and XOB. A single-ended CMOS input is also an option, but it can produce spurious spectral content when the duty cycle is not 50%. When using XOA as a single-ended input, connect a 0.1 μ F capacitor from XOB to ground. |
| 43 | XOB | Input | Differential input | Complementary System Clock Input. Complementary signal to XOA. XOB contains internal dc biasing and is ac-coupled with a 0.1 μ F capacitor except when using a crystal. When a crystal is used, connect the crystal across XOA and XOB. |
| 46 | REFAA | Input | 1.8 V single- ended or differential input | Reference AA input or Complementary REFA Input. If REFA is in differential mode, the REFA complementary signal is on this pin. No connection is necessary to this pin if REFA is a single-ended input and REFAA is not used. If dc-coupled, the input is single-ended 1.8 V CMOS. |
| 47 | REFA | Input | 1.8 V single- ended or differential input | Reference A Input. This internally biased input is typically ac-coupled; when configured in this manner, it can accept any differential signal with a single-ended swing up to the VDD power supply. If dc-coupled, the input can be LVDS or single-ended 1.8 V CMOS. |
| 48 | RESETB | Input | 1.8 V CMOS logic | Active Low Chip Reset. This pin has an internal 100 k Ω pull-up resistor. When asserted, the chip goes into reset. Changes to the VDDIOA supply voltage affect the V_{IH} values for this pin. |
| EP | EPAD | Output | Exposed pad | Exposed Pad. The exposed pad is the ground connection on the chip. The exposed pad must be soldered to the analog ground of the PCB to ensure proper functionality and for heat dissipation, noise, and mechanical strength benefits. |

TYPICAL PERFORMANCE CHARACTERISTICS

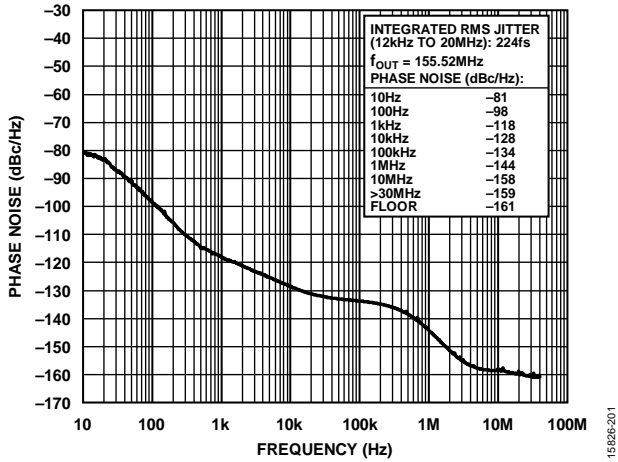


Figure 3. Absolute Phase Noise (PLL0, Configuration 1, HCSSL Mode, $f_{REF} = 38.88\text{ MHz}$, $f_{OUT} = 155.52\text{ MHz}$, $f_{SYS} = 52\text{ MHz}$ Crystal, 50 Hz DPLL BW)

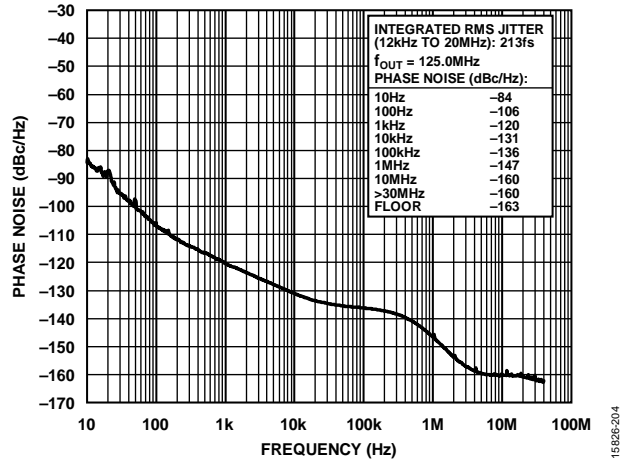


Figure 6. Absolute Phase Noise (PLL0, Configuration 4, HCSSL Mode, $f_{REF} = 125\text{ MHz}$, $f_{OUT} = 125.0\text{ MHz}$, $f_{COMP} = 19.2\text{ MHz}$ TCXO, $f_{SYS} = 52\text{ MHz}$ Crystal, 0.1 Hz DPLL BW, Phase Buildout Mode)

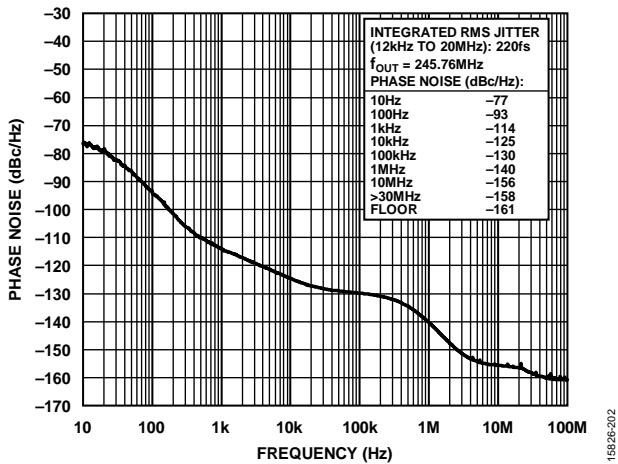


Figure 4. Absolute Phase Noise (PLL0, Configuration 2, HCSSL Mode, $f_{REF} = 30.72\text{ MHz}$, $f_{OUT} = 245.76\text{ MHz}$, $f_{SYS} = 52\text{ MHz}$ Crystal, 50 Hz DPLL BW)

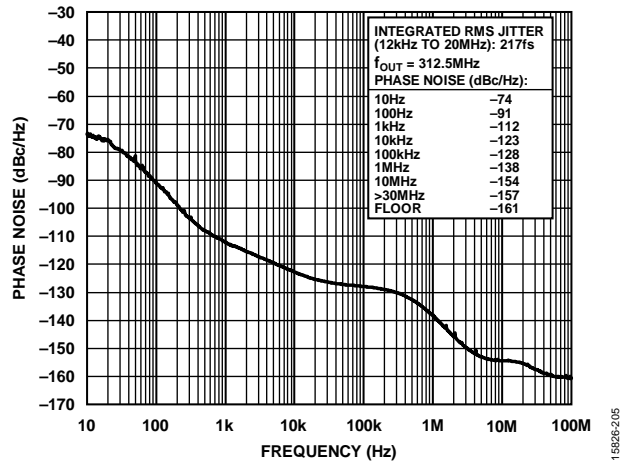


Figure 7. Absolute Phase Noise (PLL0, Configuration 5, HCSSL Mode, $f_{REF} = 25\text{ MHz}$, $f_{OUT} = 312.5\text{ MHz}$, $f_{SYS} = 52\text{ MHz}$ Crystal, 50 Hz DPLL BW, Phase Buildout Mode)

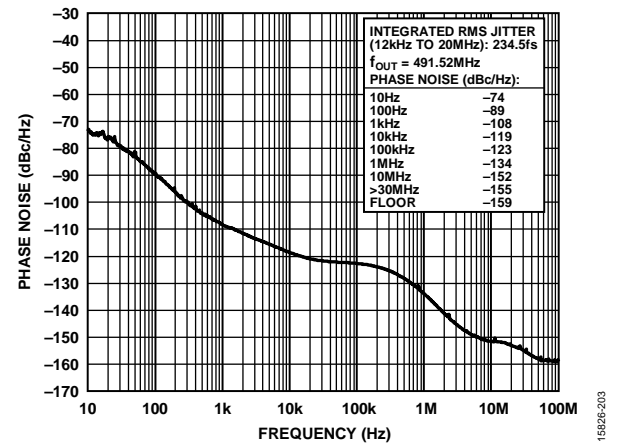


Figure 5. Absolute Phase Noise (PLL0, Configuration 3, HCSSL Mode, $f_{REF} = 1\text{ Hz}$, $f_{OUT} = 491.52\text{ MHz}$, $f_{COMP} = 19.2\text{ MHz}$ TCXO, $f_{SYS} = 52\text{ MHz}$ Crystal, 50 MHz DPLL BW)

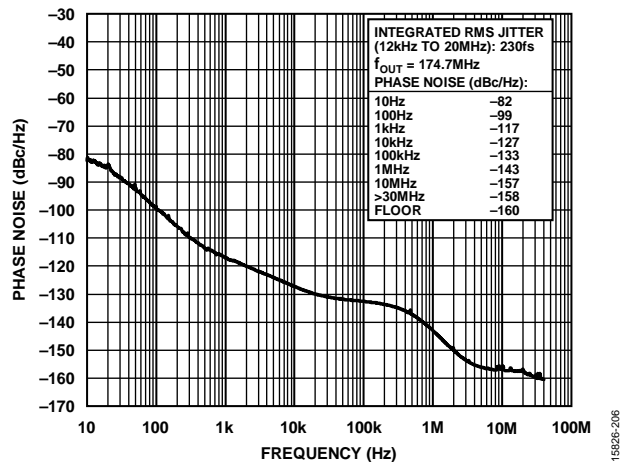
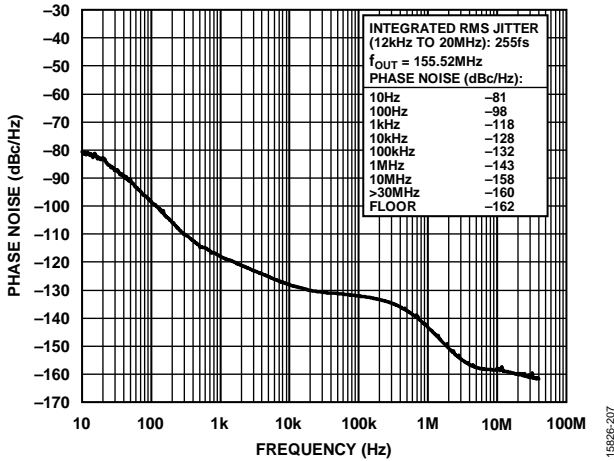
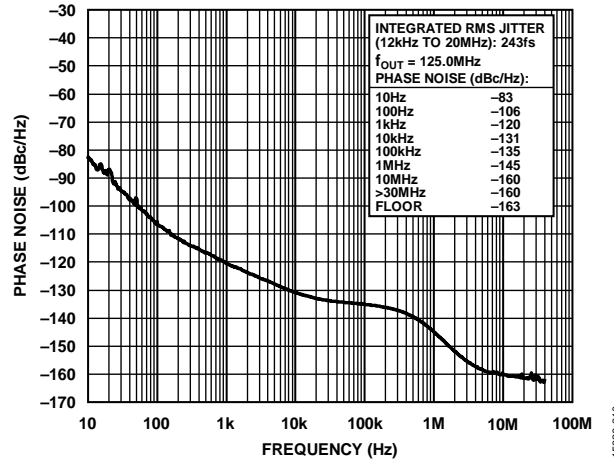


Figure 8. Absolute Phase Noise (PLL0, Configuration 6, HCSSL Mode, $f_{REF} = 155.52\text{ MHz}$, $f_{OUT} = 174.7\text{ MHz}$, $f_{SYS} = 52\text{ MHz}$ Crystal, 50 Hz DPLL BW, Phase Buildout Mode)



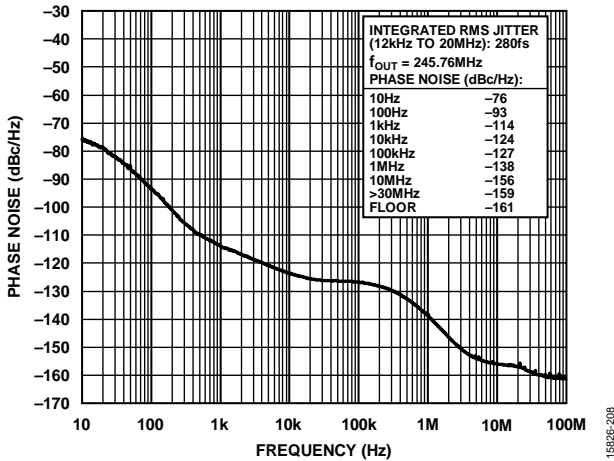
15826-207

Figure 9. Absolute Phase Noise (PLL1, Configuration 1, HCSL Mode, $f_{REF} = 38.88\text{ MHz}$, $f_{OUT} = 155.52\text{ MHz}$, $f_{SYS} = 52\text{ MHz}$ Crystal, 50 Hz DPLL BW)



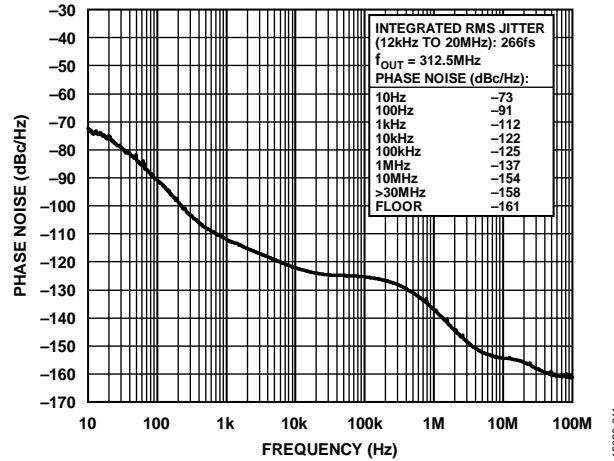
15826-210

Figure 12. Absolute Phase Noise (PLL1, Configuration 4, HCSL Mode, $f_{REF} = 125\text{ MHz}$, $f_{OUT} = 125\text{ MHz}$, $f_{COMP} = 19.2\text{ MHz}$ TCXO, $f_{SYS} = 52\text{ MHz}$ Crystal, 0.1 Hz DPLL BW, Phase Buildout Mode)



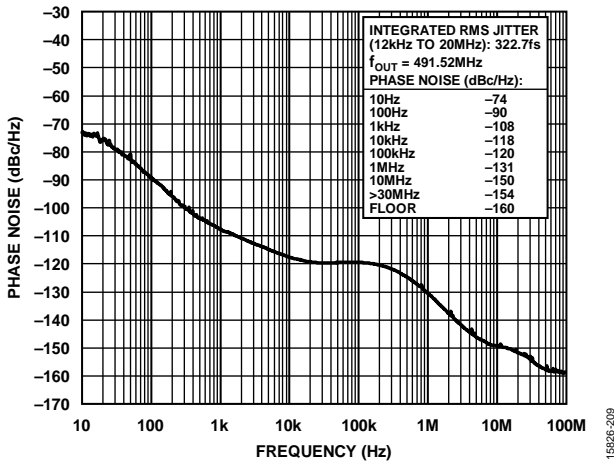
15826-208

Figure 10. Absolute Phase Noise (PLL1, Configuration 2, HCSL Mode, $f_{REF} = 30.72\text{ MHz}$, $f_{OUT} = 245.76\text{ MHz}$, $f_{SYS} = 52\text{ MHz}$ Crystal, 50 Hz DPLL BW)



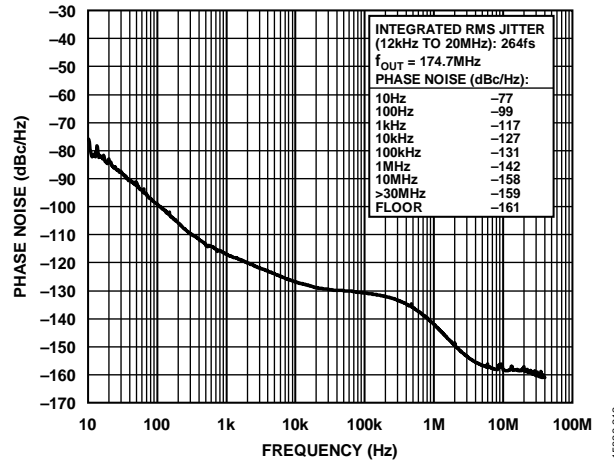
15826-211

Figure 13. Absolute Phase Noise (PLL1, Configuration 5, HCSL Mode, $f_{REF} = 25\text{ MHz}$, $f_{OUT} = 312.5\text{ MHz}$, $f_{SYS} = 52\text{ MHz}$ Crystal, 50 Hz DPLL BW, Phase Buildout Mode)



15826-209

Figure 11. Absolute Phase Noise (PLL1, Configuration 3, HCSL Mode, $f_{REF} = 1\text{ Hz}$, $f_{OUT} = 491.52\text{ MHz}$, $f_{COMP} = 19.2\text{ MHz}$ TCXO, $f_{SYS} = 52\text{ MHz}$ Crystal, 50 MHz DPLL BW)



15826-212

Figure 14. Absolute Phase Noise (PLL1, Configuration 6, HCSL Mode, $f_{REF} = 155.52\text{ MHz}$, $f_{OUT} = 174.7\text{ MHz}$, $f_{SYS} = 52\text{ MHz}$ Crystal, 50 Hz DPLL BW, Phase Buildout Mode)

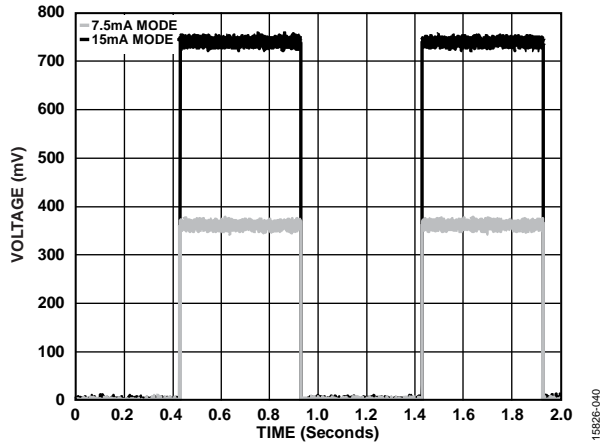


Figure 15. DC-Coupled, Single-Ended, 1 Hz Output Waveforms Using HCSL 7.5 mA and 15 mA Mode Terminated 50 Ω to GND per Figure 38; Slew Rate: ~7 V/ns for 15 mA Mode; ~3.5 V/ns for 7.5 mA Mode

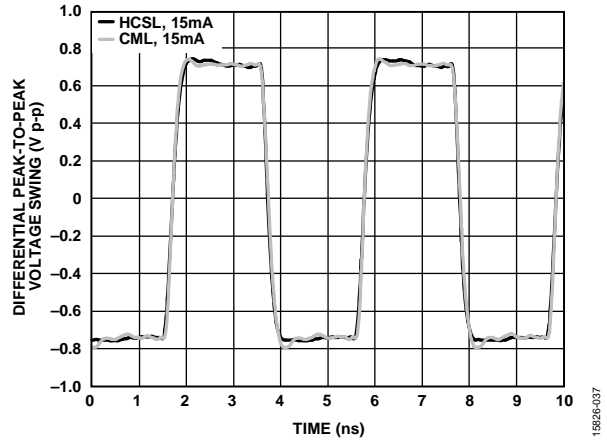


Figure 18. 245.76 MHz Output Waveform for 15 mA Driver Settings; HCSL Drivers Terminated 50 Ω to GND per Figure 32; CML Drivers Terminated 50 Ω to 1.8 V per Figure 33

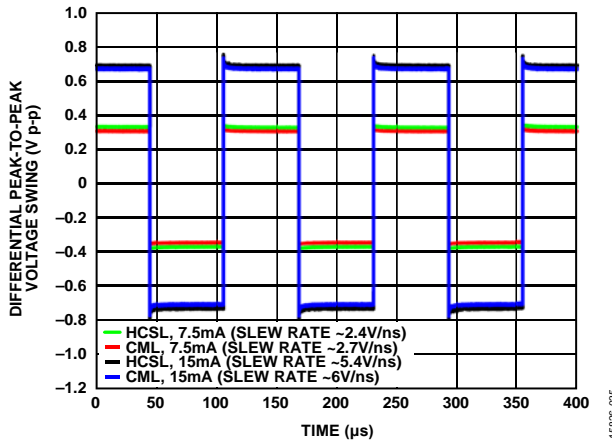


Figure 16. 8 kHz Output Waveforms for Various Driver Settings; HCSL Drivers Terminated 50 Ω to GND per Figure 32; CML Drivers Terminated 50 Ω to 1.8 V per Figure 33

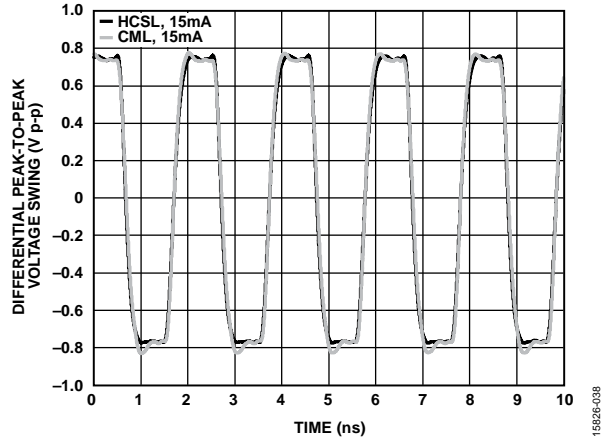


Figure 19. 491.52 MHz Output Waveform for 15 mA Driver Settings; HCSL Drivers Terminated 50 Ω to GND per Figure 32; CML Drivers Terminated 50 Ω to 1.8 V per Figure 33

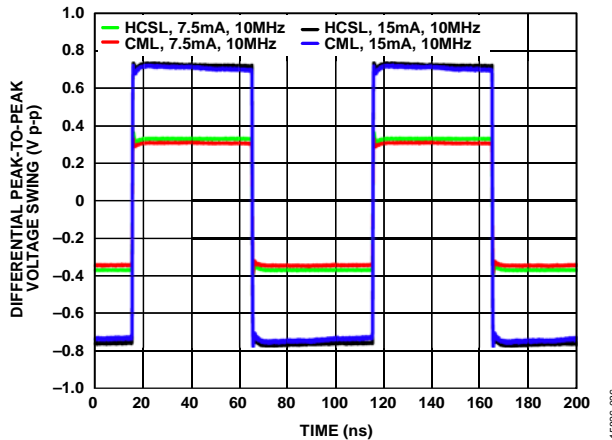


Figure 17. 10 MHz Output Waveforms for Various Driver Settings; HCSL Drivers Terminated 50 Ω to GND per Figure 32; CML Drivers Terminated 50 Ω to 1.8 V per Figure 33

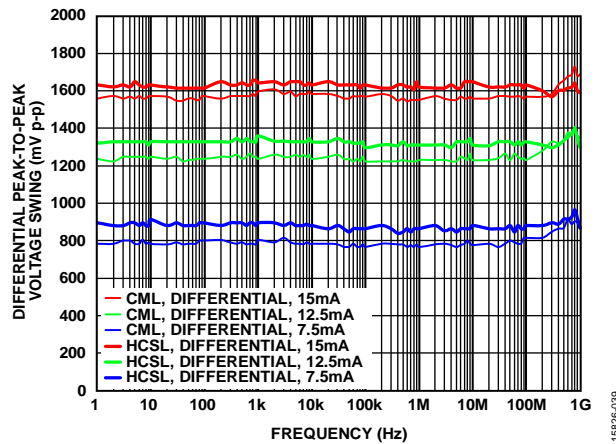


Figure 20. Differential Output Amplitude Waveforms; HCSL Drivers Terminated 50 Ω to GND per Figure 32; CML Drivers Terminated 50 Ω to 1.8 V per Figure 33

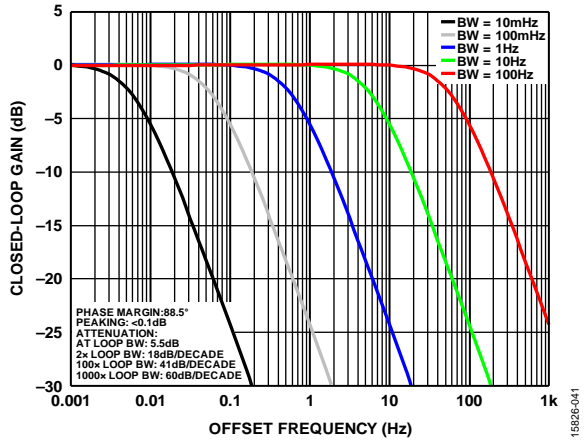


Figure 21. DPLL Closed-Loop Transfer Function Nominal Phase Margin Loop Filter Setting

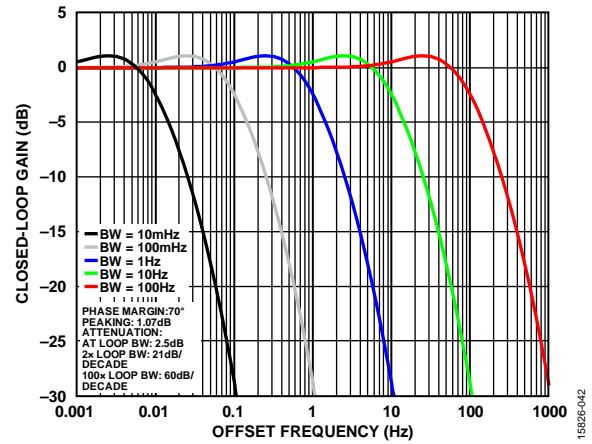


Figure 22. DPLL Closed-Loop Transfer Function High Phase Margin Loop Filter Setting

TERMINOLOGY

Zero Delay

Zero delay is seen in an integer-N PLL architecture that establishes zero (or nearly zero, but constant) phase offset between the final output signal and the signal appearing at the reference input of the PLL phase detector. A PLL with zero delay provides minimal input to output phase offset in the static (steady state) sense. That is, phase slewing at the output typically occurs any time the PLL is in the process of phase or frequency acquisition (for example, when a multiple input PLL switches from one input reference signal to another).

Hitless Switchover

Hitless switchover applies to PLLs with the ability to switch from one reference signal to another while maintaining a constant phase relationship from the active input to output. Hitless switchover is the ability of a PLL to switch between reference signals having an arbitrary initial instantaneous phase offset. In hitless switching, the output signal slews in a prescribed manner from its initial phase to the new phase, and the absolute phase relationship from active input to output is maintained. The reference switching scheme is hitless if the phase slewing is gradual enough to not cause traffic hits caused by the output clock phase slewing. A PLL employing hitless switchover capability requires the output/input frequency ratio to be an integer greater than or equal to 1. Hitless output phase transient limitation applies any time the PLL is in the process of phase or frequency acquisition (that is, it is not necessarily limited to reference switching).

Phase Buildout (PBO) Switchover

PBO only applies to PLLs with the ability to switch from one reference signal to another. PBO is the ability of a PLL to switch between two reference signals having an arbitrary initial instantaneous phase offset, whereby the phase of the output signal remains fixed. This mode of operation implies the ability of the PLL to absorb the phase difference between the two reference input signals, the goal being to prevent a phase disturbance at the output when switching between two reference signals. Prevention of a phase disturbance at the output means there is no guarantee of phase alignment between the input and output signals. Unlike hitless switchover, PBO places no restriction on the output/input frequency ratio. PBO output phase transient prevention applies any time the PLL is in the process of phase or frequency acquisition (that is, it is not necessarily limited to reference switching).

For more information, see the [AN-1420 Application Note, Phase Buildout and Hitless Switchover with Digital Phase-Locked Loops \(DPLLs\)](#).

THEORY OF OPERATION

OVERVIEW

The AD9542 provides clocking outputs that are directly related in phase and frequency to the selected (active) reference but with jitter characteristics governed by the system clock, the DCO, and the analog output PLL (APLL). The AD9542 supports up to four reference inputs and input frequencies ranging from 2 kHz to 750 MHz. The cores of this device are two DPLLs. Each DPLL has a programmable digital loop filter that greatly reduces jitter transferred from the active reference to the output, and these four DPLLs operate completely independently of each other. The AD9542 supports both manual and automatic holdover. While in holdover, the AD9542 continues to provide an output as long as the system clock is present. The holdover output frequency is a time average of the output frequency history prior to the transition to the holdover condition. The device offers manual and automatic reference switchover capability if the active reference is degraded or fails completely. The AD9542 includes a system clock multiplier and two DPLLs, each cascaded with its own APLL.

The input signal goes first to the DPLL, which performs the jitter cleaning and most of the frequency translation. Each DPLL features a 48-bit DCO output that generates a signal in the range of 162 MHz to 350 MHz.

The DCO output goes to the APLL, which multiplies the signal up to a range of 2.424 GHz to 3.232 GHz (for Channel 0) or 3.232 GHz to 4.040 GHz (for Channel 1). After division by 2, this signal is sent to the clock distribution section, which consists of the 32-bit Q divider and output driver for each output. Channel 0 has six Q dividers and Channel 1 has four Q dividers.

The XOA and XOB inputs provide the input for the system clock. These pins accept a reference clock in the 20 MHz to 300 MHz range or a 25 MHz to 52 MHz crystal connected directly across the XOA and XOB inputs. The system clock provides the clocks to the frequency monitors, the DPLLs, and internal switching logic.

The AD9542 has five differential output drivers. Each of the five output drivers has a dedicated 32-bit programmable Q divider. Each differential driver operates up to 500 MHz and is configurable as a CML driver with external pull-up resistors, or an HCSL driver with external pull-down resistors. There are three drive strengths:

- The 7.5 mA mode is used for CML and HCSL and ac-coupled LVDS. When used as an LVDS-compatible driver, it must be ac-coupled and terminated with a 100 Ω resistor across the differential pair.
- The 15 mA mode produces a voltage swing and is compatible with LVPECL. If LVPECL dc signal levels are required, the designer must ac couple and rebias the AD9542 output. The 15 mA mode can also be used with the termination scheme shown in Figure 34 and Figure 35 to produce an LVDS signal with the correct LVDS dc bias.
- The 12 mA mode is halfway in between the two other settings.

REFERENCE INPUT PHYSICAL CONNECTIONS

Two pairs of pins (REFA/REFAA and REFB/REFBB) provide access to the reference clock receivers. The user can reconfigure each differential pair into two single-ended reference inputs. To accommodate input signals with slow rising and falling edges, both the differential and single-ended input receivers employ hysteresis. Hysteresis also ensures that a disconnected or floating input does not cause the receiver to oscillate.

When configured for differential operation, the input receivers accommodate either ac-coupled or dc-coupled input signals. If the input receiver is configured for dc-coupled LVDS mode, the input receivers are capable of accepting dc-coupled LVDS signals. The receiver is internally dc biased to handle ac-coupled operation; however, there is no internal 50 Ω or 100 Ω termination.

INPUT/OUTPUT TERMINATION RECOMMENDATIONS

SYSTEM CLOCK INPUTS

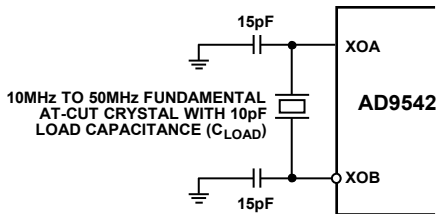


Figure 23. System Clock Input (XOA/XOB) in Crystal Mode (Each 15 pF Shunt Capacitor Shown Must Equal $2 \times (C_{LOAD} - C_{STRAY})$, Where Typical $C_{STRAY} = 2$ pF to 5 pF)

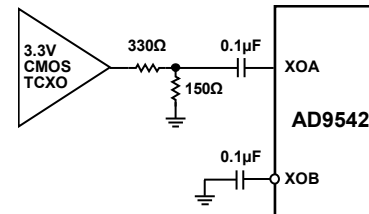


Figure 24. System Clock Input (XOA, XOB) when Using a TCXO/OCXO with 3.3 V CMOS Output

REFERENCE CLOCK INPUTS

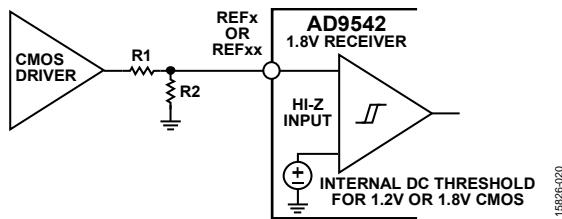


Figure 25. Single-Ended DC-Coupled Mode, 1.2 V or 1.8 V CMOS

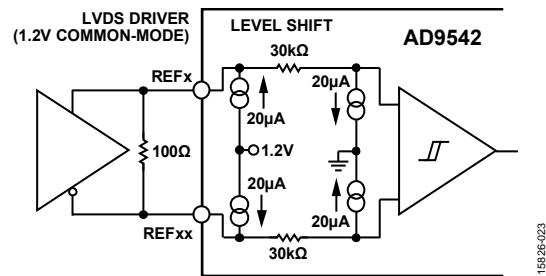


Figure 28. Differential LVDS Input Mode

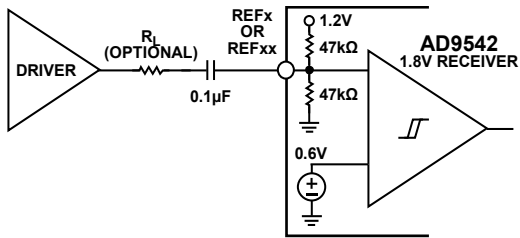


Figure 26. Single-Ended AC-Coupled Mode

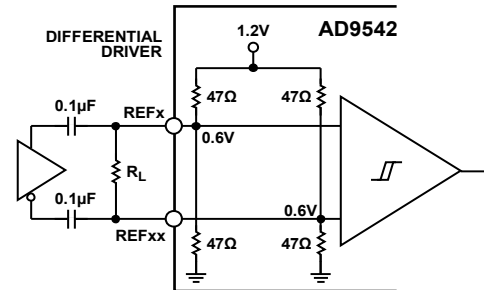


Figure 29. Differential AC-Coupled Mode ($R_L = 100 \Omega$ Is Recommended, Except For HCSL)

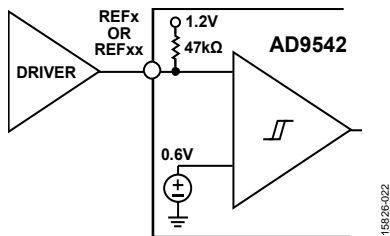


Figure 27. Single-Ended Internal Pull-Up Mode

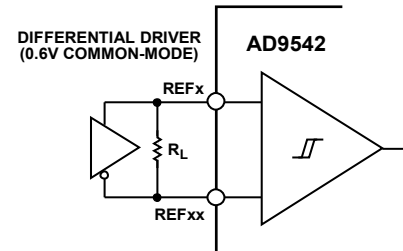


Figure 30. Differential DC-Coupled Mode

CLOCK OUTPUTS

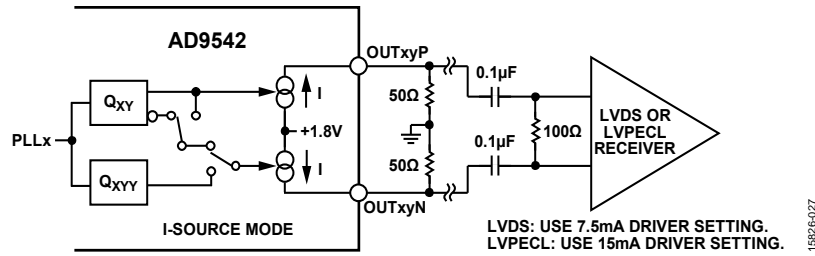


Figure 31. LVDS-Compatible Output Swing, AC-Coupled ($V_{p-p} \approx 375 \text{ mV}$ per Section for $I = 15 \text{ mA}$)

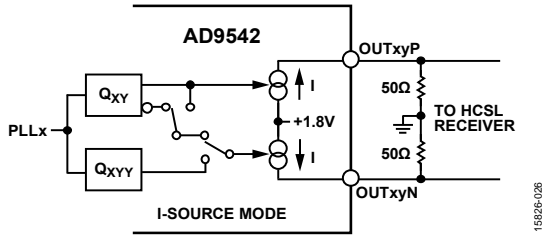


Figure 32. HCSL Output, $V_{p-p} \approx 750 \text{ mV}$ per Section ($I = 15 \text{ mA}$)

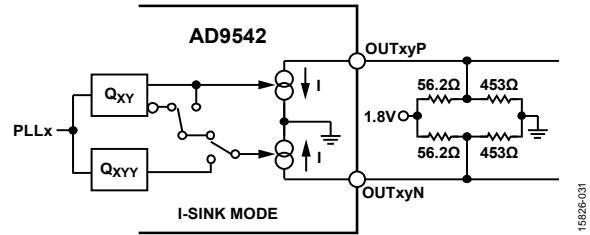


Figure 36. 2.5 V LVPECL or Double Amplitude LVDS-Compatible Boost Output, 1.5 V_{p-p} , 1.24 V Common-Mode ($I = 15 \text{ mA}$)

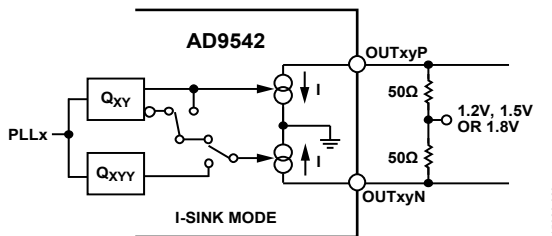


Figure 33. CML Output ($I = 7.5 \text{ mA}$; $I = 15 \text{ mA}$ Options for 1.5 V or 1.8 V Supply)

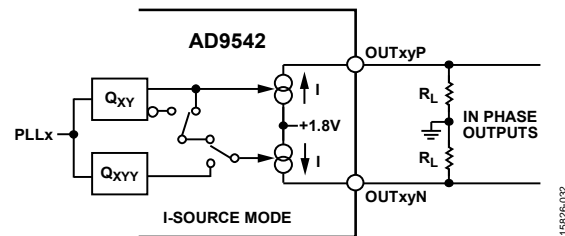


Figure 37. Single-Divider, Single-Ended Mode Providing In-Phase Outputs (Current Source Mode)

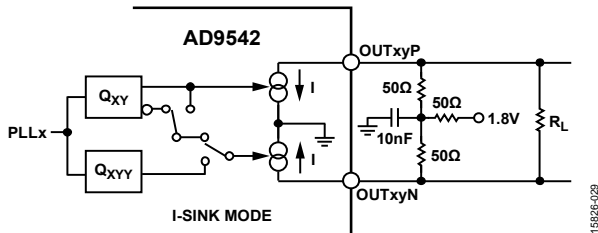


Figure 34. LVDS-Compatible Output, 1.24 V Common-Mode, T Network ($I = 7.5 \text{ mA}$; $I = 15 \text{ mA}$ With Extra 100Ω Termination, R_L)

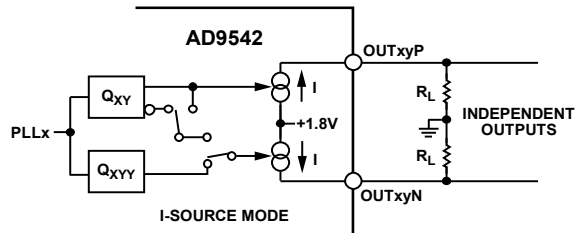


Figure 38. Dual-Divider, Single-Ended Mode Providing Independent Outputs (Current Source Mode); Note that Single-Ended CML Mode Is Also Available (See Figure 33)

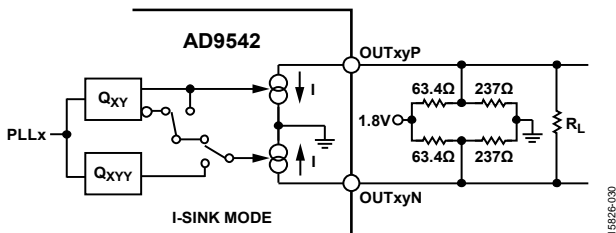


Figure 35. LVDS-Compatible Output, 1.2 V Common-Mode, Thevenin Bias Network ($I = 7.5 \text{ mA}$; 15 mA With Extra 100Ω Termination, R_L)

SYSTEM CLOCK PLL

Note that throughout the System Clock PLL section, unless otherwise specified, any referenced bits, registers, or bit fields reside in the system clock (SYSCLK) section of the register map (Register 0x0200 to Register 0x0209).

The system clock PLL (see Figure 39) comprises an integer-N frequency synthesizer with a fully integrated loop filter and voltage controlled oscillator (VCO). The VCO output is the AD9542 system clock with a frequency range of 2250 MHz to 2415 MHz. The XOA and XOB pins constitute the input to the system clock PLL to which a user connects a clock source or crystal resonator.

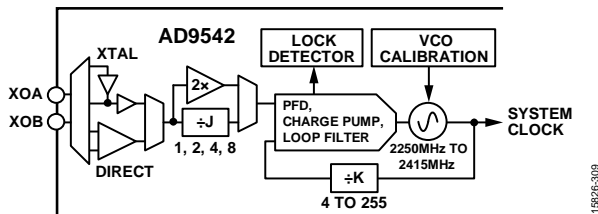


Figure 39. System Clock PLL Block Diagram

SYSTEM CLOCK INPUT FREQUENCY DECLARATION

Proper operation of the AD9542 requires the user to declare the input reference frequency to the system clock PLL. To do so, program the SYSCLK reference frequency bit field, which constitutes the nominal frequency of the system clock PLL input reference. The AD9542 evaluation software frequency planning wizard calculates this value for the user.

SYSTEM CLOCK SOURCE

The XOA and XOB pins serve as the input connection to the system clock PLL, giving the user access to a crystal path or a direct path. Path selection is via the enable maintaining amplifier bit, where a Logic 0 (default) selects the direct path and Logic 1 selects the crystal path. The optimal reference source for the system clock input is a crystal resonator in the 50 MHz range or an ac-coupled square wave source (single-ended or differential) with 800 mV p-p amplitude.

Crystal Path

The crystal path supports crystal resonators in the 25 MHz to 60 MHz frequency range. An internal maintaining amplifier provides the negative resistance required to induce oscillation. The internal amplifier expects an AT cut, fundamental mode crystal with a maximum motional resistance of 100 Ω for crystals up to 52 MHz, and 50 Ω for crystals up to 60 MHz. The following crystals, listed in alphabetical order, may meet these criteria.

- AVX/Kyocera CX3225SB
- ECS, Inc. ECX-32
- Epson/Toyocom TSX-3225
- Fox FX3225BS
- NDK NX3225SA
- Siward SX-3225
- Suntuu SCM10B48-49.152 MHz

Analog Devices, Inc., does not guarantee the operation of the AD9542 with the aforementioned crystals, nor does Analog Devices endorse one crystal supplier over another. The AD9542 reference design uses a readily available high performance 49.152 MHz crystal with low spurious content.

Direct Path

The direct path has a differential receiver with a self bias of 0.6 V dc. Generally, the presence of the bias voltage necessitates the use of ac coupling between the external source and the XOA and XOB pins. Furthermore, when using a 3.3 V CMOS oscillator as the system clock PLL reference source, in addition to ac coupling, it is important to use a voltage divider to reduce the 3.3 V swing to a maximum of 1.14 V (note that the optimal voltage swing is 800 mV p-p). The external signal must exhibit a 50% duty cycle for best performance.

The direct path supports low frequency LVPECL, LVDS, CMOS, or sinusoidal clock sources as a reference to the system clock PLL. For a sinusoidal source, however, it is best to use a frequency of 50 MHz or greater. The low slew rate of lower frequency sinusoids tends to yield nonoptimal noise performance.

Applications requiring low DPLL loop bandwidth require the improved stability provided by a TCXO or OCXO. Loop bandwidths below approximately 50 Hz may prevent the PLL from locking or cause random loss of lock events when using a less stable PLL reference source.

Although one method to mitigate this problem is to use a high stability system clock source (like an OCXO), the AD9542 provides integrated system clock compensation capability, which lessens the stability requirement on the system clock while providing the outstanding phase noise of the higher frequency crystal. To use this feature, connect a 40 MHz to 60 MHz crystal to the XOA/XOB pins (as in Figure 23), and connect either a TCXO or OCXO to either an unused reference input or an Mx pin (as shown in Figure 24).

2x FREQUENCY MULTIPLIER

The system clock PLL provides the user with the option of doubling the reference frequency via the enable SYSCLK doubler bit. Doubling the input reference frequency potentially reduces the PLL in-band noise. The reference frequency must be less than 150 MHz when using the 2x frequency multiplier to satisfy the 300 MHz maximum PFD rate. Furthermore, the 2x frequency multiplier requires the reference input signal to have very near to 50% duty cycle; otherwise, the resulting spurious content may prevent the system clock PLL from locking.

PRESCALE DIVIDER

The system clock PLL includes an input prescale divider programmable for divide ratios of 1 (default), 2, 4, or 8. The purpose of the divider is to provide flexible frequency planning for mitigating potential spurs in the output clock signals of the AD9542. The user selects the divide ratio via the 2-bit SYSCLK

input divider ratio bit field. The corresponding divide value is 2^J , where J is the decimal value of the 2-bit number in the SYSCLK input divider ratio bit field.

For example, given that the SYSCLK input divider ratio bit field is 10 (binary), $J = 2$ (decimal), yielding a divide ratio of $2^J = 2^2 = 4$.

FEEDBACK DIVIDER

The output of the system clock PLL constitutes the system clock frequency, f_s . The system clock frequency depends on the value of the feedback divider. The feedback divide ratio has a range of 4 to 255, which the user programs via the 8-bit feedback divider ratio register (the register value is the divide ratio). For example, a programmed value of 100 (0x64 hexadecimal) yields a divide ratio of 100.

SYSTEM CLOCK PLL OUTPUT FREQUENCY

Calculate the system clock frequency as follows:

$$f_s = f_{\text{SYSIN}} \times \frac{K}{J}$$

where:

f_{SYSIN} is the input frequency.

K is the feedback divide ratio.

J is the input divide ratio. $J = \frac{1}{2}$ when using the $2\times$ frequency multiplier.

The user must choose f_{SYSIN} , K , and J such that f_s satisfies the VCO range of 2250 MHz to 2415 MHz.

SYSTEM CLOCK PLL LOCK DETECTOR

The system clock PLL features a simple lock detector that compares the time difference between the reference and feedback clock edges. The user can check the status of the lock

detector via the SYSCLK locked bit in the status readback registers (Address 0x3000 to Address 0x300A) of the register map, where Logic 1 indicates locked and Logic 0 unlocked. The most common reason the system clock PLL fails to lock is due to the user employing the $2\times$ frequency multiplier with a reference input clock that deviates from the 50% duty cycle.

SYSTEM CLOCK STABILITY TIMER

Because time processing blocks within the AD9542 depend on the system clock generating a stable frequency, the system clock PLL provides an indication of its status. The status of the system clock PLL is available to the user as well as directly to certain internal time keeping blocks.

At initial power-up, the system clock status is unknown and reported as being unstable. However, after the user programs the system clock registers and the system clock PLL VCO calibrates, the system clock PLL locks shortly thereafter.

SYSTEM CLOCK INPUT TERMINATION RECOMMENDATIONS

To connect a crystal resonator to the system clock PLL XOA and XOB inputs, refer to Figure 23. Be sure to program the enable maintaining amplifier bit = 1 to select the crystal path. The 15 pF shunt capacitors shown relate to the C_{LOAD} and C_{STRAY} associated with the crystal as follows:

$$C_{\text{SHUNT}} = 2 \times (C_{\text{LOAD}} - C_{\text{STRAY}})$$

For $C_{\text{LOAD}} = 10$ pF and $C_{\text{STRAY}} = 2$ pF to 5 pF, the value of C_{SHUNT} is approximately 15 pF.

To connect a TCXO or OCXO with a 3.3 V output, refer to Figure 24. Be sure to program the enable maintaining amplifier bit = 0 to select the direct path.

DIGITAL PLL (DPLL)

OVERVIEW

Note that throughout this section, unless otherwise specified, any referenced bits, registers, or bit fields reside in the DPLL Channel 0 and DPLL Channel 1 sections (Address 0x1000 to Address 0x102A, and Address 0x1400 to Address 0x142A) of the register map.

The DPLL is an all-digital implementation of a phase-locked loop (PLL). Figure 40 shows the fundamental building blocks of an APLL and a DPLL. An APLL typically relies on a VCO as the frequency element for generating an output signal, where the output frequency depends on an applied dc voltage. A DPLL, on the other hand, uses a numerically controlled oscillator (NCO), which relies on a digital frequency tuning word (FTW) to produce the output frequency. A VCO inherently produces a timing signal because it is, by definition, an oscillator, whereas the AD9542 NCO requires an external timing source, the system clock. The fundamental difference between an APLL and a DPLL is that the VCO in an APLL can tune to any frequency within its operating bandwidth, whereas the NCO in a DPLL can only tune to discrete frequencies (by virtue of the FTW).

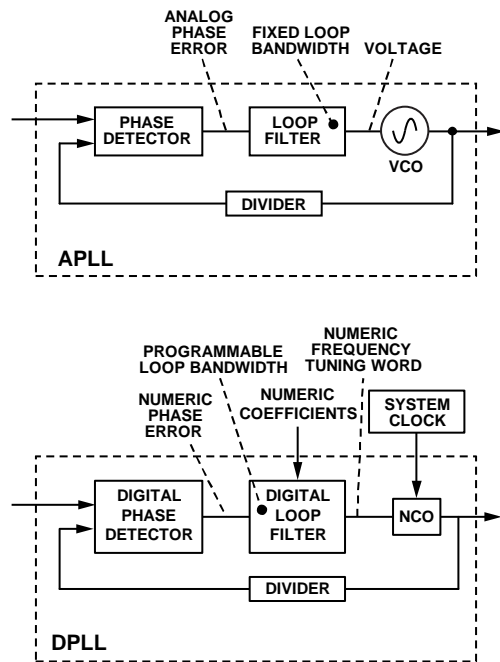


Figure 40. APLL vs. DPLL

The DPLLs in the AD9542 have a digital TDC-based phase detector and a digital loop filter with programmable bandwidth. The digital loop filter output yields a digital FTW (instead of a dc voltage, as in the case of an analog PLL) that produces a corresponding NCO output frequency.

DPLL PHASE/FREQUENCY LOCK DETECTORS

See the Lock Detectors section for details concerning the phase and frequency detectors of the DPLL.

DPLL LOOP CONTROLLER

The DPLL has several operating modes (including freerun, holdover, and active). To ensure seamless transition between modes, the DPLL has a loop controller. The loop controller sets the appropriate DPLL operating mode based on the prevailing requirements of automatic reference switching or manual control settings.

Switchover

Switchover occurs when the loop controller switches directly from one input reference to another. The AD9542 handles a reference switchover by briefly entering holdover mode, loading the new DPLL parameters, and then immediately recovering.

Holdover

Typically, the holdover state is in effect when all of the input references are invalid. However, the user can force the holdover mode even when one or more references are valid by setting the DPLLx force holdover bit (where x = 0 or 1) in the Operational Control Channel 0 and Operational Control Channel 1 sections of the register map to Logic 1. In holdover mode, the output frequency remains fixed (to the extent of the stability of the system clock). The accuracy of the AD9542 in holdover mode is dependent on the device programming and availability of the tuning word history.

Recovery from Holdover

When in holdover and an enabled translation profile becomes available, the device exits holdover operation. The loop controller restores the DPLL to closed-loop operation, locks to the selected reference, and sequences the recovery of all the loop parameters based on the profile settings for the active reference.

If the DPLLx force holdover bit (where x = 0 or 1) in the Operational Control Channel 0 and Operational Control Channel 1 sections of the register map is set to Logic 1, the device does not automatically exit holdover when a valid translation profile is available. However, automatic recovery can occur after clearing the DPLL force holdover bit.

APPLICATIONS INFORMATION

OPTICAL NETWORKING LINE CARD

In this application (shown in Figure 41), the AD9542 is used in a variety of ways.

In a loop timed (WAN) mode, one of the AD9542 DPLLs locks to the CDR and is used to remove jitter on the receiver path, sending that clock to the central timing card, as well as the framer. In some applications, the AD9542 can also perform a variety of frequency translation tasks, such as multiplying or dividing by an FEC ratio, and/or removing jitter from a gapped

clock. The other DPLL cleans jitter and provides clocking to the transmitter path.

Other tasks include frequency translation and jitter cleaning of the reference clock from the timing card, as well as seamlessly managing the reference switching from Timing Card A to Timing Card B.

Given the continually evolving nature of optical line card protocols and functions, the functions listed in this section are by no means exhaustive.

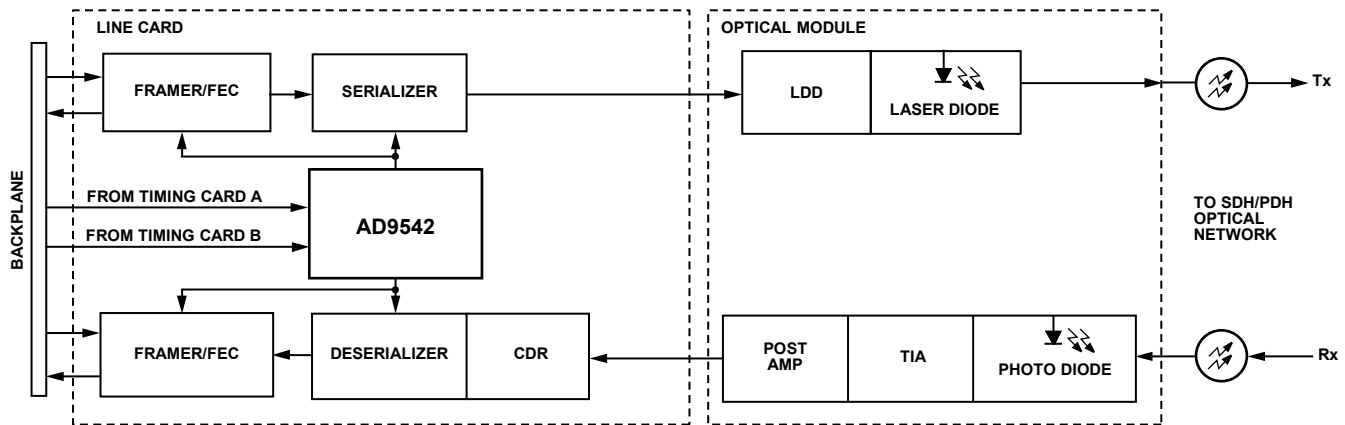


Figure 41. Optical Line Card Example

18526-043

SMALL CELL BASE STATION

In this application (shown in Figure 42), the AD9542 provides all of the synchronization to the baseband unit of a small cell base station. The built in JESD204B support enables a particularly compact and efficient design.

The AD9542 can lock to any of the following: GPS, SyncE, and/or IEEE1588 (requires a separate IEEE1588 servo and software stack) or loop timed (if using SONET/SDH backhaul).

The PLL0 of the AD9542 provides one device clock and up to four device system reference clocks that can be used to clock wireless transceivers, such as the AD9371.

The PLL1 of the AD9542 clocks the backhaul interface and, optionally, the CPU interface.

The EEPROM support of the AD9542 allows the AD9542 to load its configuration automatically at power-up.

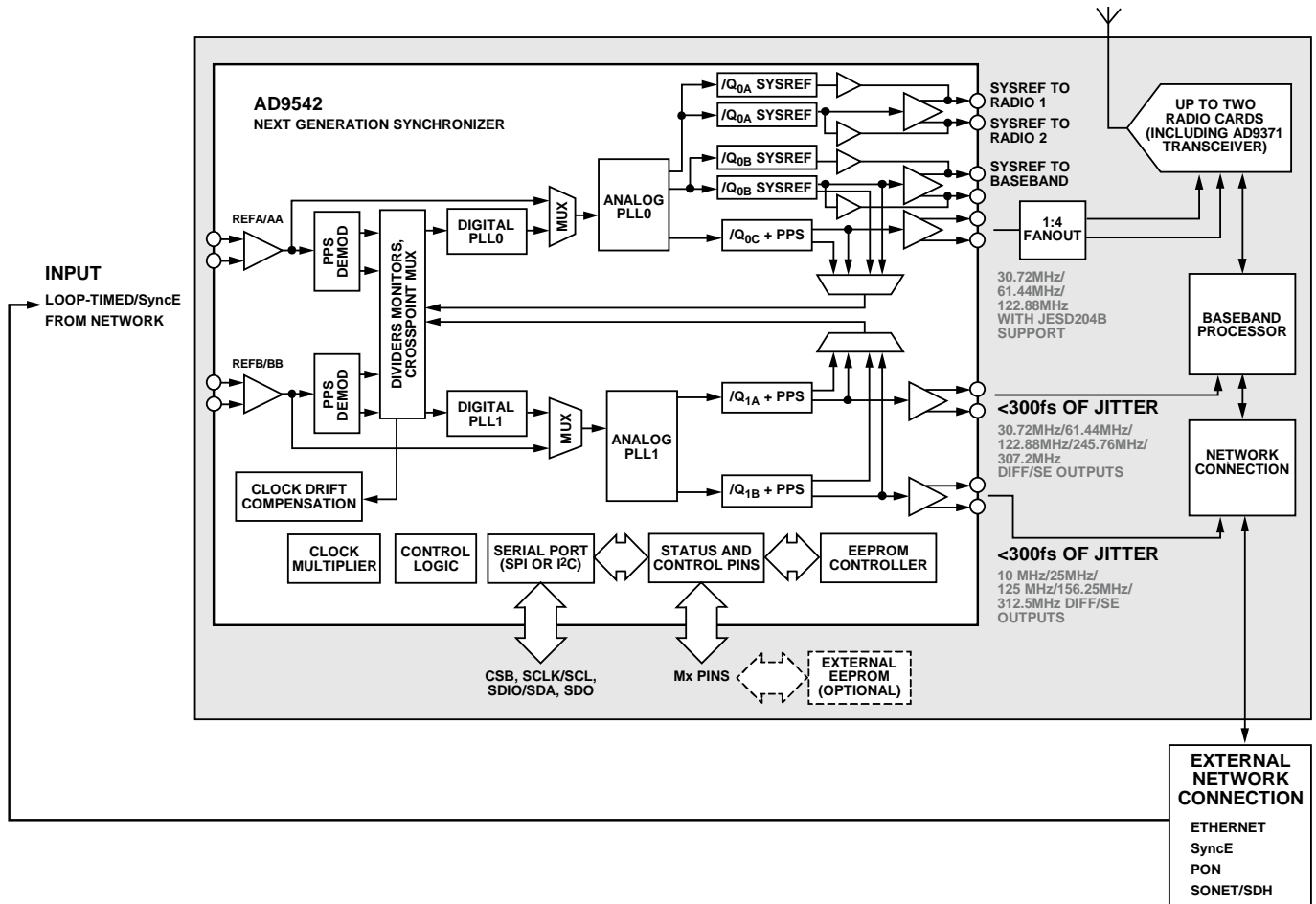


Figure 42. Small Cell Application

INITIALIZATION SEQUENCE

Figure 43, Figure 44, and Figure 45 describe the sequence for powering on and programming the AD9542.

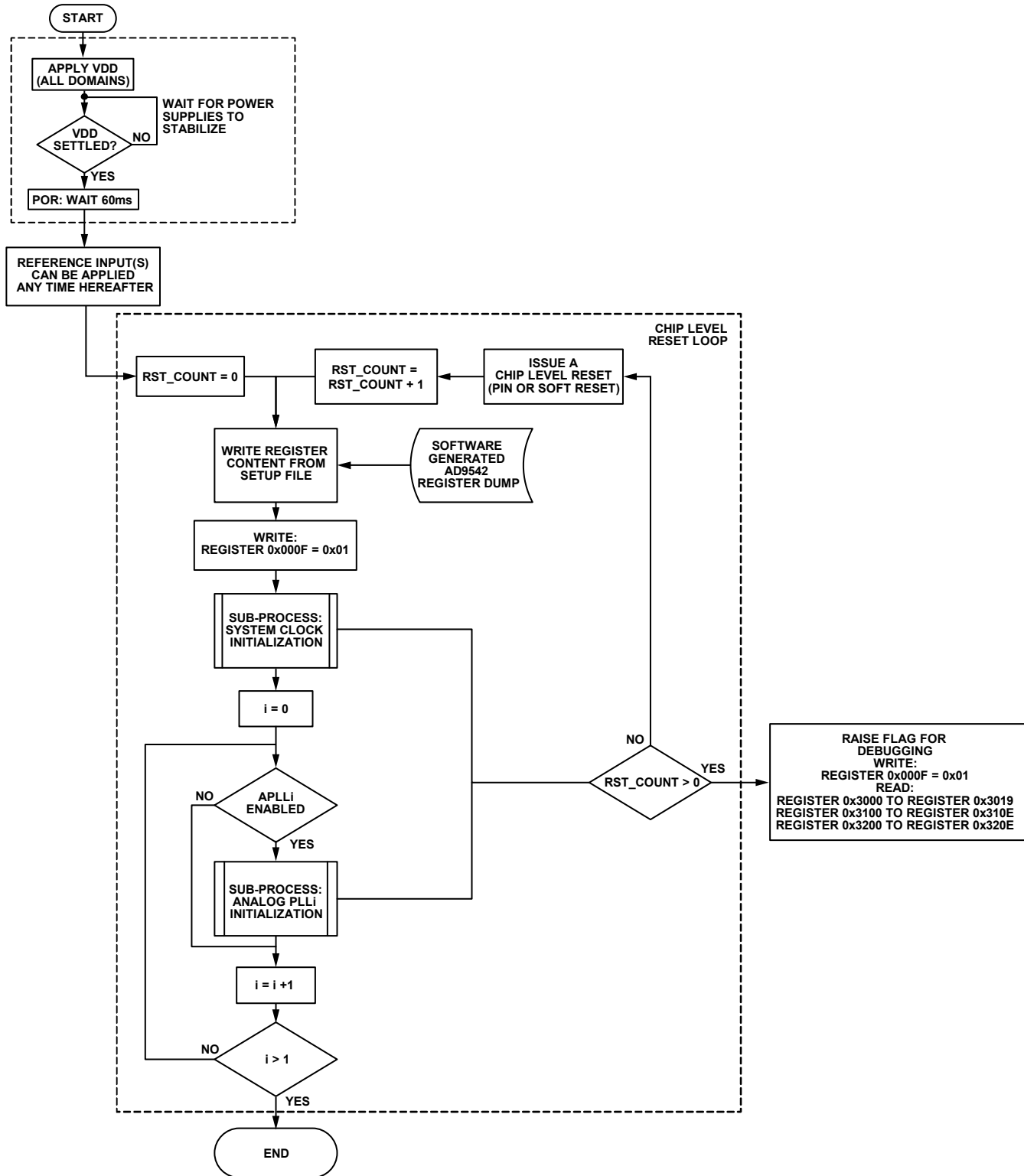
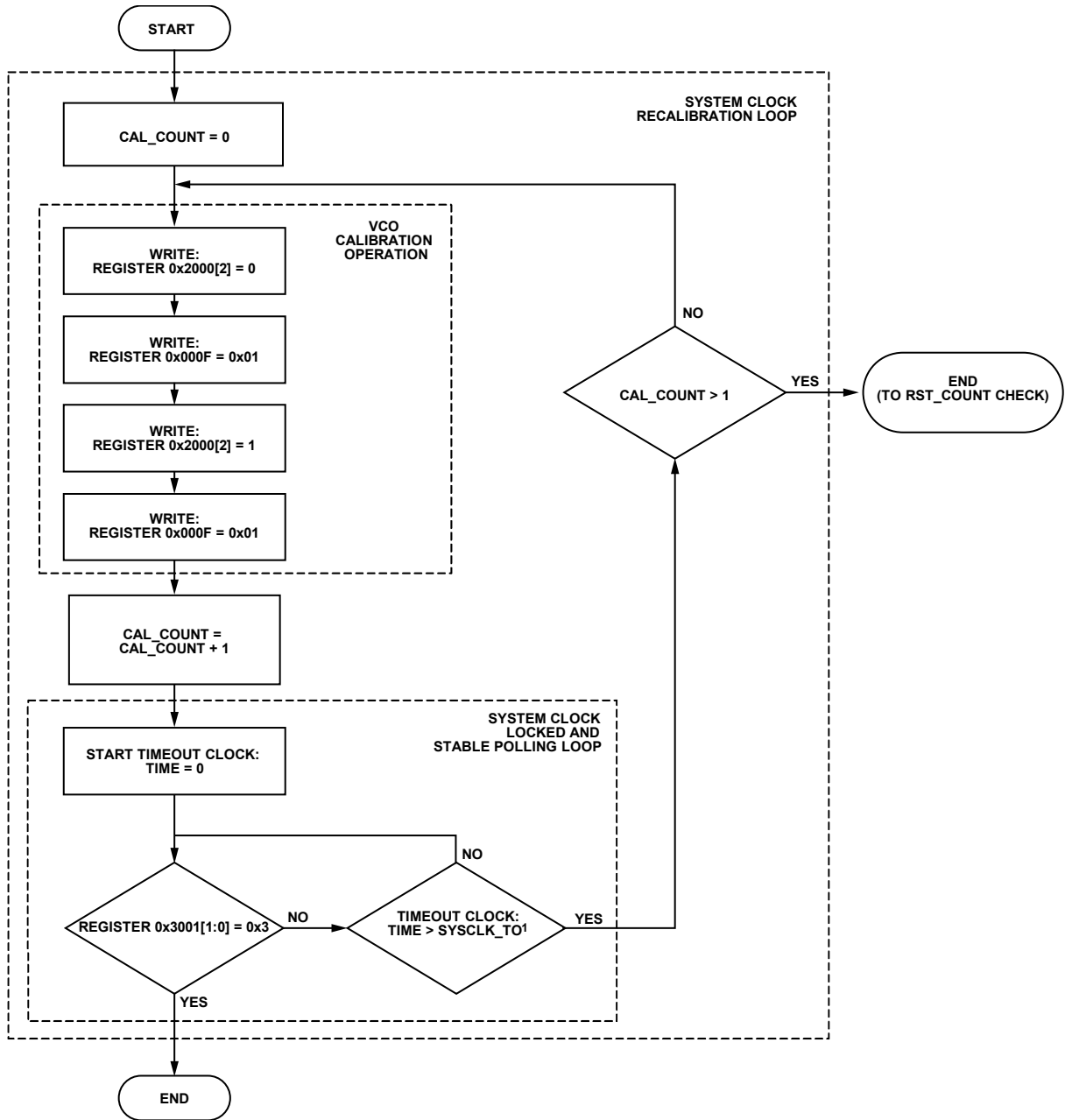


Figure 43. Programming Sequence Loop

15526-101



¹SYSCLK_TO IS A CALCULATED TIME OUT VALUE.
IT IS 50ms + SYSTEM CLOCK VALIDATION TIME (REGISTER 0x0207 TO REGISTER 0x0209 [UNITS OF ms])

Figure 44. System Clock Initialization Subprocess

155526-102

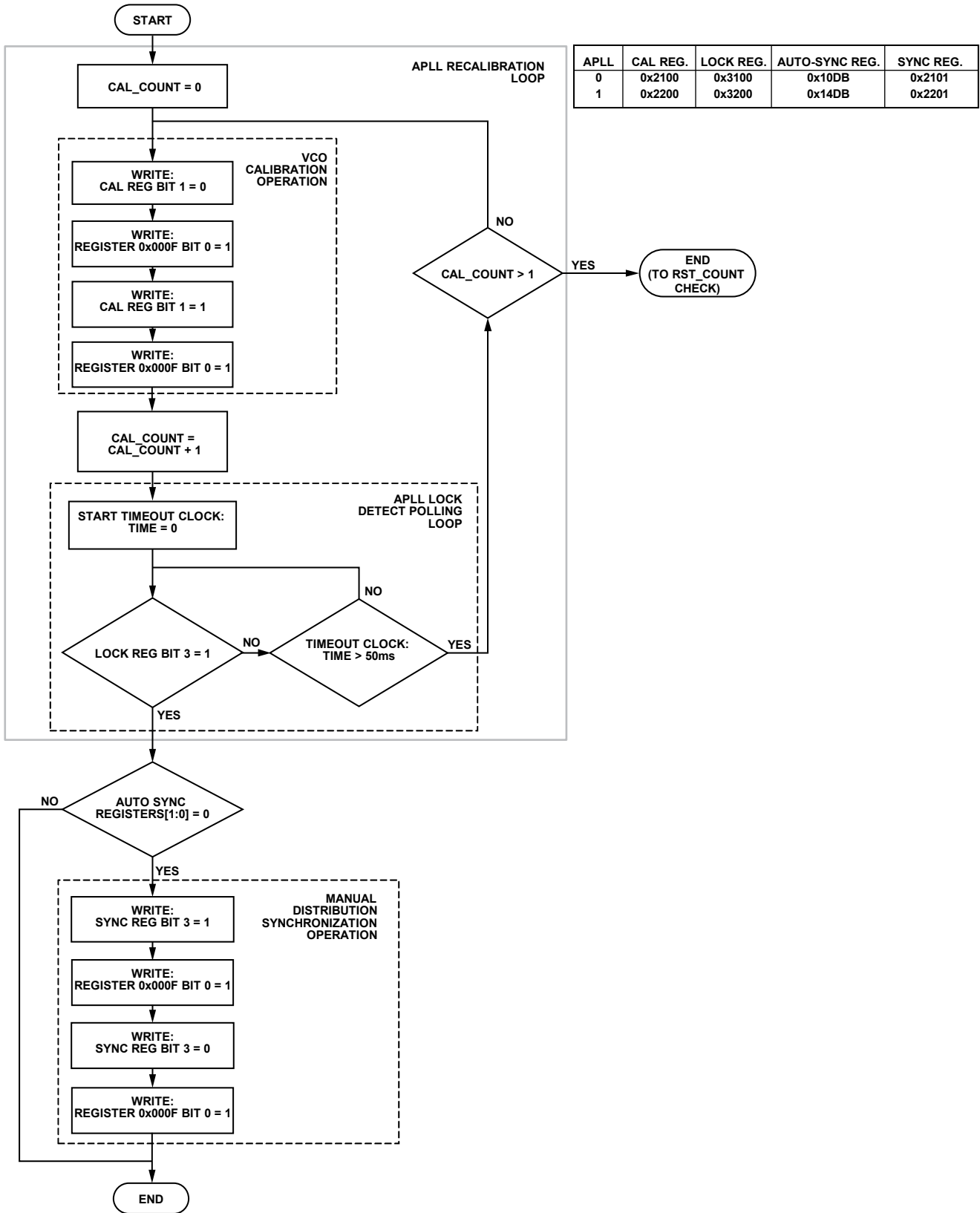


Figure 45. Analog PLL Initialization Subprocess

15826-103

STATUS AND CONTROL PINS

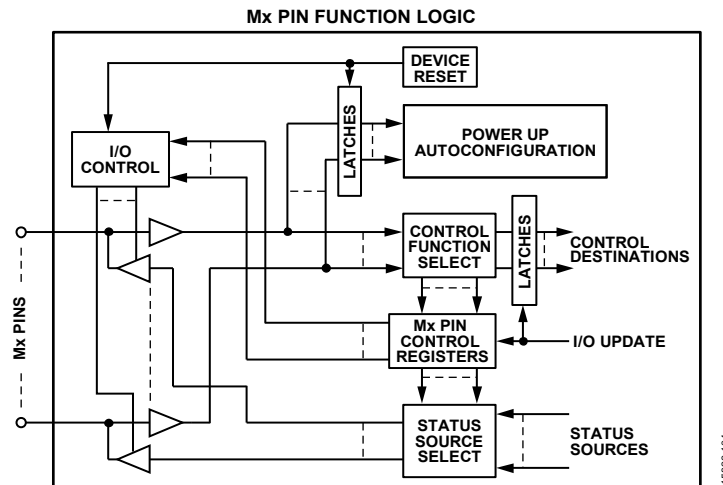


Figure 46. Mx Pin Logic

The AD9542 features seven independently configurable digital CMOS status/control pins (M0 to M6). Configuring an Mx pin as a status pin causes that pin to be an output. Conversely, configuring an Mx pin as a control pin causes that pin to be an input. Register 0x102 to Register 0x108 control both the nature of the pin (either status or control via Bit D7), as well as the selection of the status source or control destination associated with the pin via Bits[D6:D0]. During power-up or reset, the Mx pins temporarily become inputs and only allow the device to autoconfigure. Figure 46 is a block diagram of the Mx pin functionality.

The Mx pin control logic uses special register write detection logic to prevent these pins from behaving unpredictably when the Mx pin function changes, especially when changing mode from input to output or vice versa.

When an Mx pin functions as an output, it continues operating according to the prior function, even after the user programs the corresponding registers. However, assertion of an input/output update causes the corresponding pins to switch to the new function according to the newly programmed register contents. Note that changing from one output function to another output function on an Mx pin does not require special timing to avoid input/output contention on the pin.

When an Mx pin functions as an input, programming a particular Mx pin function register causes all the Mx pin control functions to latch their values. Assertion of an input/output update switches to the newly programmed pin function, at which time normal behavior resumes. Note that, when switching from one input function to another input function on the same pin, the logic state driven at the input to the pin can change freely during the interval between writing the new function to the corresponding register and asserting the input/output update.

When switching the operation of an Mx pin from an input to an output function, the recommendation is that the external drive source become high impedance during the interval between writing the new function and asserting the input/output update.

When switching the operation of an Mx pin from an output to an input, the recommendation is as follows. First, program the Mx pin input function to no operation (NOOP) and assert the input/output update. This configuration avoids input/output contention on the Mx pin or other undesired behavior because, prior to the assertion of the input/output update, the device continues to drive the Mx pin but ignores the logic level on the pin due to the programmed NOOP function. Note that the recommendation is to avoid using a high impedance source on an Mx pin configured as an input because this may cause excessive internal current consumption. Second, drive the Mx pin with Logic 0 or Logic 1 via the desired external source and program the associated Mx pin register from NOOP to the desired function.

MULTIFUNCTION PINS AT RESET/POWER-UP

At power-up or in response to a reset operation, the Mx pins enter a special operating mode. For a brief interval following a power-up or reset operation, the Mx pins function only as inputs (the internal drivers enter a high impedance state during a power-up/reset operation). During this brief interval, the device latches the logic levels at the Mx inputs and uses this information to autoconfigure the device accordingly. The Mx pins remain high-Z until either an EEPROM operation occurs, in which case M1 or M2 become an I²C master, or the user (or EEPROM) programs them to be outputs.

If the user does not connect external pull-up/pull-down resistors to the Mx pins, the M3 and M4 pins have internal pull-down resistors to ensure a predictable start-up configuration. In the absence of external resistors, the internal pull-down resistors ensure that the device starts up with the serial port in SPI mode and without automatically loading data from an external EEPROM (see Table 26). Although the M0, M1, M2, M5, and M6 pins are high impedance at startup, connect external 100 k Ω pull-down or pull-up resistors to these pins to ensure robust operation.

The Mx pin start-up conditions are shown in Table 26. M0, M1, and M2 are excluded from Table 26 because these pins have no explicit function during a power-up or reset operation.

Table 26. Mx Pin Function at Startup or Reset

| Mx Pin | Startup/Reset Function | Logic 1 | Logic 0 |
|--------|---------------------------------|-----------------------|-----------------------------------|
| M3 | EEPROM load function | Load from EEPROM | Do not load from EEPROM (default) |
| M4 | Serial port function | I ² C mode | SPI mode (default) |
| M5 | I ² C address offset | See Table 27 | See Table 27 |
| M6 | I ² C address offset | See Table 27 | See Table 27 |

When the start-up conditions select the serial port to be I²C mode (that is, M4 is Logic 1 at startup), the M5 and M6 pins determine the I²C port device address offset per Table 27. Note that the logic levels in Table 27 only apply during a power-up or reset operation.

Table 27. I²C Device Address Offset

| M6 | M5 | M4 | Address Offset |
|----------------|----------------|----|----------------|
| X ¹ | X ¹ | 0 | Not applicable |
| 0 | 0 | 1 | 1001000 (0x48) |
| 0 | 1 | 1 | 1001001 (0x49) |
| 1 | 0 | 1 | 1001010 (0x4A) |
| 1 | 1 | 1 | 1001011 (0x4B) |

¹ X means don't care.

STATUS FUNCTIONALITY

Configuring an Mx pin as a status pin gives the user access to specific internal device status/IRQ functions in the form of a hardware pin that produces a logic signal. Each Mx pin has a corresponding Mx function register. To assign an Mx pin as a status pin, write a Logic 1 to the Mx output enable bit in the corresponding Mx pin function register.

To assign a specific status/IRQ function to an Mx pin configured as a status pin, program the appropriate 7-bit code (see Table 30) to Bits[D6:D0] of the corresponding Mx function register. See the Interrupt Request (IRQ) section for details regarding IRQ functionality.

When configured as a status pin, the output mode of an Mx pin depends on a 2-bit mode code per Table 28. The 2-bit codes reside in Register 0x100 through Register 0x101, where the 2-bit codes constitute the Mx receiver/driver bit fields. Note that the Mx receiver/driver bit fields perform a different function when the Mx pin is a control pin (see the Control Functionality section).

Table 28. Mx Receiver/Driver Bit Field Codes for Mx Status Pins

| Code | Mode | Description |
|------|-------------------|---|
| 00 | CMOS, active high | Output is Logic 0 when deasserted and Logic 1 when asserted (default operating mode). |
| 01 | CMOS, active low | Output is Logic 1 when deasserted and Logic 0 when asserted. |
| 10 | PMOS, open drain | Output is high impedance when deasserted and active high when asserted. |
| 11 | NMOS, open drain | Output is high impedance when deasserted and active low when asserted. |

The PMOS open-drain mode requires an external pull-down resistor. The NMOS open-drain mode requires an external pull-up resistor. Note that the open-drain modes enable the implementation of wire-ORed functionality of multiple Mx status pins (including Mx status pins across multiple AD9542 devices or other compatible devices—for example, to implement an IRQ bus).

The drive strength of an Mx status pin is programmable via the corresponding Mx configuration bits (Bits[D6:D0] of the pin drive strength register). Logic 0 (default) selects normal drive strength (~6 mA) and Logic 1 selects weak drive strength (~3 mA).

CONTROL FUNCTIONALITY

Configuring an Mx pin as a control pin gives the user control of the specific internal device functions via an external hardware logic signal. Each Mx pin has a corresponding Mx function register. To assign an Mx pin as a control pin, write a Logic 0 to the Mx output enable bit in the corresponding Mx function register.

To assign an Mx control pin to a specific function, program the appropriate 7-bit code (see Table 30) to Bits[D6:D0] of the corresponding Mx function register. See the Interrupt Request (IRQ) section for details regarding IRQ functionality.

When configured as an Mx control pin, the logical level applied to the Mx pin translates to the selected device function. It is also possible to assign multiple Mx control pins to the same control function with the multiple pins implementing a Boolean expression. The Boolean operation associated with an Mx control pin depends on a 2-bit code per Table 29. The 2-bit codes reside in Register 0x100 through Register 0x101, where the 2-bit codes constitute the Mx receiver/driver bit fields. Note that the Mx receiver/driver bit fields perform a different function when the Mx pin is a status pin (see the Status Functionality section).

Table 29. Mx Receiver/Driver Bit Field Codes for Mx Control Pins

| Code | Boolean | Description |
|------|---------|--|
| 00 | AND | Logical AND the associated Mx control pin with the other Mx control pins assigned to the same control function. |
| 01 | NOT AND | Invert the logical state of the associated Mx control pin and AND it with the other Mx control pins assigned to the same control function. |
| 10 | OR | Logical OR the associated Mx control pin with the other Mx control pins assigned to the same control function. |
| 11 | NOT OR | Invert the logical state of the associated Mx control pin and OR it with the other Mx control pins assigned to the same control function. |

The Boolean functionality of aggregated Mx control pins follows a hierarchy whereby logical OR operations occur before logical AND operations. The OR and NOT OR operations are collectively grouped into a single result. A logical AND is then performed using that result and the remaining AND and NOT AND operations.

Consider a case where M0, M2, M3, and M6 are all assigned to the input/output update control function; that is, Bits[D6:D0] in Register 0x102 through Register 0x108 = 0x01 (see Table 30). In

addition, M0 is assigned for AND operation, M2 for NOT OR operation, M3 for NOT AND operation, and M6 for OR operation (that is, the 2-bit codes in Register 0x100 and Register 0x101 according to Table 29). With these settings, the input/output update function behaves according to the following Boolean equation:

$$\text{Input/output update} = (!M2 \parallel M6) \&\& M0 \&\& !M3$$

where:

! is logical NOT. Therefore, an input/output update occurs when M0 is Logic 1 and M3 is Logic 0, and either M2 is Logic 0 or M6 is Logic 1.

&& is logical AND.

|| is logical OR.

When an Mx control pin acts on a control function individually (rather than as part of a group, per the previous example), the Boolean functionality of the codes in Table 29 reduces to two possibilities. Namely, Code 00 and Code 10 specify a Boolean true (the Mx pin logic state applies to the corresponding control function directly), whereas Code 01 and Code 11 specify a Boolean false (the Mx pin logic state applies to the corresponding control function with a logical inversion).

Regarding the source and destination proxy columns in Table 30, the &&, || and ! symbols denote the Boolean AND, OR, and NOT operations, respectively.

Table 30. Mx Pin Status and Control Codes

| Bits[D6:D0] (Hex) | Control Function | Destination Proxy | Status Function | Source Proxy (or Description) |
|-------------------|-------------------------|-------------------------|-------------------------------------|---|
| 0x00 | No operation (NOOP) | Not applicable | Logic 0, static | Not applicable |
| 0x01 | IO_UPDATE | Register 0x000F, Bit D0 | Logic 1, static | Not applicable |
| 0x02 | Device power down | Register 0x2000, Bit D0 | Digital core clock | Not applicable |
| 0x03 | Clear watchdog timer | Register 0x2005, Bit D7 | Watchdog timer timeout | Not applicable |
| 0x04 | Sync all | Register 0x2000, Bit D3 | SYSCLK calibration in progress | Register 0x3001, Bit D2 |
| 0x05 | Unassigned | Not applicable | SYSCLK lock detect | Register 0x3001, Bit D0 |
| 0x06 | Unassigned | Not applicable | SYSCLK stable | Register 0x3001, Bit D1 |
| 0x07 | Unassigned | Not applicable | Channel 0 and Channel 1 PLLs locked | Register 0x3001, Bit D4 && Bit D5 |
| 0x08 | Unassigned | Not applicable | PLL0 locked | Register 0x3001, Bit D4 |
| 0x09 | Unassigned | Not applicable | PLL1 locked | Register 0x3001, Bit D5 |
| 0x0A | Unassigned | Not applicable | EEPROM save in progress | Register 0x3000, Bit D0 |
| 0x0B | Unassigned | Not applicable | EEPROM load in progress | Register 0x3000, Bit D1 |
| 0x0C | Unassigned | Not applicable | EEPROM fault detected | Register 0x3000, Bit D2 Bit D3 |
| 0x0D | Unassigned | Not applicable | Temperature sensor limit alarm | Register 0x3002, Bit D0 |
| 0x0E | Unassigned | Not applicable | Unassigned | Not applicable |
| 0x0F | Unassigned | Not applicable | Unassigned | Not applicable |
| 0x10 | Clear all IRQ events | Register 0x2005, Bit D0 | Any IRQ event | The logical OR of all triggered IRQ events |
| 0x11 | Clear common IRQ events | Register 0x2005, Bit D1 | Common IRQ event | The logical OR of all triggered common IRQ events |
| 0x12 | Clear PLL0 IRQ events | Register 0x2005, Bit D2 | PLL0 IRQ event | The logical OR of all triggered PLL0 IRQ events |
| 0x13 | Clear PLL1 IRQ events | Register 0x2005, Bit D3 | PLL1 IRQ event | The logical OR of all triggered PLL1 IRQ events |
| 0x14 | Unassigned | Not applicable | REFA demodulator clock | Not applicable |

| Bits[D6:D0] (Hex) | Control Function | Destination Proxy | Status Function | Source Proxy (or Description) |
|----------------------|--------------------------|---|-----------------------------------|--|
| 0x15 | Unassigned | Not applicable | Unassigned | Not applicable |
| 0x16 | Unassigned | Not applicable | REFAA demodulator clock | Not applicable |
| 0x17 | Unassigned | Not applicable | Unassigned | Not applicable |
| 0x18 | Unassigned | Not applicable | REFB demodulator clock | Not applicable |
| 0x19 | Unassigned | Not applicable | Unassigned | Not applicable |
| 0x1A | Unassigned | Not applicable | REFBB demodulator clock | Not applicable |
| 0x1B | Unassigned | Not applicable | Unassigned | Not applicable |
| 0x1C | Unassigned | Not applicable | REFA reference (R) divider resync | Register 0x300D, Bit D3 |
| 0x1D | Unassigned | Not applicable | REFAA R divider resync | Register 0x300D, Bit D7 |
| 0x1E | Unassigned | Not applicable | REFB R divider resync | Register 0x300E, Bit D3 |
| 0x1F | Unassigned | Not applicable | REFBB R divider resync | Register 0x300E, Bit D7 |
| 0x20 | Fault REFA | Register 0x2003, Bit D0 | REFA faulted | Register 0x3005, Bit D3 |
| 0x21 | Fault REFAA | Register 0x2003, Bit D1 | REFAA faulted | Register 0x3006, Bit D3 |
| 0x22 | Fault REFB | Register 0x2003, Bit D2 | REFB faulted | Register 0x3007, Bit D3 |
| 0x23 | Fault REFBB | Register 0x2003, Bit D3 | REFBB faulted | Register 0x3008, Bit D3 |
| 0x24 | Unassigned | Not applicable | REFA valid | Register 0x3005, Bit D4 |
| 0x25 | Unassigned | Not applicable | REFAA valid | Register 0x3006, Bit D4 |
| 0x26 | Unassigned | Not applicable | REFB valid | Register 0x3007, Bit D4 |
| 0x27 | Unassigned | Not applicable | REFBB valid | Register 0x3008, Bit D4 |
| 0x28 | Timeout REFA validation | Register 0x2002, Bit D0 (validate REFA if faulted; otherwise, no action) | REFA active | This function represents a logical combination of several registers and bits |
| 0x29 | Timeout REFAA validation | Register 0x2002, Bit D1 (validate REFAA if faulted; otherwise, no action) | REFAA active | This function represents a logical combination of several registers and bits |
| 0x2A | Timeout REFB validation | Register 0x2002, Bit D2 (validate REFB if faulted; otherwise, no action) | REFB active | This function represents a logical combination of several registers and bits |
| 0x2B | Timeout REFBB validation | Register 0x2002, Bit D3 (validate REFBB if faulted; otherwise, no action) | REFBB active | This function represents a logical combination of several registers and bits |
| 0x2C | Unassigned | Not applicable | Not applicable | Not applicable |
| 0x2D | Unassigned | Not applicable | Not applicable | Not applicable |
| 0x2E | Unassigned | Not applicable | Feedback 0 active | Not applicable |
| 0x2F | Unassigned | Not applicable | Feedback 1 active | Not applicable |
| 0x30 | Not applicable | Not applicable | DPLL0 phase locked | Register 0x3100, Bit D1 |
| 0x31 | Not applicable | Not applicable | DPLL0 frequency locked | Register 0x3100, Bit D2 |
| 0x32 | Not applicable | Not applicable | APLL0 locked | Register 0x3100, Bit D3 |
| 0x33 | Unassigned | Not applicable | APLL0 calibration in progress | Register 0x3100, Bit D4 |
| 0x34 | Unassigned | Not applicable | DPLL0 active | Register 0x3009, Bit D5 Bit D4 Bit D3 Bit D2 Bit D1 Bit D0 |
| 0x35 | Unassigned | Not applicable | DPLL0 freerun | Register 0x3101, Bit D0 |
| 0x36 | Unassigned | Not applicable | DPLL0 holdover | Register 0x3101, Bit D1 |
| 0x37 | Unassigned | Not applicable | DPLL0 switching | Register 0x3101, Bit D2 |
| 0x38 | Unassigned | Not applicable | DPLL0 tuning word history status | Register 0x3102, Bit D0 |
| 0x39 | Unassigned | Not applicable | DPLL0 tuning word history updated | Register 0x 3010, Bit D2 |
| 0x3A | Unassigned | Not applicable | DPLL0 frequency clamped | Register 0x3102, Bit D1 |
| 0x3B | Unassigned | Not applicable | DPLL0 phase slew limited | Register 0x3102, Bit D2 |

| Bits[D6:D0] (Hex) | Control Function | Destination Proxy | Status Function | Source Proxy (or Description) |
|----------------------|---|-------------------------|---------------------------------------|---|
| 0x3C | Unassigned | Not applicable | PLL0 distribution synchronized | Register 0x3013, Bit D4 |
| 0x3D | Unassigned | Not applicable | Unassigned | Not applicable |
| 0x3E | Unassigned | Not applicable | DPLL0 phase step detected | Register 0x3010, Bit D0 |
| 0x3F | Unassigned | Not applicable | DPLL0 fast acquisition active | Register 0x3102, Bit D4 |
| 0x40 | PLL0 power-down | Register 0x2100, Bit D0 | DPLL0 fast acquisition complete | Register 0x3102, Bit D5 |
| 0x41 | DPLL0 user freerun | Register 0x2105, Bit D0 | DPLL0 feedback divider resync | Register 0x3012, Bit D4 |
| 0x42 | DPLL0 user holdover | Register 0x2105, Bit D1 | PLL0 distribution phase slew enable | Indicates when any one of the PLL0 distribution phase slew limiters is actively limiting |
| 0x43 | DPLL0 clear tuning word history | Register 0x2107, Bit D1 | PLL0 distribution configuration error | Indicates when any one of the PLL0 distribution channel dividers encountered a phase offset error |
| 0x44 | Synchronize PLL0 distribution dividers | Register 0x2101, Bit D3 | Unassigned | Not applicable |
| 0x45 | DPLL0 translation profile select, Bit 0 | Register 0x2105, Bit D4 | Unassigned | Not applicable |
| 0x46 | DPLL0 translation profile select, Bit 1 | Register 0x2105, Bit D5 | Unassigned | Not applicable |
| 0x47 | DPLL0 translation profile select, Bit 2 | Register 0x2105, Bit D6 | Unassigned | Not applicable |
| 0x48 | Unassigned | Not applicable | Unassigned | Not applicable |
| 0x49 | Unassigned | Not applicable | Unassigned | Not applicable |
| 0x4A | Unassigned | Not applicable | Unassigned | Not applicable |
| 0x4B | Unassigned | Not applicable | Unassigned | Not applicable |
| 0x4C | Unassigned | Not applicable | Unassigned | Not applicable |
| 0x4D | Unassigned | Not applicable | Unassigned | Not applicable |
| 0x4E | Unassigned | Not applicable | Unassigned | Not applicable |
| 0x4F | Unassigned | Not applicable | Unassigned | Not applicable |
| 0x50 | Mute OUT0A | Register 0x2102, Bit D2 | DPLL1 phase locked | Register 0x3200, Bit D1 |
| 0x51 | Mute OUT0AA | Register 0x2102, Bit D3 | DPLL1 frequency locked | Register 0x3200, Bit D2 |
| 0x52 | Reset OUT0A/ OUT0AA driver | Register 0x2102, Bit D5 | APLL1 locked | Register 0x3200, Bit D3 |
| 0x53 | Mute OUT0B | Register 0x2103, Bit D2 | APLL1 calibration in progress | Register 0x3200, Bit D4 |
| 0x54 | Mute OUT0BB | Register 0x2103, Bit D3 | DPLL1 active | Register 0x300A, Bit D5 Bit D4 Bit D3 Bit D2 Bit D1 Bit D0 |
| 0x55 | Reset OUT0B/ OUT0BB driver | Register 0x2103, Bit D5 | DPLL1 freerun | Register 0x3201, Bit D0 |
| 0x56 | Mute OUT0C | Register 0x2104, Bit D2 | DPLL1 holdover | Register 0x3201, Bit D1 |
| 0x57 | Mute OUT0CC | Register 0x2104, Bit D3 | DPLL1 switching | Register 0x3201, Bit D2 |
| 0x58 | Reset OUT0C/ OUT0CC driver | Register 0x2104, Bit D5 | DPLL1 tuning word history status | Register 0x3202, Bit D0 |
| 0x59 | Mute OUT0xP/ OUT0xN | Register 0x2101, Bit D1 | DPLL1 tuning word history updated | Register 0x3015, Bit D2 |
| 0x5A | Reset OUT0xP/ OUT0xN drivers | Register 0x2101, Bit D2 | DPLL1 frequency clamped | Register 0x3202, Bit D1 |
| 0x5B | Channel 0 N-shot request | Register 0x2101, Bit D0 | DPLL1 phase slew limited | Register 0x3202, Bit D2 |
| 0x5C | Unassigned | Not applicable | PLL1 distribution synchronized | Register 0x3018, Bit D4 |
| 0x5D | Unassigned | Not applicable | Unassigned | Not applicable |

| Bits[D6:D0] (Hex) | Control Function | Destination Proxy | Status Function | Source Proxy (or Description) |
|----------------------|---|-------------------------|---|---|
| 0x5E | Unassigned | Not applicable | DPLL1 phase step detected | Register 0x3015, Bit D0 |
| 0x5F | Unassigned | Not applicable | DPLL1 fast acquisition active | Register 0x3202, Bit D4 |
| 0x60 | PLL1 power-down | Register 0x2200, Bit D0 | DPLL1 fast acquisition complete | Register 0x3202, Bit D5 |
| 0x61 | DPLL1 force freerun | Register 0x2205, Bit D0 | DPLL1 feedback divider resync | Register 0x3017, Bit D4 |
| 0x62 | DPLL1 force holdover | Register 0x2205, Bit D1 | PLL1 distribution phase slew enable OR'ed | Indicates when any one of the PLL1 distribution phase slew limiters is actively limiting |
| 0x63 | DPLL1 clear tuning word history | Register 0x2207, Bit D1 | PLL1 distribution phase control error OR'ed | Indicates when any one of the PLL1 distribution channel dividers encountered a phase offset error |
| 0x64 | Synchronize PLL1 distribution dividers | Register 0x2201, Bit D3 | Unassigned | Not applicable |
| 0x65 | DPLL1 translation profile select, Bit 0 | Register 0x2205, Bit D4 | Unassigned | Not applicable |
| 0x66 | DPLL1 translation profile select, Bit 1 | Register 0x2205, Bit D5 | Unassigned | Not applicable |
| 0x67 | DPLL1 translation profile select, Bit 2 | Register 0x2205, Bit D6 | Unassigned | Not applicable |
| 0x68 | Unassigned | Not applicable | Unassigned | Not applicable |
| 0x69 | Unassigned | Not applicable | Unassigned | Not applicable |
| 0x6A | Unassigned | Not applicable | Unassigned | Not applicable |
| 0x6B | Unassigned | Not applicable | Unassigned | Not applicable |
| 0x6C | Unassigned | Not applicable | Unassigned | Not applicable |
| 0x6D | Unassigned | Not applicable | Unassigned | Not applicable |
| 0x6E | Unassigned | Not applicable | Unassigned | Not applicable |
| 0x6F | Unassigned | Not applicable | Unassigned | Not applicable |
| 0x70 | Mute OUT1A | Register 0x2202, Bit D2 | Not applicable | Not applicable |
| 0x71 | Mute OUT1AA | Register 0x2202, Bit D3 | Not applicable | Not applicable |
| 0x72 | Reset OUT1A/OUT1AA driver | Register 0x2202, Bit D5 | Not applicable | |
| 0x73 | Mute OUT1B | Register 0x2203, Bit D2 | Not applicable | |
| 0x74 | Mute OUT1BB | Register 0x2203, Bit D3 | Not applicable | |
| 0x75 | Reset OUT1B/OUT1BB driver | Register 0x2203, Bit D5 | Not applicable | |
| 0x76 | Mute OUT1xP/OUT1xN drivers | Register 0x2201, Bit D1 | Unassigned | Not applicable |
| 0x77 | Reset OUT1xP/OUT1xN drivers | Register 0x2201, Bit D2 | Unassigned | Not applicable |
| 0x78 | Channel 1 N-shot request | Register 0x2201, Bit D0 | Timestamp 0 event detected | Register 0x300E, Bit D2 |
| 0x79 | Unassigned | Not applicable | Timestamp 1 event detected | Register 0x300E, Bit D3 |
| 0x7A | Unassigned | Not applicable | Skew measurement detected | Register 0x300E, Bit D4 |
| 0x7B | Unassigned | Not applicable | Unassigned | Not applicable |
| 0x7C | Unassigned | Not applicable | Unassigned | Not applicable |
| 0x7D | Unassigned | Not applicable | Unassigned | Not applicable |
| 0x7E | Unassigned | Not applicable | Unassigned | Not applicable |
| 0x7F | Unassigned | Not applicable | Unassigned | Not applicable |

INTERRUPT REQUEST (IRQ)

The AD9542 monitors certain internal device events potentially allowing them to trigger an IRQ event. Three groups of registers (see Figure 47) control the IRQ functionality within the AD9542:

- IRQ monitor registers (Register 0x300B through Register 0x3019)
- IRQ mask registers (Register 0x10C through Register 0x11A)
- IRQ clear registers (Register 0x2006 through Register 0x2014)

The IRQ logic can indicate an IRQ event status result for any specific device event(s) via the logical OR of the status of all the IRQ monitor bits. In addition, the IRQ logic offers IRQ event status results for particular groups of specific IRQ events, namely, the PLL0 IRQs, PLL1 IRQs, and common IRQs (see Figure 47).

The PLL0 IRQ group includes all device events associated with DPLL0 and APLL0. The PLL1 IRQ group includes all device events associated with DPLL1 and APLL1. The common IRQ group includes events associated with the system clock, the watchdog timer, and the EEPROM.

IRQ MONITOR

The IRQ monitor registers (in the general status section of the register map) maintain a record of specific IRQ events. The occurrence of a specific device event results in the setting and latching of the corresponding bit in the IRQ monitor. The output of the IRQ monitor provides the mechanism for generating IRQ event status results (see the PLL0 IRQ, PLL1 IRQ, common IRQ, or any IRQ signal shown in Figure 47).

IRQ MASK

The IRQ mask registers (in the Mx pin status and control section of the register map) comprise a bit for bit correspondence with the specific IRQ event bits within the IRQ monitor. Writing a Logic 1 to a mask bit enables (unmasks) the corresponding specific device event to the IRQ monitor. A Logic 0 (default) disables (masks) the corresponding specific device event to the IRQ monitor. Therefore, a specific IRQ event is the result of a logical AND of a specific device event and its associated IRQ, mask bit.

The presence of the IRQ mask allows the user to select certain device events for generating an IRQ event, while ignoring (masking) all other specific device events from contributing to an IRQ event status result (PLL0 IRQ, PLL1 IRQ, common IRQ

or any IRQ signal in Figure 47). Note that the default state of the IRQ mask register bits is Logic 0; therefore, the device is not capable of generating an IRQ event status result until the user populates the IRQ mask with a Logic 1 to unmask the desired specific IRQ events. Writing a Logic 1 to an IRQ mask bit may result in immediate indication of an IRQ status event result if the corresponding specific device event is already asserted (that is, the device previously registered the corresponding device event).

IRQ CLEAR

The IRQ clear registers (in the operational controls section of the register map) comprise a bit for bit correspondence with the IRQ monitor. Writing a Logic 1 to an IRQ clear bit forces the corresponding IRQ monitor bit to Logic 0, thereby clearing that specific IRQ event. Note that the IRQ clear registers are autoclearing; therefore, after writing a Logic 1 to an IRQ clear bit, the device automatically restores the IRQ clear bit to Logic 0. The IRQ event status results remain asserted until the user clears all of the bits in the IRQ monitor responsible for the IRQ status result (that is, the entire group of status bits associated with PLL0 IRQ, PLL1 IRQ, common IRQ, or any IRQ signal shown in Figure 47).

Although it is not recommended, in certain applications, it may be desirable to clear an entire IRQ group all at one time. Register 0x2005 provides four bits for clearing IRQ groups. Bit D0 clears all IRQ monitor bits. Bit D1 clears the common IRQ bits. Bit D2 clears the PLL0 IRQ bits. Bit D3 clears the PLL1 IRQ bits.

Alternately, the user can program any of the multifunction pins as an input for clearing an IRQ group, which allows clearing an IRQ group with an external logic signal rather than by writing to Register 0x2005 (see Figure 47).

The recommendation for clearing IRQ status events is to first service the specific IRQ event (as needed) and then clear the specific IRQ for that particular IRQ event. Clearing IRQ groups via Register 0x2005 or via an Mx pin requires great care. Clearing an IRQ group all at one time may result in the unintentional clearing of one or more asserted IRQ monitor bits. Clearing asserted IRQ monitor bits eliminates the record of the associated device events, subsequently erasing any history of those events having occurred.

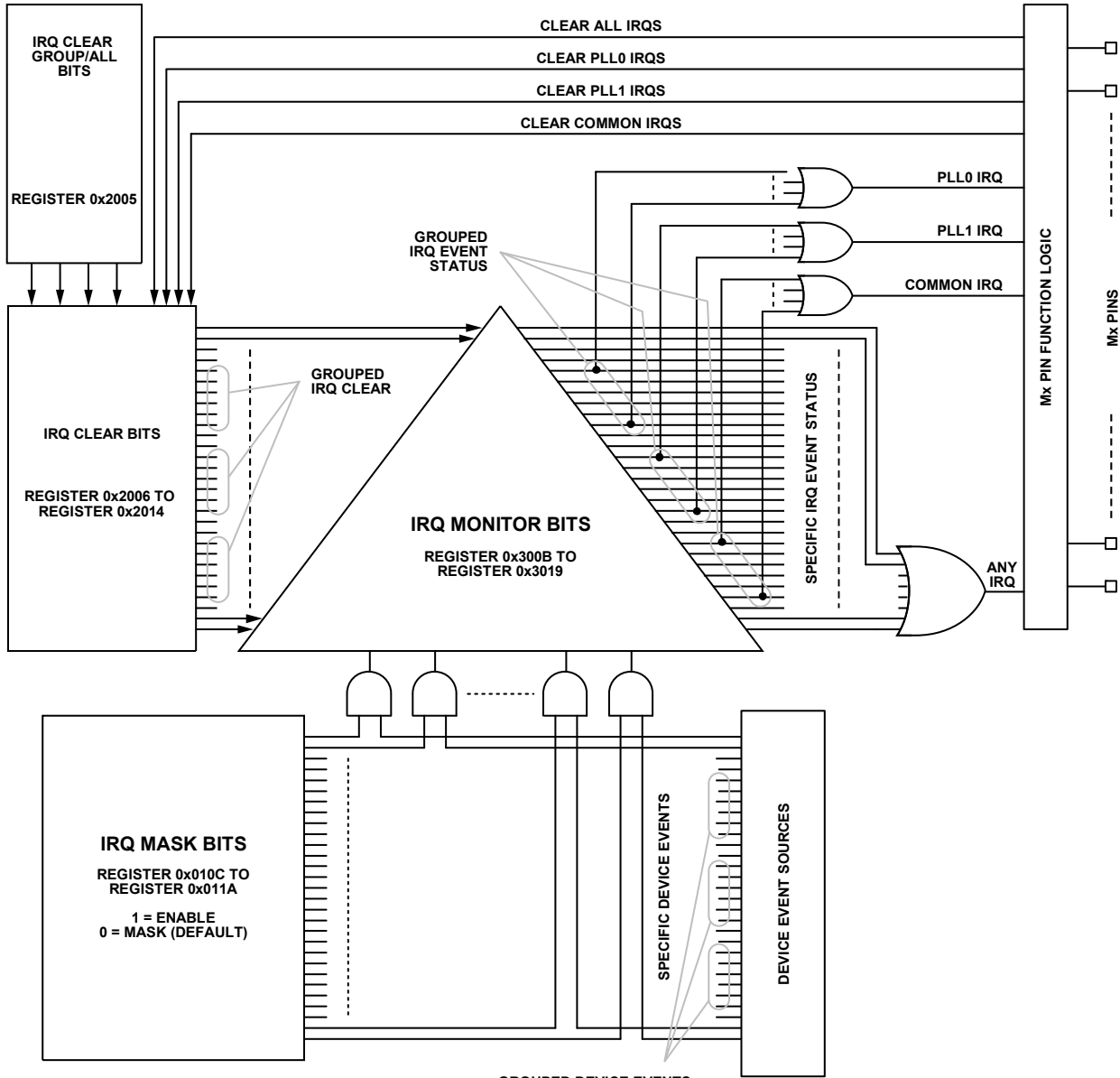


Figure 47. IRQ System Diagram

15926-005

WATCHDOG TIMER

The watchdog timer is a general-purpose programmable timer capable of triggering a specific IRQ event (see Figure 48). The timer relies on the system clock, however; therefore, the system clock must be present and locked for the watchdog timer to be functional. The bit fields associated with the watchdog timer reside in the Mx pin status and control function section of the register map.

The user sets the period of the watchdog timer by programming the watchdog timer (ms) bit field with a 16-bit timeout value. A nonzero value sets the timeout period in units of milliseconds, providing a range of 1 ms to 65.535 sec, whereas a zero value (0x0000, the default value) disables the timer. The relative accuracy of the timer is approximately 0.1% with an uncertainty of 0.5 ms. Note that whenever the user writes a 16-bit timeout value to the watchdog timer, it automatically clears the timer, ensuring a correct timeout period (per the new value) starting from the moment of the bit field update.

The watchdog timer (ms) bit field relates to the timeout period as follows:

$$\text{Watchdog Timer (ms)} = \text{Timeout Period} \times 10^3$$

To determine the value of the watchdog timer (ms) bit field necessary for a timeout period of 10 sec,

$$\begin{aligned} \text{Watchdog Timer (ms)} &= \text{Timeout Period} \times 103 \\ &= 10 \times 103 \\ &= 10,000 \\ &= 0x2710 \text{ (hexadecimal)} \end{aligned}$$

If enabled, the timer runs continuously and generates a timeout IRQ event when the timeout period expires. The user has access to the watchdog timer status via its associated IRQ monitor bit

or by assigning it directly to an Mx status pin. In the case of an Mx status pin, the timeout event of the watchdog timer is a pulse spanning 96 system clock periods (approximately 40 ns).

There are two ways to reset the watchdog timer, thereby preventing it from indicating a timeout event. The first method is by writing a Logic 1 to the clear watchdog bit (an autoclearing bit) in the operational controls section of the register map.

Alternatively, the user can program any of the multifunction pins as a control pin to reset the watchdog timer, which allows the user to reset the timer by means of a hardware pin rather than using the serial port.

There are two typical cases for employing the watchdog timer. Both cases assume that the watchdog timer output appears at the output of an appropriately configured Mx status pin (the watchdog timer output for the following case descriptions). The first case is for an external device (for example, an FPGA or microcontroller) to monitor the watchdog timer output using it as a signal to carry out periodic housekeeping functions. The second case is to have the watchdog timer output connected to the external device, such that the assertion of the watchdog output resets the external device. In this way, under normal operation, the external device repeatedly resets the watchdog timer by either writing Logic 1 to the clear watchdog bit or by asserting an Mx control pin configured for clearing the watchdog. In this way, as long as the external device keeps resetting the watchdog timer before it times out, the watchdog timer does not generate an output signal. As such, the watchdog timer does not reset the external device. However, if the external device fails to reset the watchdog timer before its timeout period expires, the watchdog timer eventually times out, resetting the external device via the appropriately configured Mx status pin.

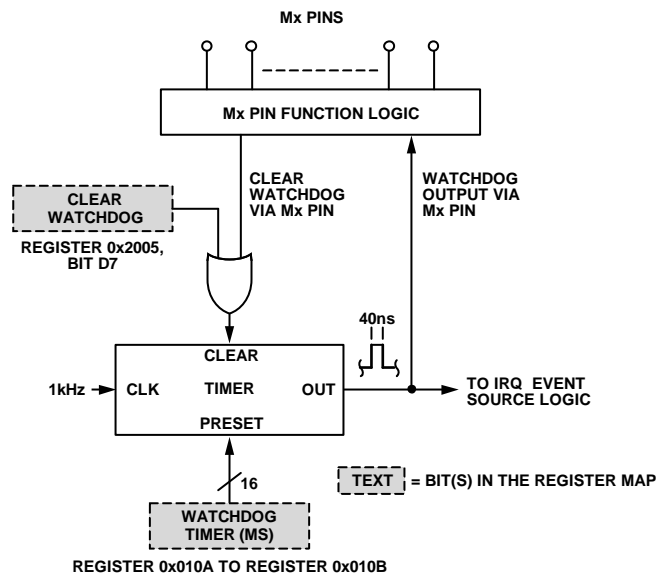


Figure 48. Watchdog Timer

LOCK DETECTORS

DPLL LOCK DETECTORS

DPLL Phase Lock Detector

Each DPLL channel (DPLL0 and DPLL1) contains an all digital phase lock detector. The user controls the threshold sensitivity and hysteresis of the phase detector via the source profiles.

The phase lock detector provides the user with two status bits in the status readback PLLx section the register map. The DPLLx phase lock bit latches to Logic 1 when the DPLL changes state from not phase locked to phase locked. The DPLLx phase unlock bit latches to Logic 1 when the DPLL changes state from phase locked to not phase locked. The DPLLx phase lock bits are located in Register 0x3100 and 0x3200, respectively. Because these bits can change dynamically, it is strongly recommended that the user set an IRQ for these bits. When using the IRQ function, it is possible for the IRQ status to indicate Logic 1 for an IRQ function that was just enabled if that condition is true at the time the IRQ is enabled. Therefore, the user must clear them via the IRQ map clear DPLL0 (Register 0x200B to Register 0x200F), IRQ map clear DPLL1 (Register 0x2010 to Register 0x2014), sections of the register map to obtain visibility of subsequent state transitions of the phase lock detector.

The phase lock detector behaves in a manner analogous to water in a tub (see Figure 49). The total capacity of the tub is 4096 units, with -2048 denoting empty, 0 denoting the 50% point, and $+2047$ denoting full. The tub also has a safeguard to prevent overflow. Furthermore, the tub has a low water mark at -1025 and a high water mark at $+1024$. To change the water level, the phase lock detector adds water with a fill bucket or removes water with a drain bucket. To specify the size of the fill and drain buckets, use the unsigned 8-bit Profile x phase lock fill rate and Profile x phase lock fill rate bit field (where x is a value from 0 through 7, corresponding to a particular source profile).

The water level in the tub is what the lock detector uses to determine the lock and unlock conditions. When the water level is below the low water mark (-1025), the lock detector indicates an unlock condition. Conversely, when the water level is above the high water mark ($+1024$), the lock detector indicates a lock condition. When the water level is between the marks, the lock detector holds its last condition. Figure 49 shows this concept with an overlay of an example of the instantaneous water level (vertical) vs. time (horizontal) and the resulting lock/unlock states.

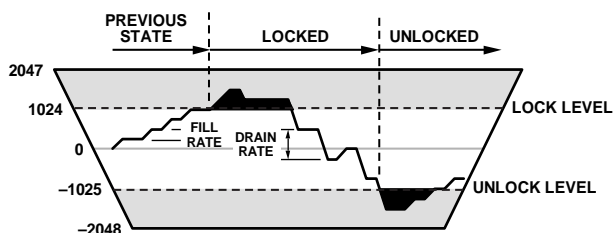


Figure 49. Lock Detector Diagram

During any given PFD phase error sample, the lock detector either adds water with the fill bucket or removes water with the drain bucket (one or the other, but not both). The decision of whether to add or remove water depends on the threshold level specified by the user in the 24-bit unsigned Profile x phase lock threshold bit field. The bit field value is the desired threshold in picoseconds. Thus, the phase lock threshold extends from 0 ps to $16.7 \mu\text{s}$ and represents the phase error at the output of the PFD. Though the programming range supports 0 ps as a lower limit, in practice, the minimum value must be greater than 50 ps.

The phase lock detector compares the absolute value of each phase error sample at the output of the PFD to the programmed phase threshold value. If the absolute value of the phase error sample is less than or equal to the programmed phase threshold value, the detector control logic adds one fill bucket into the tub. Otherwise, it removes one drain bucket from the tub. Note that it is the magnitude, relative to the phase threshold value, that determines whether to fill or drain the bucket, and not the polarity of the phase error sample.

An exception to the fill/drain process occurs when the phase slew limiter is active. When the phase slew limiter is actively in the limiting process, the lock detector blocks fill events, allowing only drain events to occur.

When more filling is taking place than draining, the water level in the tub eventually rises above the high water mark ($+1024$), which causes the lock detector to indicate lock. When more draining is taking place than filling, the water level in the tub eventually falls below the low water mark (-1024), which causes the lock detector to indicate unlock. The ability to specify the threshold level, fill rate, and drain rate enables the user to tailor the operation of the lock detector to the statistics of the timing jitter associated with the input reference signal. Note that, for debug purposes, the user can make the fill or drain rate zero to force the lock detector to indicate a lock or unlock state, respectively.

Note that whenever the AD9542 enters freerun or holdover mode, the DPLL phase lock detector indicates an unlocked state.

For more information on how to choose the appropriate phase lock threshold, fill rate, and drain rate values for a given application, refer to the [AN-1061 Application Note](#).

DPLL Frequency Lock Detector

The operation of the frequency lock detector is identical to that of the phase lock detector, with two exceptions:

- The fill or drain decision is based on the period deviation between the reference of the DPLL and the feedback signals, instead of the phase error at the output of the PFD.
- The frequency lock detector is unaffected by the state of the phase slew limiter.

The frequency lock detector provides the user with two status bits in the IRQ map DPLLx mask section of the register map. The DPLLx frequency lock bit (where x is 0 or 1) latches to Logic 1 when the DPLL changes state from not frequency locked to frequency locked. The DPLLx frequency unlock bit latches to Logic 1 when the DPLL changes state from frequency locked to not frequency locked. Because these are latched bits, the user must clear them via the IRQ map DPLLx clear section of the register map to obtain visibility of subsequent state transitions of the frequency lock detector.

The frequency lock detector uses the 24-bit unsigned Profile x frequency lock threshold bit field (where x is a value from 0 through 7, corresponding to a particular source profile), specified in units of picoseconds. Thus, the frequency threshold value extends from 0 ps to 16.7 μ s and represents the absolute value of the difference in period between the reference and feedback signals at the input to the DPLL.

$$\text{Profile } x \text{ Frequency Lock Threshold} = |1/f_{REF} - 1/f_{FB}|/10^{-12}$$

where:

f_{REF} is the frequency of the signal at the DPLL PFD reference input.

f_{FB} is the frequency of the signal at the DPLL PFD feedback input.

Consider a case where it is desirable to set the Profile x frequency lock threshold bit field to meet the frequency threshold when the signal from the reference TDC is 80 kHz and the signal from the feedback TDC is 79.32 kHz (or vice versa).

$$\begin{aligned} \text{Profile } x \text{ Frequency Lock Threshold} &= |1/f_{REF} - 1/f_{FB}|/10^{-12} \\ &= |1/80,000 - 1/79,320|/10^{-12} \\ &= 170,161 \text{ (nearest integer)} \\ &= 0x0298B1 \text{ (hexadecimal)} \end{aligned}$$

For more information on how to choose the appropriate frequency lock threshold, fill rate, and drain rate values for a given application, refer to the [AN-1061 Application Note](#).

PHASE STEP DETECTOR

PHASE STEP LIMIT

Although the AD9542 has the ability to switch between multiple reference inputs, some applications use only one input and handle reference switching externally (see Figure 50). Unfortunately, this arrangement forfeits the ability of the AD9542 to mitigate the output disturbance associated with a reference switchover, because the reference switchover is not under the control of the AD9542. However, the AD9542 offers a phase transient threshold detection feature to help identify when an external reference switchover occurs and to act accordingly.

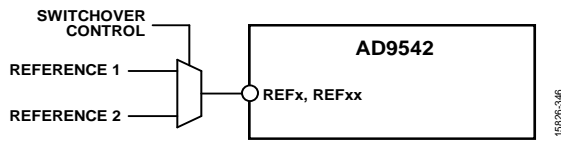


Figure 50. External Reference Switching

Phase transient threshold detection works by monitoring the output of the DPLL phase detector for phase transients, but in a manner that is somewhat jitter tolerant. Otherwise, the phase transient threshold detector is prone to false positives.

To activate the phase transient threshold detection, program the 32-bit unsigned Profile x phase step threshold bit field (where x is a value from 0 through 7, corresponding to a particular source profile). The default value is zero, which disables the phase transient threshold detector. A nonzero value denotes the desired phase step threshold in units of picoseconds per the following equation:

$$\text{Phase Step Threshold} = \text{Profile } x \text{ Phase Step Threshold} \times 10^{-12}$$

Note that the phase transient threshold detector is not active unless the DPLL indicates frequency locked status.

As an example, determine the value of the Profile x phase step threshold bit field necessary for a 12 ns limit. Solving the previous equation for the phase step limit yields

$$\begin{aligned} \text{Profile } x \text{ Phase Step Threshold} &= (12 \times 10^{-9})/10^{-12} \\ &= 12,000 \\ &= 0x00002EE0 \text{ (hexadecimal)} \end{aligned}$$

To reduce the likelihood of jitter induced threshold violations, choose a phase step threshold of at least two times the expected rms jitter (σ_{JITTER}) associated with the input reference signal.

$$\text{Profile } x \text{ Phase Step Threshold} \geq 2 \times \sigma_{\text{JITTER}}$$

As such, in the previous example with Profile x phase step threshold = 12,000, an input signal with 12 ns rms jitter is likely to produce false positives because the signal violates the previously described inequality. To reduce the likelihood of a false positive, the inequality indicates Profile x phase step threshold = 24,000 is a better choice. In fact, even with a value of 24,000, there is still a slight probability of a jitter sample exceeding $2 \times \sigma_{\text{JITTER}}$. As such, scaling σ_{JITTER} by four to six is an even better choice.

When a phase transient occurs that exceeds the prescribed value, one or both of the following two events occurs, depending on the state of the enable step detect reference fault bit in the operational control Channel 0 and Channel 1 (DPLL0 and DPLL1) sections of the register map:

- Logic 0: the DPLL initiates a new acquisition sequence.
- Logic 1: the reference monitor is reset.

When the enable step detect reference fault bit is Logic 0 (default), detection of a phase step causes only the first event to occur. By initializing a new DPLL acquisition sequence, the DPLL can take advantage of the fast acquisition feature, assuming it is active, which is especially helpful for very low loop bandwidth applications. In addition, a new acquisition manages the impact of the phase step by either building out the phase or slewing to the new phase in a hitless manner.

When the enable step detect reference fault bit is Logic 1, detection of a phase step causes both events to occur. Because exceeding the phase step threshold in this case implies an external switch to a new reference, resetting the reference monitor forces it to establish new reference statics.

The phase transient threshold detector provides the user with a live status bit in the status readback PLL x section of the register map, as well as a latched status bit in the IRQ map DPLL x read section of the register map. The DPLL x phase step detect bit (where x is 0 or 1) latches to Logic 1 on threshold violation of the phase transient threshold detector. Because this is a latched bit, the user must clear it via the IRQ map DPLL x clear section of the register map to obtain visibility of subsequent threshold violations detected by the phase step detector.

Mitigating Phase Step Limit False Positives

When enabled, the phase transient threshold detector operates continuously, as long as the associated reference is the active reference for the DPLL (DPLL0 or DPLL1, assuming the DPLL is frequency locked). As such, any phase disturbance at the input to the phase detector of the DPLL is subject to violating the threshold of the phase transient threshold detector. This violation includes a user induced phase adjustment via the DPLL x phase offset bit field or the Profile x phase skew bit field. To mitigate false triggering of the phase transient threshold detector (when enabled) due to intentional phase adjustments, the user can employ the phase slew rate limiter DPLL.

The following formula relates the maximum phase slew rate (MPSR) necessary to prevent inadvertent triggering of the phase transient threshold detector:

$$\text{MPSR} = (P + F)/7$$

where:

P is the phase transient threshold detector limit (in picoseconds).

F is the frequency (in Hz) at the input of the DPLL phase detector.

Note that this formula ignores other contributors to phase error, including jitter, frequency offset, and propagation delay variation.

If the user has a prior knowledge of the timing of an external event, such as the switching of the reference input clock source via an external mux, rather than using the phase transient step detector, a more robust solution is to invalidate the associated reference manually. To do so, force a reference fault condition via the appropriate operational controls bit field. Using this method imposes the least impact on the steady state operation of the device. The only steady state impact is that the validation timer of the associated reference must be set to a duration that is longer (with suitable margin) than the duration between the assertion of the force fault condition and the occurrence of the external event.

SKEW ADJUSTMENT

Skew adjustment allows the user to associate a fixed phase offset with a reference input, which, for example, is useful in applications with redundant GNSS/GPS reference sources. That is, a user

may have two or more GNSS/GPS sources that have identical frequency but may exhibit a fixed time offset due to a mismatch between antenna cable lengths.

To activate the skew adjustment feature, program the 24-bit signed Profile x phase skew bit field (where x is the profile number, Profile 0 to Profile 7). The default value is zero, which disables the skew adjustment feature. A nonzero value enables the skew adjustment feature and denotes the desired time skew in units of picoseconds per the following equation:

$$\text{Time Skew} = \text{Profile } x \text{ Phase Skew} \times 10^{-12}$$

As an example, determine the value of the Profile x phase skew bit field necessary for a time skew of -35 ns. Solving the previous equation for the Profile x phase skew yields

$$\begin{aligned} \text{Profile } x \text{ Phase Skew} &= (-35 \times 10^{-9}) / 10^{-12} \\ &= -35,000 \\ &= 0xFF7748 \text{ (hexadecimal)} \end{aligned}$$

EEPROM USAGE

OVERVIEW

The AD9542 supports an external, I²C-compatible, EEPROM with dedicated access. With some restrictions, the AD9542 also supports multidevice access to a single external EEPROM on a shared I²C common bus. The AD9542 has an on-chip I²C master to interface to the EEPROM through the Mx pins.

Because the default register settings of the AD9542 do not define a particular frequency translation, the user must factory program the EEPROM content before it can be downloaded to the register map (either automatically or manually). If desired, the user can store custom device configurations by manually forcing an upload to the EEPROM via the register map.

EEPROM CONTROLLER GENERAL OPERATION

EEPROM Controller

The EEPROM controller governs all aspects of communication with the EEPROM. Because the I²C interface uses a 100 kHz (normal mode) or 400 kHz (fast mode) communication link, the controller runs synchronous to an on-chip generated clock source suitable for use as the I²C serial clock. The on-chip oscillator enables asynchronously immediately on a request for activation of the controller. When the oscillator starts, it notifies the controller of its availability, and the controller activates. After the requested controller operation is complete, the controller disables the clock generator and returns to an idle state.

EEPROM Download

An EEPROM download transfers contents from the EEPROM to the AD9542 programming registers and invokes specific actions per the instructions stored in the EEPROM (see Table 31). Automatic downloading is the most common method for initiating an EEPROM download sequence, which initiates at power-up of the AD9542, provided Pin M3 is Logic 1 at power-up (see the Multifunction Pins at Reset/Power-Up section). Alternatively, instead of cycling power to the AD9542 to initiate an EEPROM download, the user can force the RESETB pin to Logic 0, force Pin M3 to Logic 1, and then return the RESETB pin to Logic 1 and remove the drive source from Pin M3.

The user can also request an EEPROM download on demand (that is, without resetting or cycling power to the AD9542) by writing a Logic 1 to the EEPROM load bit in the EEPROM section of the register map.

Note that the load from EEPROM bit does not require an input/output update. Writing a Logic 1 to this bit immediately triggers a download sequence.

The EEPROM controller sets the EEPROM load in progress bit (in the general status section of the register map) to Logic 1 while the download sequence is in progress as an indication to the user that the controller is busy.

EEPROM Upload

To store the AD9542 register contents in the EEPROM, the user must write a Logic 1 to the EEPROM save bit in the EEPROM section of the register map. The EEPROM save bit does not require an input/output update. Writing a Logic 1 to this bit immediately triggers an upload sequence.

The AD9542 has the equivalent of a write protect feature in that the user must write a Logic 1 to the EEPROM write enable bit (in the EEPROM section of the register map) prior to requesting an upload to the EEPROM. Attempting to upload to the EEPROM without first setting the EEPROM write enable bit results in a fault indication (that is, the AD9542 asserts the EEPROM fault bit in the general status section of the register map).

A prerequisite to an EEPROM upload is the existence of an upload sequence stored in the 15-byte EEPROM sequence section of the register map. That is, the user must store a series of upload instructions (see the EEPROM Instruction Set section) in the EEPROM sequence section of the register map prior to executing an EEPROM upload.

The EEPROM controller performs an upload sequence by reading the instructions stored in the EEPROM sequence section of the register map byte by byte and executing them in order. That is, the data stored in the EEPROM sequence section of the register map are instructions to the EEPROM controller on what to store in the EEPROM (including operational commands and AD9542 register data).

Note that the EEPROM controller sets the EEPROM save in progress bit (in the status readback section of the register map) to Logic 1 while the upload sequence is in progress as an indication to the user that the controller is busy.

Because the EEPROM sequence section of the register map is only 15 bytes, it typically cannot hold enough instructions to upload a complete set of AD9542 data to the EEPROM. Therefore, most upload sequences necessitate that the user upload a series of subsequences. For example, to accomplish a required upload sequence consisting of 20 bytes of instructions, perform the following procedure:

1. Write the first 14 instructions to the EEPROM sequence registers in the EEPROM section of the register map, with the 15th instruction being a pause instruction (see Table 31).
2. Initiate an EEPROM upload by writing Logic 1 to the EEPROM save bit. When the EEPROM controller reaches the pause instruction, it suspends the upload process and waits for another assertion of the EEPROM save bit.
3. While the controller pauses, write the remaining six bytes of the upload sequence into the EEPROM sequence registers in the EEPROM section of the register map, followed by an end of data instruction (see Table 31).
4. Initiate an EEPROM upload by writing Logic 1 to the EEPROM save bit. When the EEPROM controller reaches the end of data instruction, it terminates the upload process.

The previous procedure is an example of an upload sequence consisting of two subsequences. Most upload sequences require more than two subsequences; however, the procedure is the same. Specifically, partition a long sequence into several subsequences by using the pause instruction at the end of each subsequence and the end of data instruction at the end of the final subsequence.

EEPROM Checksum

When the EEPROM controller encounters an end of data instruction (see Table 31) during an upload sequence, it computes a 32-bit cyclic redundancy check (CRC) checksum and appends it to the stored data in the EEPROM. Similarly, when the EEPROM controller executes a download sequence, it computes a checksum on the fly. At the end of a download sequence, the EEPROM controller compares the newly computed checksum to the one stored in the EEPROM. If the two checksums do not match, the EEPROM controller asserts the EEPROM CRC error bit in the status readback section of the register map.

To minimize the possibility of downloading a corrupted EEPROM data set, the user can execute a checksum test by asserting the verify EEPROM CRC bit in the EEPROM section of the register map, which causes the EEPROM controller to execute a download sequence, but without actually transferring data to the AD9542 registers. The controller still computes an on the fly checksum, performs the checksum comparison, and asserts the EEPROM CRC error bit if the checksums do not match. Therefore, after the device deasserts the EEPROM load in progress bit, the user can check the EEPROM CRC fault bit to determine if the test passed (that is, EEPROM CRC error = 0).

However, even if the test fails, device operation is unaffected because there was no transfer of data to the AD9542 registers.

EEPROM Header

The EEPROM controller adds a header to stored data that carries information related to the AD9542, such as

- Vendor ID
- Chip type
- Product ID
- Chip revision

At the beginning of an EEPROM download sequence, the EEPROM controller compares the stored header values to the values in the corresponding registers of the AD9542. If the controller detects a mismatch, it asserts the EEPROM fault bit in the status readback section of the register map and terminates the download.

EEPROM INSTRUCTION SET

The EEPROM controller relies on a combination of instructions and data. An instruction consists of a single byte (eight bits). Some instructions require subsequent bytes of payload data. That is, some instructions are self contained operations, whereas others are directions on how to process subsequent payload data. A summary of the EEPROM controller instructions is shown in Table 31.

When the controller downloads the EEPROM contents to the AD9542 registers, it does so in a linear fashion, stepping through the instructions stored in the EEPROM. However, when the controller uploads to the EEPROM, the sequence is a nonlinear combination of various parts of the register map, as well as computed data that the controller calculates on the fly.

Table 31. EEPROM Controller Instruction Set Summary

| Instruction Code (Hexadecimal) | Response | Comments |
|--------------------------------|--------------------------------|--|
| 0x00 to 0x7F | Register transfer | Requires a 2-byte register address suffix |
| 0x80 | Input/output update | Assert input/output update during download |
| 0x81 to 0x8F | Not applicable | Undefined |
| 0x90 | Calibrate APLLs | Calibrate the system clock PLL, APLL0, and APLL1 during download |
| 0x91 | Calibrate the system clock PLL | Calibrate only the system clock PLL during download |
| 0x92 | Calibrate APLL0 | Calibrate only APLL0 during download |
| 0x93 | Calibrate APLL1 | Calibrate only APLL1 during download |
| 0x94 to 0x97 | Not applicable | Reserved/unused |
| 0x98 | Force freerun | Force DPLL0 and DPLL1 to freerun during download |
| 0x99 | Force DPLL0 freerun | Force only DPLL0 to freerun during download |
| 0x9A | Force DPLL1 freerun | Force only DPLL1 to freerun during download |
| 0x9B to 0x9F | Not applicable | Reserved/unused |
| 0xA0 | Synchronize outputs | Synchronize all distribution outputs during download |
| 0xA1 | Synchronize Channel 0 | Synchronize only Channel 0 distribution outputs during download |
| 0xA2 | Synchronize Channel 1 | Synchronize only Channel 1 distribution outputs during download |
| 0xA3 to 0xAF | Not applicable | Reserved/unused |
| 0xB0 | Clear condition | Apply Condition 0 and reset the condition map |
| 0xB1 to 0xBF | Set condition | Apply Condition 1 to Condition 15, respectively |
| 0xC0 to 0xFD | Not applicable | Undefined |
| 0xFE | Pause | Pause the EEPROM upload sequence |
| 0xFF | End of data | Marks the end of the instruction sequence |

Register Transfer Instructions (0x00 to 0x7F)

Instructions with a hexadecimal value from 0x00 through 0x7F indicate a register transfer operation. Register transfer instructions require a 2-byte suffix, which constitutes the starting address of the AD9542 register targeted for transfer (where the first byte to follow the data instruction is the most significant byte of the register address). When the EEPROM controller encounters a data instruction, it knows to interpret the next two bytes as the register map target address.

Note that the value of the register transfer instruction encodes the payload length (number of bytes). That is, the EEPROM controller knows how many register bytes to transfer to/from the indicated register by adding 1 to the instruction value. For example, Data Instruction 0x1A has a decimal value of 26; therefore, the controller knows to transfer 27 bytes to and from the target register (that is, one more than the value of the instruction).

Input/Output Update Instruction (0x80)

When the EEPROM controller encounters an input/output update instruction during an upload sequence, it stores the instruction in EEPROM. When encountered during a download sequence, however, the EEPROM controller initiates an input/output update event (equivalent to the user asserting the IO_UPDATE bit in the serial port section of the register map).

Device Action Instructions (0x90 to 0xAF)

When the EEPROM controller encounters a device action instruction during an upload sequence, it stores the instruction in EEPROM. When encountered during a download sequence, however, the EEPROM controller executes the specified action per Table 31.

Conditional Instructions (0xB0 to 0xBF)

The conditional instructions allow conditional execution of EEPROM instructions during a download sequence. During an upload sequence, however, they are stored as is and have no effect on the upload process.

Conditional processing makes use of four elements:

- The conditional instruction.
- The condition value.
- The condition ID.
- The condition map.

Conditional Instruction

When the EEPROM controller encounters a conditional instruction during an upload sequence, it stores the instruction in the EEPROM. When the EEPROM controller detects a conditional instruction during a download sequence, it affects the condition map as well as the outcome of conditional processing.

Condition Value

The condition value has a one to one correspondence to the conditional instruction. Specifically, the condition value is the value of the conditional instruction minus 0xB0. Therefore, condition values have a range of 0 to 15. The EEPROM controller uses condition values in conjunction with the condition map, while the user uses a condition value to populate the EEPROM load condition bit field of the register map with a condition ID.

Condition ID

The condition ID is the value stored in the 4-bit EEPROM load condition bit field in the EEPROM section of the register map. The EEPROM controller uses the condition ID in conjunction with the condition map to determine which instructions to execute or ignore during a download sequence.

Condition Map

The condition map is a table maintained by the EEPROM controller consisting of a list of condition values. When the EEPROM controller encounters a conditional instruction during a download sequence, it determines the corresponding condition value of the instruction (0 to 15). If the condition value is nonzero, the EEPROM controller places the condition value in the condition map. Conversely, if the condition value is zero, the controller clears the condition map and applies Condition 0. Condition 0 causes all subsequent instructions to execute unconditionally (until the controller encounters a new conditional instruction that causes conditional processing).

Conditional Processing

While executing a download sequence, the EEPROM controller executes or skips instructions depending on the condition ID and the contents of the condition map (except for the conditional and end of data instructions, which always execute unconditionally).

If the condition map is empty or the condition ID is zero, all instructions execute unconditionally during download. However, if the condition ID is nonzero and the condition map contains a condition value matching the condition ID, the EEPROM controller executes the subsequent instructions. Alternatively, if the condition ID is nonzero but the condition map does not contain a condition value matching the condition ID, the EEPROM controller skips instructions until it encounters a conditional instruction with a condition value of zero or a condition value matching the condition ID.

Note that the condition map allows multiple conditions to exist at any given moment. This multiconditional processing mechanism enables the user to have one download instruction sequence with many possible outcomes, depending on the value of the condition ID and the order in which the controller encounters conditional instructions. An example of the use of conditional processing is shown in Table 32.

Table 32. Example Conditional Processing Sequence

| Instruction | Operation |
|--------------|--|
| 0x00 to 0x7F | A sequence of register transfer instructions that execute unconditionally |
| 0xB1 | Apply Condition 1 |
| 0x00 to 0x7F | A sequence of register transfer instructions that execute only if the condition ID is 1 |
| 0xB2 | Apply Condition 2 |
| 0xB3 | Apply Condition 3 |
| 0x00 to 0x7F | A sequence of register transfer instructions that execute only if the condition ID is 1, 2, or 3 |
| 0x91 | Calibrate the system clock PLL |
| 0xB0 | Clear condition map |
| 0x80 | Input/output update |
| 0xFF | Terminate sequence |

Pause Instruction (0xFE)

The EEPROM controller only recognizes the pause instruction during an upload sequence. Upon encountering a pause instruction, the EEPROM controller enters an idle state, but preserves the current value of the EEPROM address pointer.

One use of the pause instruction is for saving multiple, yet distinct, values of the same AD9542 register, which is useful for sequencing power-up conditions.

The pause instruction is also useful for executing an upload sequence requiring more space than is available in the EEPROM sequence registers in the EEPROM section of the register map (see the EEPROM Upload section).

End of Data Instruction (0xFF)

When the EEPROM controller encounters an end of data instruction during an upload sequence, it stores the instruction in EEPROM along with the computed checksum, clears the EEPROM address pointer, and then enters an idle state. When encountered during a download sequence, however, the EEPROM controller clears the EEPROM address pointer, verifies the checksum, and then enters an idle state.

Note that during EEPROM downloads, condition instructions always execute unconditionally.

MULTIDEVICE SUPPORT

Multidevice support enables multiple AD9542 devices to share the contents of a single EEPROM. There are two levels of multidevice support. Level 1 supports a configuration where multiple AD9542 devices share a single EEPROM through a dedicated I²C bus. Level 2 supports a configuration where multiple AD9542 devices share a single EEPROM connected to a common I²C bus that includes other I²C master devices. Figure 51 and Figure 52 show the Level 1 and Level 2 configurations, respectively.

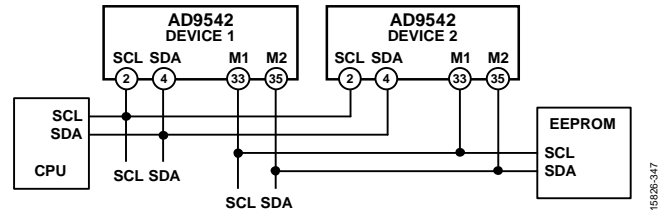


Figure 51. Level 1 Multidevice Configuration

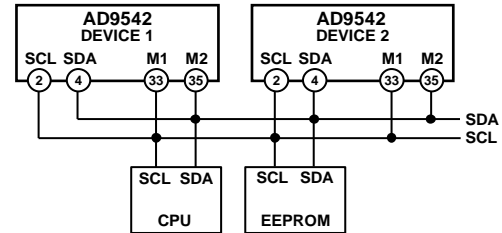


Figure 52. Level 2 Multidevice Configuration

Multidevice Bus Arbitration

The EEPROM controller implements bus arbitration by continuously monitoring the SDA and SCL bus signals for start and stop conditions. The controller can determine whether the bus is idle or busy. If the bus is busy, the EEPROM controller delays its pending I²C transfer until a stop condition indicates that the bus is available.

Bus arbitration is essential in cases where two I²C master devices simultaneously attempt an I²C transfer. For example, if one I²C master detects that SDA is Logic 0 when it is intended to be Logic 1, it assumes that another I²C master is active and immediately terminates its own attempt to transfer data. Similarly, if one I²C master detects that SCL is Logic 0 prior to entering a start state, it assumes that another I²C master is active and stalls its own attempt to drive the bus.

In either case, the prevailing I²C master completes its current transaction before releasing the bus. Because the postponed I²C master continuously monitors the bus for a stop condition, it attempts to seize the bus and carry out the postponed transaction on detection of such a stop condition.

The EEPROM controller includes an arbitration timer to optimize the bus arbitration process. Specifically, when the EEPROM controller postpones an I²C transfer as a result of detecting bus contention, it starts the arbitration timer. If the EEPROM controller fails to detect a stop condition within 255 SCL cycles, it attempts to force another transaction. If the bus is still busy, the EEPROM controller restarts the arbitration timer, and the process continues until the EEPROM controller eventually completes the pending transaction.

Multidevice Configuration Example

Consider two AD9542 devices (Device 1 and Device 2) that share a single EEPROM, and assume both devices have a common PLL0 configuration but differing PLL1 configurations.

A template for an EEPROM sequence that accomplishes this configuration is shown in Table 33. The sequence relies on conditional processing to differentiate between Device 1 and Device 2. Therefore, the user must program the condition ID of both devices prior to executing an EEPROM download. Specifically, the user must program the EEPROM load condition bit field of Device 1 with a condition ID of 1 and Device 2 with a condition ID of 2.

Table 33. Template for a Multidevice EEPROM Sequence

| Instructions | Comment |
|---------------------|--|
| 0x00 to 0x7F | A sequence of register transfer instructions associated with the PLL0 configuration common to both devices |
| 0xB1 | Apply Condition 1 |
| 0x00 to 0x7F | A sequence of register transfer instructions associated with the PLL1 configuration specific to Device 1 |
| 0xB0 | Clear the condition map |
| 0xB2 | Apply Condition 2 |
| 0x00 to 0x7F | A sequence of register transfer instructions associated with the PLL1 configuration specific to Device 2 |
| 0xB0 | Clear the condition map |
| 0x80 | Input/output update |
| 0xFF | End of sequence |

SERIAL CONTROL PORT

The AD9542 serial control port is a flexible, synchronous serial communications port that provides a convenient interface to many industry-standard microcontrollers and microprocessors. The AD9542 serial control port is compatible with most synchronous transfer formats, including I²C, Motorola SPI, and Intel SSR protocols. The serial control port allows read/write access to the AD9542 register map.

The AD9542 uses the Analog Devices unified SPI protocol (see the [Analog Devices Serial Control Interface Standard](#)). The unified SPI protocol guarantees that all new Analog Devices products using the unified protocol have consistent serial port characteristics. The SPI port configuration is programmable via Register 0x00.

Unified SPI differs from the SPI port found on older Analog Devices products, such as the [AD9557](#) and [AD9558](#), in the following ways:

- Unified SPI does not have byte counts. A transfer is terminated when the CSB pin goes high. The W1 and W0 bits in the traditional SPI become the A12 and A13 bits of the register address. This is similar to streaming mode in the traditional SPI.
- The address ascension bit (Register 0x00) controls whether register addresses are automatically incremented or decremented regardless of the LSB/MSB first setting. In traditional SPI, LSB first mode dictated auto-incrementing and MSB first mode dictated auto-decrementing of the register address.
- The first 16 register addresses of devices that adhere to the unified serial port have a consistent structure.

SPI/I²C PORT SELECTION

Although the AD9542 supports both the SPI and I²C serial port protocols, only one is active following power-up (as determined by the M4 multifunction pin during the start-up sequence). The only way to change the serial port protocol is to reset (or power cycle) the device. See Table 27 for the I²C address assignments.

SPI SERIAL PORT OPERATION

Pin Descriptions

The serial clock (SCLK) pin serves as the serial shift clock. This pin is an input. SCLK synchronizes serial control port read and write operations. The rising edge SCLK registers write data bits, and the falling edge registers read data bits. The SCLK pin supports a maximum clock rate of 50 MHz.

The SPI port supports both 3-wire (bidirectional) and 4-wire (unidirectional) hardware configurations and both MSB first and LSB first data formats. Both the hardware configuration and data format features are programmable. The 3-wire mode uses the serial data input/output (SDIO) pin for transferring data in both directions. The 4-wire mode uses the SDIO pin for transferring data to the AD9542, and the SDO pin for transferring data from the AD9542.

The chip select (CSB) pin is an active low control that gates read and write operations. Assertion (active low) of the CSB pin initiates a write or read operation to the AD9542 SPI port. The user can transfer any number of data bytes in a continuous stream. The register address is automatically incremented or decremented based on the setting of the address ascension bit (Register 0x00). The user must deassert the CSB pin following the last byte transferred, thereby ending the stream mode. This pin is internally connected to a 10 k Ω pull-up resistor. When CSB is high, the SDIO and SDO pins enter a high impedance state.

Implementation Specific Details

The [Analog Devices Serial Control Interface Standard](#) provides a detailed description of the unified SPI protocol and covers items such as timing, command format, and addressing. The unified SPI protocol defines the following device specific items:

- Analog Devices unified SPI protocol revision: 1.0
- Chip type: 0x5
- Product ID: 0x012
- Physical layer: 3-wire and 4-wire supported and 1.5 V, 1.8 V, and 2.5 V operation supported
- Optional single-byte instruction mode: not supported
- Data link: not used
- Control: not used

Communication Cycle—Instruction Plus Data

The unified SPI protocol consists of a two-part communication cycle. The first part is a 16-bit instruction word coincident with the first 16 SCLK rising edges. The second part is the payload, the bits of which are coincident with SCLK rising edges. The instruction word provides the AD9542 serial control port with information regarding the payload. The instruction word includes the R/W bit that indicates the direction of the payload transfer (that is, a read or write operation). The instruction word also indicates the starting register address of the first payload byte.

Write

When the instruction word indicates a write operation, the payload is written into the serial control port buffer of the AD9542. Data bits are registered on the rising edge of SCLK. Generally, it does not matter what data is written to blank registers; however, it is customary to use 0s. Note that the user must verify that all reserved registers within a specific range have a default value of 0x00; however, Analog Devices makes every effort to avoid having reserved registers with nonzero default values.

Most of the serial port registers are buffered; therefore, data written into buffered registers does not take effect immediately. To transfer buffered serial control port contents to the registers that actually control the device requires an additional operation, an IO_UPDATE operation, implemented in one of two ways. One is to write a Logic 1 to Register 0x0F, Bit 0 (this bit is an

autoclearing bit). The other is to use an external signal via an appropriately programmed multifunction pin. The user can change as many register bits as desired before executing an input/output update. The input/output update operation transfers the buffer register contents to their active register counterparts.

Read

If the instruction word indicates a read operation, the next $N \times 8$ SCLK cycles clock out the data starting from the address specified in the instruction word, where N is the number of data bytes to read. Read data appears on the appropriate data pin (SDIO or SDO) on the falling edge of SCLK. The user must latch the read data on the rising edge of SCLK. Note that the internal SPI control logic does not skip over blank registers during a readback operation.

A readback operation takes data from either the serial control port buffer registers or the active registers, as determined by Register 0x01, Bit 5.

SPI Instruction Word (16 Bits)

The MSB of the 16-bit instruction word is $\overline{R/W}$, which indicates whether the ensuing operation is read or write. The next 15 bits are the register address (A14 to A0), which indicates the starting register address of the read/write operation (see Table 35). Note that SPI controller ignores A14, treating it as Logic 0, because the AD9542 has no register addresses requiring more than a 14-bit address word.

Table 35. Serial Control Port, 16-Bit Instruction Word

| MSB | | | | | | | | | | | | | | | LSB |
|-----|-----|-----|-----|-----|-----|----|----|----|----|----|----|----|----|----|-----|
| I15 | I14 | I13 | I12 | I11 | I10 | I9 | I8 | I7 | I6 | I5 | I4 | I3 | I2 | I1 | I0 |
| R/W | A14 | A13 | A12 | A11 | A10 | A9 | A8 | A7 | A6 | A5 | A4 | A3 | A2 | A1 | A0 |

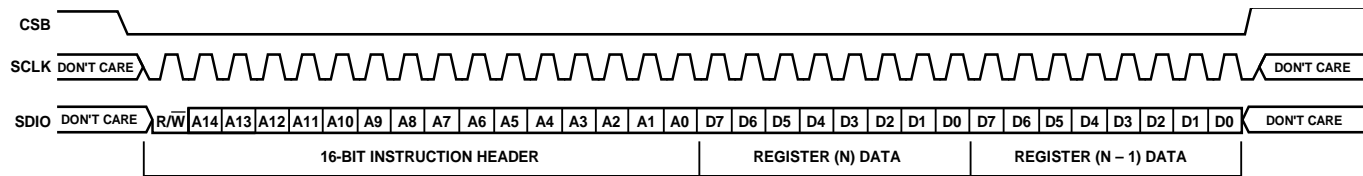


Figure 53. Serial Control Port Write—MSB First, Address Decrement, Two Bytes of Data

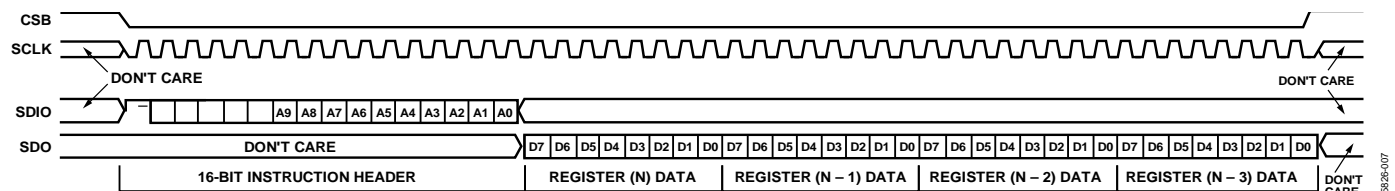


Figure 54. Serial Control Port Read—MSB First, Address Decrement, Four Bytes of Data

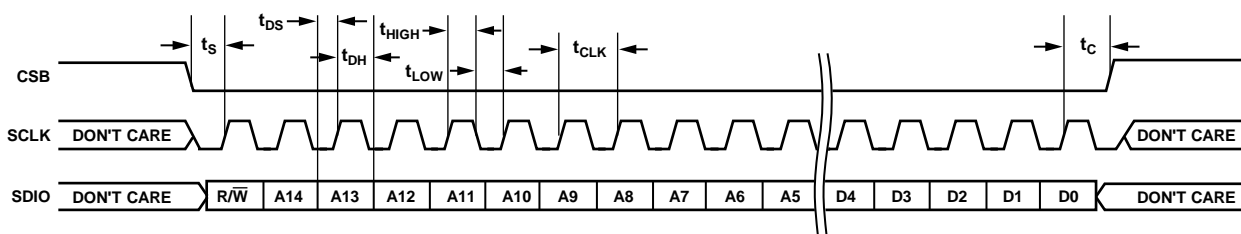


Figure 55. Timing Diagram for Serial Control Port Write—MSB First

SPI MSB-/LSB-First Transfers

The AD9542 instruction word and payload can be transferred MSB first or LSB first. The default for the AD9542 is MSB first. To invoke LSB first mode, write a Logic 1 to Register 0x00, Bit 6. Immediately after invoking LSB first mode, subsequent serial control port operations are LSB first.

Address Ascension

If the address ascension bit (Register 0x00, Bit 5) is Logic 0, serial control port register addressing decrements from the specified starting address toward Address 0x00. If the address ascension bit (Register 0x00, Bit 5) is Logic 1, serial control port register addressing increments from the starting address toward Address 0x3A3B. Reserved addresses are not skipped during multibyte input/output operations; therefore, write the default value to a reserved register and Logic 0s to unmapped registers. Note that it is more efficient to issue a new write command than to write the default value to more than two consecutive reserved (or unmapped) registers.

Table 34. Streaming Mode (No Addresses Skipped)

| Address Ascension | Stop Sequence |
|-------------------|-------------------|
| Increment | 0x0000 ... 0x3A3B |
| Decrement | 0x3A3B ... 0x0000 |

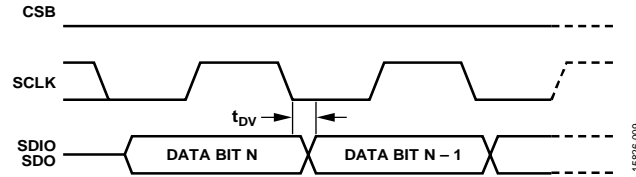


Figure 56. Timing Diagram for Serial Control Port Register Read—MSB First

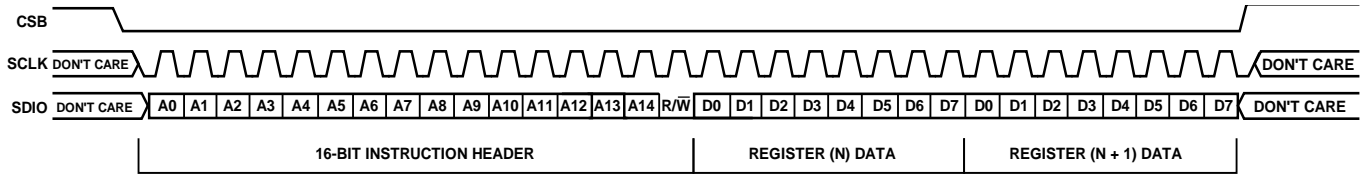


Figure 57. Serial Control Port Write—LSB First, Address Increment, Two Bytes of Data

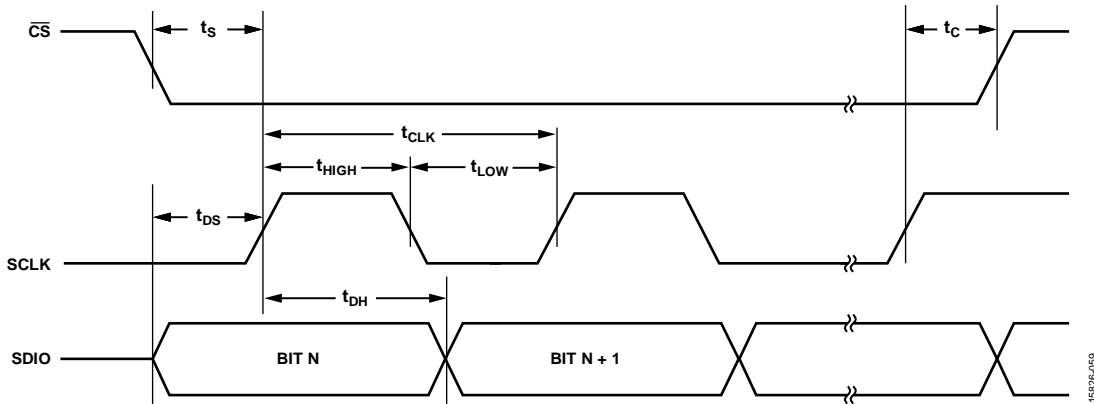


Figure 58. Serial Control Port Timing—Write

Table 36. Serial Control Port Timing

| Parameter | Description |
|------------|---|
| t_{DS} | Setup time between data and the rising edge of SCLK |
| t_{DH} | Hold time between data and the rising edge of SCLK |
| t_{CLK} | Period of the clock |
| t_s | Setup time between the CSB falling edge and the SCLK rising edge (start of the communication cycle) |
| t_c | Setup time between the SCLK rising edge and CSB rising edge (end of the communication cycle) |
| t_{HIGH} | Minimum period that SCLK is in a logic high state |
| t_{LOW} | Minimum period that SCLK is in a logic low state |
| t_{DV} | SCLK to valid SDIO (see Figure 56) |

I²C SERIAL PORT OPERATION

The I²C interface is popular because it requires only two pins and easily supports multiple devices on the same bus. Its main disadvantage is its maximum programming speed of 400 kbps. The AD9542 I²C port supports the 400 kHz fast mode as well as the 100 kHz standard mode.

To support 1.5 V, 1.8 V, and 2.5 V I²C operation, the AD9542 does not strictly adhere to every requirement in the original I²C specification. In particular, it does not support specifications such as slew rate limiting and glitch filtering. Therefore, the AD9542 is I²C compatible, but not necessarily fully I²C compliant.

The AD9542 I²C port consists of a serial data line (SDA) and a serial clock line (SCL). In an I²C bus system, the AD9542 connects to the serial bus (data bus SDA and clock bus SCL) as a slave device; that is, the AD9542 does not generate an I²C clock. The AD9542 uses direct 16-bit memory addressing rather than 8-bit memory addressing, which is more common.

The AD9542 allows up to four unique slave devices to occupy the I²C bus via a 7-bit slave address transmitted as part of an I²C packet. Only the device with a matching slave address responds to subsequent I²C commands. Table 37 lists the supported device slave addresses.

I²C Bus Characteristics

A summary of the various I²C abbreviations appears in Table 37.

Table 37. I²C Bus Abbreviation Definitions

| Abbreviation | Definition |
|--------------|----------------|
| S | Start |
| Sr | Repeated start |
| P | Stop |
| A | Acknowledge |
| \bar{A} | Nonacknowledge |
| \bar{W} | Write |
| R | Read |

An example of valid data transfer appears in Figure 59. One clock pulse is required for each data bit transferred. The data on the SDA line must be stable during the high period of the clock. The high or low state of the data line can change only when the clock signal on the SCL line is low.

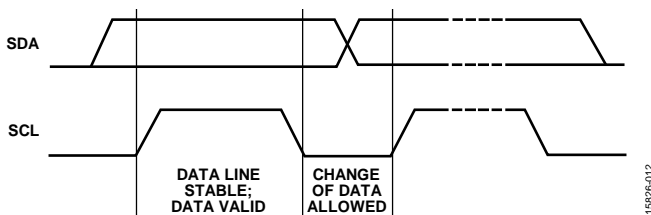


Figure 59. Valid Bit Transfer

Start and stop functionality appears in Figure 60. The start condition is a high to low transition on the SDA line while SCL is high. The master always generates the start condition to initialize a data transfer. The stop condition is a low to high transition on the SDA line while SCL is high. The master always

generates the stop condition to terminate a data transfer. The SDA line must always transfer eight bits (one byte). Each byte must be followed by an acknowledge bit; bytes are sent MSB first.

The acknowledge bit (A) is the ninth bit attached to any 8-bit data byte. An acknowledge bit is always generated by the receiver to inform the transmitter that the byte has been received. Acknowledgement consists of pulling the SDA line low during the ninth clock pulse after each 8-bit data byte.

The nonacknowledge bit (\bar{A}) is the ninth bit attached to any 8-bit data byte. A nonacknowledge bit is always generated by the receiver to inform the transmitter that the byte has not been received. Nonacknowledgment consists of leaving the SDA line high during the ninth clock pulse after each 8-bit data byte. After issuing a nonacknowledge bit, the AD9542 I²C state machine goes into an idle state.

Data Transfer Process

The master initiates data transfer by asserting a start condition, which indicates that a data stream follows. All I²C slave devices connected to the serial bus respond to the start condition.

The master then sends an 8-bit address byte over the SDA line, consisting of a 7-bit slave address (MSB first) plus a R/W bit. This bit determines the direction of the data transfer, that is, whether data is written to or read from the slave device (Logic 0 indicates write, and Logic 1 indicates read).

The peripheral whose address corresponds to the transmitted address responds by sending an acknowledge bit. All other devices on the bus remain idle while the selected device waits for data to be read from or written to it. If the R/W bit is Logic 0, the master (transmitter) writes to the slave device (receiver). If the R/W bit is Logic 1, the master (receiver) reads from the slave device (transmitter). The format for these commands appears in the Data Transfer Format section.

Data is then sent over the serial bus in the format of nine clock pulses, one data byte (eight bits) from either master (write mode) or slave (read mode), followed by an acknowledge bit from the receiving device. The protocol allows a data transfer to consist of any number of bytes (that is, the payload size is unrestricted). In write mode, the first two data bytes immediately after the slave address byte are the internal memory (control registers) address bytes (the higher address byte first). This addressing scheme gives a memory address of up to $2^{16} - 1 = 65,535$. The data bytes after these two memory address bytes are register data written to or read from the control registers. In read mode, the data bytes following the slave address byte consist of register data written to or read from the control registers.

When all the data bytes are read or written, stop conditions are established. In write mode, the master device (transmitter) asserts a stop condition to end data transfer during the clock pulse following the acknowledge bit for the last data byte from the slave device (receiver). In read mode, the master device (receiver) receives the last data byte from the slave device (transmitter) but does not pull SDA low during the ninth clock pulse (a

nonacknowledge bit). By receiving the nonacknowledge bit, the slave device knows that the data transfer is finished and enters idle mode. The master device then takes the data line low during the low period before the 10th clock pulse, and high during the 10th clock pulse to assert a stop condition.

A start condition can be used instead of a stop condition. Furthermore, a start or stop condition can occur at any time, and partially transferred bytes are discarded.

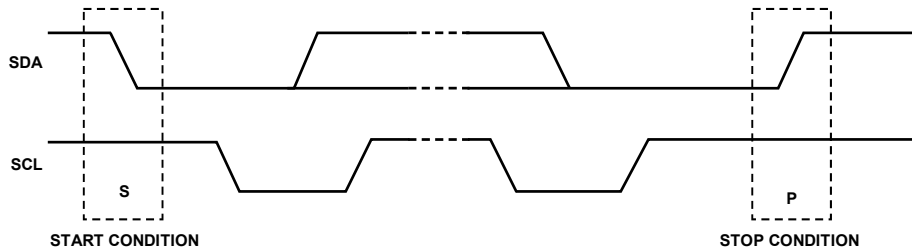


Figure 60. Start and Stop Conditions

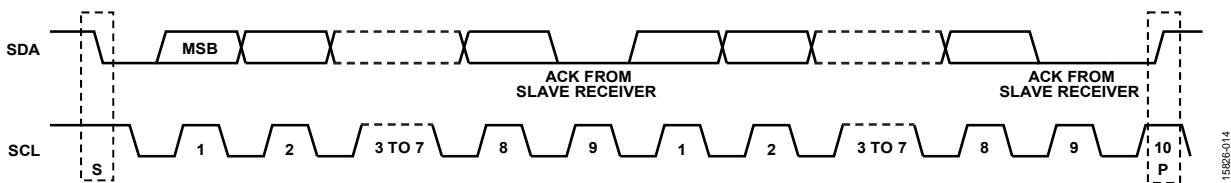


Figure 61. Acknowledge Bit

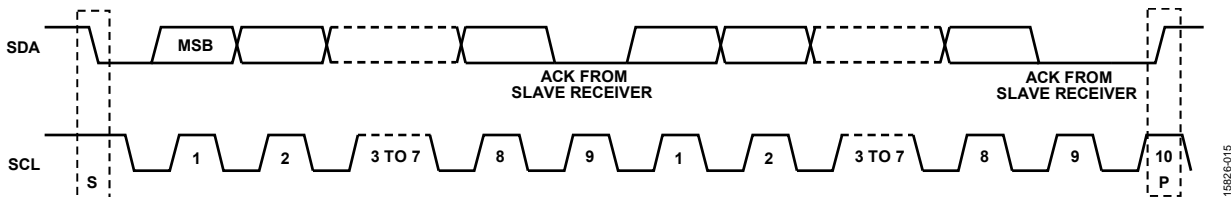


Figure 62. Data Transfer Process (Master Write Mode, 2-Byte Transfer)

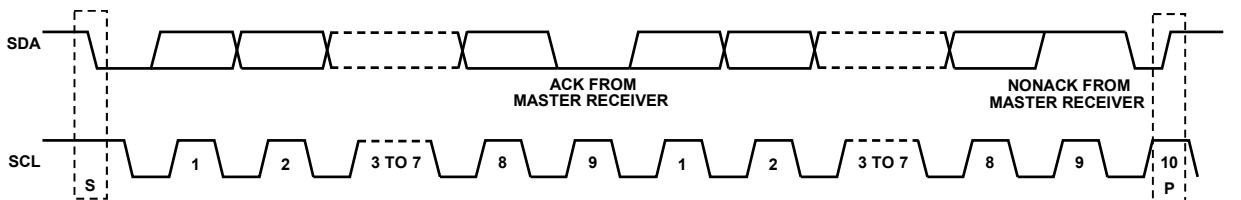


Figure 63. Data Transfer Process (Master Read Mode, 2-Byte Transfer), First Acknowledge From Slave

Data Transfer Format

The write byte format is used to write a register address to the RAM starting from the specified RAM address (see Table 38). The send byte format is used to set up the register address for

subsequent reads (see Table 39). The receive byte format is used to read the data byte(s) from RAM starting from the current address (see Table 40). The read byte format is the combined format of the send byte and the receive byte (see Table 41).

Table 38. Write Byte Format

| | | | | | | | | | | | | | | |
|---|---------------|-----------|---|-----------------------|---|----------------------|---|------------|---|------------|---|------------|---|---|
| S | Slave address | \bar{W} | A | RAM address high byte | A | RAM address low byte | A | RAM Data 0 | A | RAM Data 1 | A | RAM Data 2 | A | P |
|---|---------------|-----------|---|-----------------------|---|----------------------|---|------------|---|------------|---|------------|---|---|

Table 39. Send Byte Format

| | | | | | | | | |
|---|---------------|-----------|---|-----------------------|---|----------------------|---|---|
| S | Slave address | \bar{W} | A | RAM address high byte | A | RAM address low byte | A | P |
|---|---------------|-----------|---|-----------------------|---|----------------------|---|---|

Table 40. Receive Byte Format

| | | | | | | | | | | |
|---|---------------|---|---|------------|---|------------|---|------------|-----------|---|
| S | Slave address | R | A | RAM Data 0 | A | RAM Data 1 | A | RAM Data 2 | \bar{A} | P |
|---|---------------|---|---|------------|---|------------|---|------------|-----------|---|

Table 41. Read Byte Format

| | | | | | | | | | | | | | | | | | | |
|---|---------------|-----------|---|-----------------------|---|----------------------|---|----|---------------|---|---|------------|---|------------|---|------------|-----------|---|
| S | Slave address | \bar{W} | A | RAM address high byte | A | RAM address low byte | A | Sr | Slave address | R | A | RAM Data 0 | A | RAM Data 1 | A | RAM Data 2 | \bar{A} | P |
|---|---------------|-----------|---|-----------------------|---|----------------------|---|----|---------------|---|---|------------|---|------------|---|------------|-----------|---|

I²C Serial Port Timing

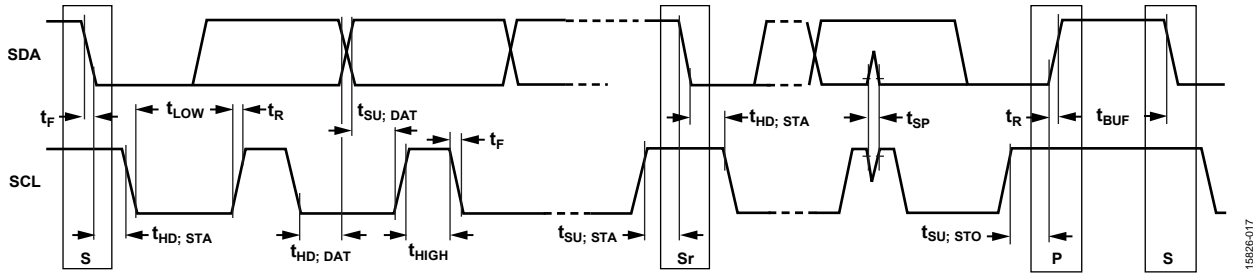
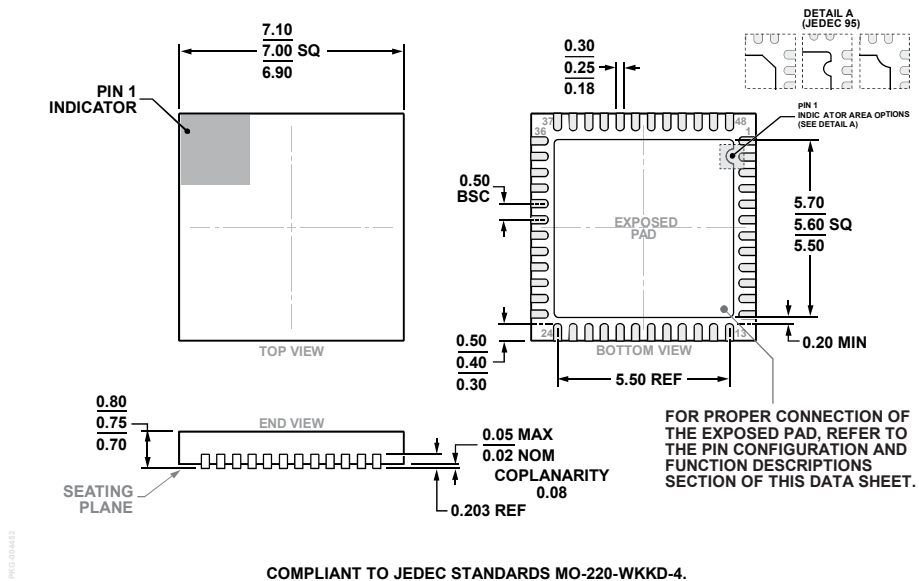


Figure 64. I²C Serial Port Timing

Table 42. I²C Timing Definitions

| Parameter | Description |
|----------------------|---|
| f _{SCL} | Serial clock |
| t _{BUF} | Bus free time between stop and start conditions |
| t _{HD; STA} | Repeated hold time start condition |
| t _{SU; STA} | Repeated start condition setup time |
| t _{SU; STO} | Stop condition setup time |
| t _{HD; DAT} | Data hold time |
| t _{SU; DAT} | Data setup time |
| t _{LOW} | SCL clock low period |
| t _{HIGH} | SCL clock high period |
| t _R | Minimum/maximum receive SCL and SDA rise time |
| t _F | Minimum/maximum receive SCL and SDA fall time |
| t _{SP} | Pulse width of voltage spikes that must be suppressed by the input filter |

OUTLINE DIMENSIONS



COMPLIANT TO JEDEC STANDARDS MO-220-WKKD-4.
 Figure 65. 48-Lead Lead Frame Chip Scale Package [LFCSP]
 7 mm × 7 mm Body and 0.75 mm Package Height
 (CP-48-13)
 Dimensions shown in millimeters

ORDERING GUIDE

| Model ¹ | Temperature Range | Package Description | Package Option |
|--------------------|-------------------|---|----------------|
| AD9542BCPZ | -40°C to +85°C | 48-Lead Lead Frame Chip Scale Package [LFCSP] | CP-48-13 |
| AD9542BCPZ-REEL7 | -40°C to +85°C | 48-Lead Lead Frame Chip Scale Package [LFCSP] | CP-48-13 |
| AD9542/PCBZ | | Evaluation Board | |

¹ Z = RoHS Compliant Part.

¹C refers to a communications protocol originally developed by Philips Semiconductors (now NXP Semiconductors).