# 4-Channel, 12-/16-Bit, Current Output DAC with On-Chip Reference and SPI

## FEATURES

- 4-channel, 12-/16-bit, current output DAC
- Programmable output current ranges per channel
  - 50mA, 150mA, 200mA, and 300mA
- Flexible 1.65V to AVDD (≤3.6V) output supply voltages
- Separate voltage supply per output channel
- 0.3V maximum dropout voltage per channel
- Internal switches to ground on IOUT pins
- 2.5V on-chip voltage reference
- 50MHz SPI interface
- Load DAC, A/B toggle, and dither functions
- Integrated analog diagnostic monitoring
- Integrated thermal warning alert
- 36-ball 2.55mm × 2.55mm WLCSP
- 28-pin 4mm × 4mm LFCSP
- Operating temperature: –40ºC to +125ºC

## APPLICATIONS

- Photonics control
- Optical communications
- LED driver programmable current source
- Current-mode biasing

## GENERAL DESCRIPTION

The AD5705R/AD5706R are 4-channel, 12-/16-bit resolution, low noise programmable current output, digital-to-analog converters (DACs). The devices are ideal for applications such as photonics control and current-mode biasing.

The devices feature an integrated 2.5V voltage reference, along with load DAC, A/B toggle, and dither functions. An onboard multiplexer allows for monitoring of supply and output voltages, output currents, and the internal die temperature.

The AD5705R/AD5706R support several programmable output current ranges, with a maximum of 300mA. With addressable pins, up to four devices can share a single SPI bus for communication.

Each DAC channel operates independently with its own positive power supply rail (PVDDx) that can range from 1.65V to 3.6V, optimizing both power efficiency and thermal performance. The devices require a 2.9V to 3.6V AVDD supply and are rated for operation across a wide temperature range from −40°C to +125°C.

***Table 1. Family Models***

| Model | Description |
|---|---|
| AD5705R | 12-bit, 4-channel current output DAC |
| AD5706R | 16-bit, 4-channel current output DAC |

## FUNCTIONAL BLOCK DIAGRAM

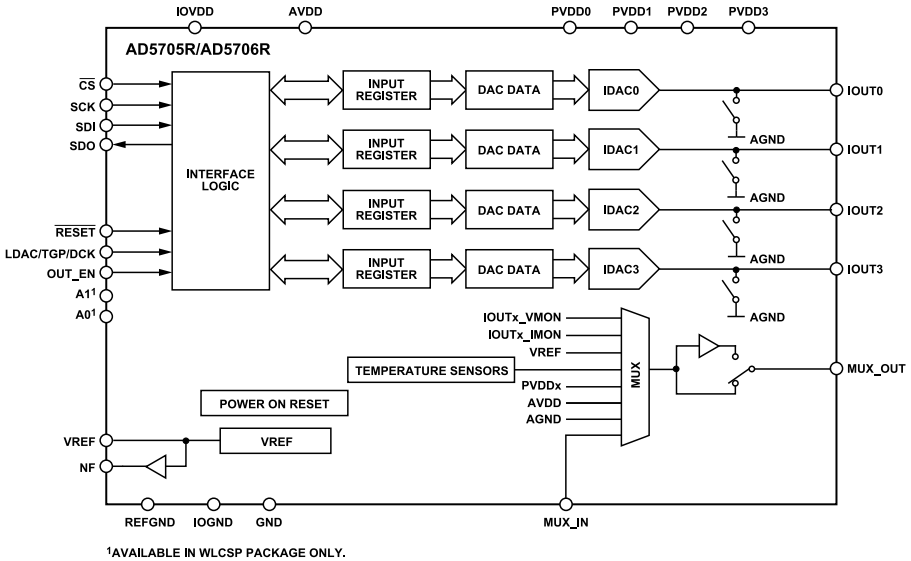



***Figure 1. Functional Block Diagram***

## TABLE OF CONTENTS



## REVISION HISTORY

**4/2026—Rev. 0 to Rev. A**

Added AD5705R (Universal) ... 1
Changes to Features Section ... 1
Changes to General Description Section ... 1
Added Table 1; Renumbered Sequentially ... 1
Changes to Table 2 ... 4
Changes to DAC to DAC Crosstalk (Current Area) Parameter, Table 3 ... 6
Changes to Note 4, Table 4 ... 7
Changes to $t_{DAC\_ACT}$ Parameter, Table 5 ... 9
Changes to Figure 18 and Figure 19 ... 15
Added Figure 20 and Figure 21; Renumbered Sequentially ... 15
Changes to Figure 32 Caption and Figure 33 Caption ... 17
Changes to Device Information Section ... 25

## TABLE OF CONTENTS



**1/2026—Revision 0: Initial Version**

## SPECIFICATIONS

AVDD = 2.9V to 3.6V, PVDDx = 1.65V to AVDD, IOVDD = 1.14V to 1.89V, VREF = 2.5V (internal or external), operating temperature range = −40°C ($T_A$) to +125°C ($T_J$), and external passive components $C_{REF}$ = 1µF, $C_{NF}$ = 0.1µF, unless otherwise noted. Typical values specified at $T_A$ = 25°C with nominal supply conditions AVDD = 3.3V, PVDDx = 3.3V, and IOVDD = 1.8V.

### DC SPECIFICATIONS

*Table 2. DC Specifications*

| Parameter[1] | Min | Typ | Max | Unit | Test Conditions/Comments |
|---|---|---|---|---|---|
| OPERATING TEMPERATURE RANGE | | | | | |
| | −40 | | | °C | Minimum operating ambient temperature ($T_{A\_MIN}$) |
| | | | 125 | °C | Maximum operating junction temperature ($T_{J\_MAX}$) |
| DAC STATIC PERFORMANCE | | | | | |
| Resolution | | | | | |
| AD5706R | 16 | | | Bits | All output current ranges |
| AD5705R | 12 | | | Bits | All output current ranges |
| Integral Nonlinearity (INL)[2] | | | | | |
| AD5706R | −32 | | +32 | $LSB_{16}$[3] | |
| AD5705R | −8 | | +8 | $LSB_{12}$[3] | |
| Differential Nonlinearity (DNL)[2] | | | | | Per channel, guaranteed monotonic |
| AD5706R | −1 | ±0.25 | +1 | $LSB_{16}$[3] | |
| AD5705R | −0.5 | | +0.5 | $LSB_{12}$[3] | |
| Total Unadjusted Error (TUE) | −1 | | +1 | %FSR | |
| Offset Error | −0.4 | | +0.4 | %FSR | |
| Offset Error Drift | | 5 | | ppm/°C | |
| Gain Error[2] | −0.9 | | +0.9 | %FSR | |
| Gain Error Drift | | 10 | | ppm/°C | |
| DC Crosstalk[4] | | 110 | | µA | Result of a 638mW change in dissipated power, with internal reference |
| DC Power Supply Rejection Ratio (PSRR) | | | | | 50mA range, DAC code = 3/4 full scale |
| AVDD | | 5 | | µA/V | AVDD = 3.3V ± 10% |
| PVDDx | | 3 | | µA/V | PVDD = 3.3V ± 10% |
| DAC OUTPUT CHARACTERISTICS | | | | | |
| Output Current ($I_{OUT}$) Ranges[5] | | | | | |
| | 0 | | +50 | mA | |
| | 0 | | +150 | mA | |
| | 0 | | +200 | mA | |
| | 0 | | +300 | mA | |
| Output Compliance Voltage[6] | 0 | | PVDDx − 0.3 | V | All current ranges |
| Ground Switch Resistance | | 10 | | Ω | Sinking 100mA from IOUTx to GND |
| INTERNAL REFERENCE OUTPUT | | | | | Internal reference enabled, decoupling 1µF |
| Voltage ($V_{REF}$) | 2.49 | 2.5 | 2.51 | V | $T_A$ = 25°C |
| Temperature Coefficient | | 15 | | ppm/°C | Across operating temperature range |
| Output Current Load Capability | | +5 | | mA | Across operating temperature range |
| Line Regulation | | 2 | | µV/V | Due to change in AVDD |
| Load Regulation, Sourcing | | 2 | | ppm/mA | $\Delta V_{OUT}/\Delta I_{LOAD}$ |
| EXTERNAL REFERENCE INPUT | | | | | Internal reference disabled (default) |
| Input Voltage | 2.495 | 2.5 | 2.505 | V | $T_A$ = 25°C |
| Input Impedance | | 16 | | kΩ | |
| TEMPERATURE SENSING DIODE | | | | | |
| Accuracy | | 5 | | °C | $T_A$ = 25°C |

## SPECIFICATIONS

*Table 2. DC Specifications (Continued)*

| Parameter[1] | Min | Typ | Max | Unit | Test Conditions/Comments |
|---|---|---|---|---|---|
| Reading | | 1.395 | | V | MUX_OUT reading at $T_A$ = 25°C |
| Temperature Coefficient | | −3.8 | | mV/°C | |
| Range | −40 | | | °C | $T_A$ |
| | | | 125 | °C | $T_J$ |
| THERMAL ALARMS | | | | | |
| Overheat Warning Temperature | | 125 | | °C | $T_J$ |
| Overheat Warning Hysteresis | | 5 | | °C | $T_J$ |
| MUX OUTPUT (MUX_OUT) | | | | | |
| DC Output Impedance | | 1 | | Ω | Buffered, output drive = ±1mA |
| | | 7 | | kΩ | Nonbuffered, output drive = ±1µA |
| Leakage Current | | 1 | | nA | Nonbuffered, $T_A$ = 125°C |
| Output Voltage Offset | −10 | | +10 | mV | Buffered |
| Output Drive Current | | ±5 | | mA | Buffered |
| MUX INPUT (MUX_IN) | | | | | |
| Input Voltage Range | 0 | | VREF | V | |
| Input Leakage Current | | 10 | | nA | |
| Input Impedance | | 2 | | kΩ | Nonbuffered MUX signal path |
| LOGIC INPUTS[7] | | | | | |
| Input Current | −1 | | +1 | µA | Per pin |
| Input Low Voltage ($V_{IL}$) | 0 | | 0.3 × IOVDD | V | |
| Input High Voltage ($V_{IH}$) | 0.7 × IOVDD | | IOVDD | V | |
| Pin Capacitance | | 5 | | pF | Per pin |
| LOGIC OUTPUTS | | | | | |
| Output Low Voltage ($V_{OL}$) | | | 0.4 | V | $I_{SINK}$ = 100µA |
| Output High Voltage ($V_{OH}$) | IOVDD − 0.4 | | | V | $I_{SOURCE}$ = 100µA |
| Floating State Output Capacitance | | 5 | | pF | |
| POWER SUPPLIES | | | | | |
| AVDD | 2.9 | 3.3 | 3.6 | V | |
| PVDDx[8] | 1.65 | 3.3 | AVDD | V | |
| IOVDD | 1.14 | 1.8 | 1.89 | V | |
| POWER PERFORMANCE | | | | | |
| AVDD Supply Current | | 29.7 | 37 | mA | Internal reference, all IOUTx outputs enabled, and at zero scale or in high-Z state |
| | | 29.4 | | mA | External reference, all IOUTx outputs enabled, and at zero scale or in high-Z state |
| PVDDx Supply Current | | 0.27 | 1.7 | mA | Per PVDD pin, all IOUTx outputs enabled, and at zero scale or high-Z state |
| IOVDD Supply Current | | 0.05 | 30 | µA | Digital inputs = 0V or IOVDD |
| Quiescent Power Dissipation | | 100 | | mW | AVDD = 3.3V, internal reference, all IOUTx outputs enabled, and at zero scale or in high-Z state |

[1] See the Terminology section.

[2] Linearity is defined from Code 24 to Code 4095 for AD5705R and Code 384 to Code 65535 for AD5706R. Offset Error is computed at Code 24 for AD5705R and Code 384 for AD5706R.

[3] LSB size varies with DAC resolution and current output range.

[4] Crosstalk is measured with a step change from 0mA to 300mA on all aggressor channels, with PVDD = 3.3V and $R_{LOAD}$ = 3.9Ω. Total power dissipation change is 638mW. The monitored channel is held at midscale on 300mA range, with AVDD = 3.5V, PVDD = 3.3V, and $R_{LOAD}$ = 8.33Ω.

[5] See Table 16.

## SPECIFICATIONS

[6] When sourcing current, the output compliance voltage is the maximum voltage at the IOUTx pin, for which the output current is within 0.1% of the measured full-scale range.

[7] See the Timing Specifications section for more details.

[8] PVDD must either be tied to AVDD or sourced to a lower level than AVDD.

## AC SPECIFICATIONS

*Table 3. AC Specifications*

| Parameter [1] | Min | Typ | Max | Unit | Test Conditions/Comments |
|---|---|---|---|---|---|
| DAC DYNAMIC PERFORMANCE | | | | | |
| Output Current Settling Time | | | | | DAC code changed from zero scale to full scale and vice-versa, with settling to ± $4LSB_{16}$[2] |
| | | 9 | | µs | 0mA to +200mA range |
| | | 11 | | µs | 0mA to +50mA range |
| Slew Rate | | | | | $I_{OUT}$ = 10% to 90% of FSR and vice-versa |
| | | 285 | | mA/µs | 0mA to +200mA range |
| | | 67 | | mA/µs | 0mA to +50mA range |
| Digital-to-Analog Glitch Energy | | 6 | | pA-sec | |
| Digital Feedthrough (Current Area) | | 0.03 | | nA-sec | |
| Digital Crosstalk | | 0.03 | | nA-sec | |
| DAC to DAC Crosstalk (Current Area) | | | | | For 200mA range on adjacent DAC channel, AD5706R |
| | | 40 | | pA-sec | DAC code changed from 1/4 scale to 3/4 scale in a step change and vice-versa |
| | | 70 | | pA-sec | DAC code changed from zero scale to full scale in a step change and vice-versa |
| MUX to DAC Crosstalk (Current Area) | | | | | DAC code = midscale |
| | | 0.03 | | nA-sec | 0mA to +300mA range |
| AC PSRR | | | | | PVDD = 3V, refer Figure 39 and Figure 40. |
| PVDDx | | −72 | | dB | f = 100kHz, 0mA to +300mA range |
| | | −80 | | dB | f = 100kHz, 0mA to +50mA range |
| AVDD | | −82 | | dB | f = 10kHz |
| Output Noise Spectral Density (NSD) | | | | | DAC code = 3/4 full scale, no external filter |
| | | 7.7 | | nA/√Hz | 1kHz, 0mA to +300mA range |
| | | 6 | | nA/√Hz | 10kHz, 0mA to +300mA range |
| | | 5.3 | | nA/√Hz | 1kHz, 0mA to +200mA range |
| | | 4.1 | | nA/√Hz | 10kHz, 0mA to +200mA range |
| | | 4.2 | | nA/√Hz | 1kHz, 0mA to +150mA range |
| | | 3.2 | | nA/√Hz | 10kHz, 0mA to +150mA range |
| | | 2.2 | | nA/√Hz | 1kHz, 0mA to +50mA range |
| | | 1.2 | | nA/√Hz | 10kHz, 0mA to +50mA range |
| Output Noise (0.1Hz to 10Hz) | | | | | DAC code = 3/4 full scale, no external filter |
| | | 620 | | nA rms | 0mA to +300mA range |
| | | 425 | | nA rms | 0mA to +200mA range |
| | | 332 | | nA rms | 0mA to +150mA range |
| | | 148 | | nA rms | 0mA to +50mA range |
| REFERENCE OUTPUT | | | | | Internal reference enabled, decoupling 1µF |
| Output Voltage Noise (0.1Hz to 10Hz) | | 4 | | µV rms | |
| Output Voltage Noise Density | | 112 | | nV/√Hz | f = 1kHz |
| Turn On Settling Time | | 500 | | µs | Decoupling 1µF, settling to ±0.1% |
| MUX OUTPUT CHARACTERISTICS | | | | | |
| −3dB Input Bandwidth | | 2.2 | | MHz | Buffered, $C_{LOAD}$ = 3pF |
| | | 1.4 | | MHz | Nonbuffered, standalone, $C_{LOAD}$ = 3pF |

## SPECIFICATIONS

*Table 3. AC Specifications (Continued)*

| Parameter [1] | Min | Typ | Max | Unit | Test Conditions/Comments |
|---|---|---|---|---|---|
| Maximum Output Capacitive Load | | 100 | | pF | Buffered |

[1] See the Terminology section.

[2] LSB size varies with current output range.

## TIMING SPECIFICATIONS

AVDD = 2.9V to 3.6V, PVDDx = 1.65V to AVDD, IOVDD = 1.14V to 1.89V, and operating temperature range = −40°C ($T_A$) to +125°C ($T_J$), unless otherwise noted. All timing values are specified for a standalone device.

All digital input signals are specified with rise time ($t_R$) = fall time ($t_F$) = 1ns/V (10% to 90% of IOVDD) and timed from a voltage level of ($V_{IL}$ + $V_{IH}$)/2.

### Serial Interface Timing

See Figure 2 to Figure 5 for related timing diagrams.

*Table 4. Digital Interface Timing Specifications*

| Parameter | Description | Min | Typ | Max | Unit |
|---|---|---|---|---|---|
| $t_{SCK}$ | SCK cycle time, write operation.[1] | 20 | | | ns |
| | SCK cycle time, write and read operations.[2] | 40 | | | ns |
| $f_{SCK}$ | SCK frequency, write operation.[1] | | | 50 | MHz |
| | SCK frequency, write and read operations.[2] | | | 25 | MHz |
| $t_{SCKH\_PW}$ | SCK high time. | 8 | | | ns |
| $t_{SCKL\_PW}$ | SCK low time. | 8 | | | ns |
| $t_{CSL\_SCKH}$ | $\overline{CS}$ falling edge to SCK rising edge setup time | 6 | | | ns |
| $t_{CSH\_SCKH}$ | $\overline{CS}$ rising edge to SCK rising edge setup time. | 10 | | | ns |
| $t_{SCKH\_CSH}$ | Rising edge of SCK[3] to rising edge of $\overline{CS}$. | 10 | | | ns |
| $t_{SCKH\_CSL}$ | Rising edge of SCK to falling edge of $\overline{CS}$. | 10 | | | ns |
| $t_{SDI\_SET}$ | SDI data setup time. | 5 | | | ns |
| $t_{SDI\_HOL}$ | SDI data hold time. | 5 | | | ns |
| $t_{CSH\_PW}$ | $\overline{CS}$ pulse width high. | 10 | | | ns |
| $t_{SCKL\_SDO}$ | SCK falling edge to SDO data available delay. | | | 15 | ns |
| $t_{SDO\_VALID}$ | SCK falling Edge to SDO data remains valid. | | | 12 | ns |
| $t_{CSH\_SDOZ}$ | $\overline{CS}$ rising edge to SDO disabled. | | | 14 | ns |
| $t_{SCKL\_SDOEN}$ | SCK falling edge to SDO enabled. | | | 14 | ns |
| $t_{RESETL\_PW}$ | $\overline{RESET}$ pulse width low. | 20 | | | ns |
| $t_{RESET\_ACT}$[4] | $\overline{RESET}$ assertion (low level triggered) to channel output getting reset, or rising edge of SCK[3] to channel output getting reset. | | <100 | | ns |
| $t_{RESET\_DELAY}$[5] | $\overline{RESET}$ pulse delay upon power-on reset. | | | 200 | µs |

[1] With send status and CRC disabled.

[2] With send status and CRC enabled.

[3] The 8th or 16th SCK, whichever shifts in the final bit of register data in the data phase.

[4] Channel output settling time is not included in RESET timing specifications and must be considered separately.

[5] Guaranteed by design; not production tested.

## SPECIFICATIONS

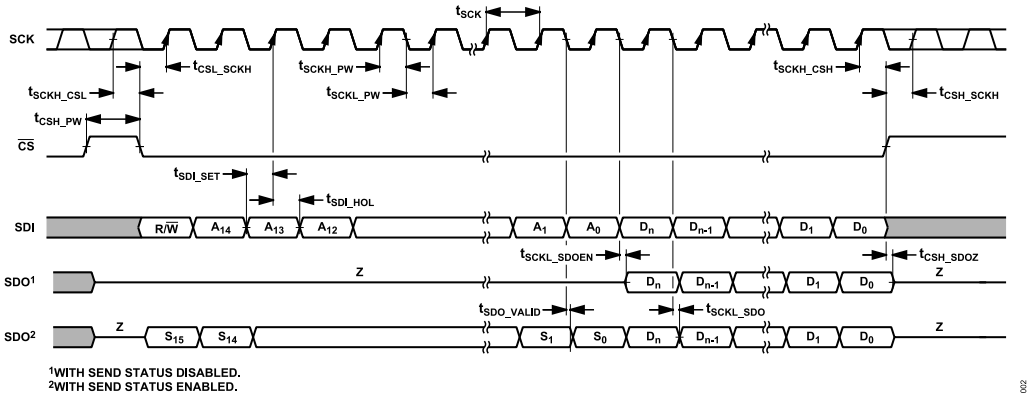



*Figure 2. Register Read/Write Timing Diagram (CPOL = 0 and CPHA = 0)*

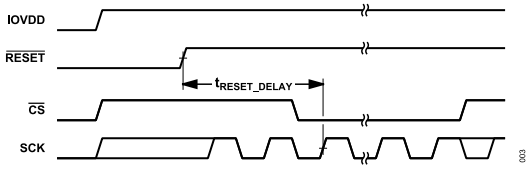



*Figure 3. Power On Reset (POR) Timing Diagram*

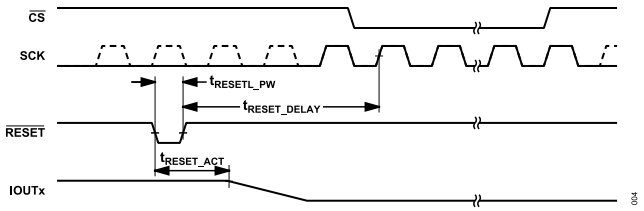



*Figure 4. Hardware Reset Timing Diagram*

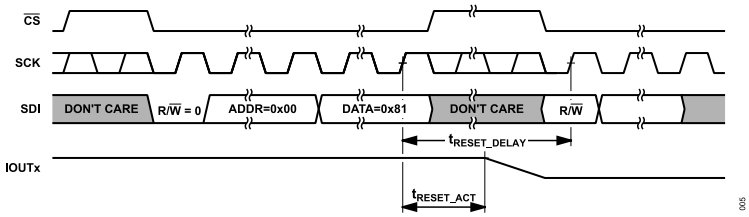



*Figure 5. Software Reset Timing Diagram*

## SPECIFICATIONS

### DAC Output Update Timing

See Figure 6 to Figure 15 for related timing diagrams.

*Table 5. DAC Output Update Timing Specifications*

| Parameter[1] | Description | Min | Typ | Max | Unit |
|---|---|---|---|---|---|
| $t_{OUTEN\_ACT}$[2] | Delay from OUT_EN low level trigger to DAC channel in high-Z state (measured on 200mA range) | | 350 | | ns |
| $t_{OUTENL\_SCKH}$[3] | Delay from the OUT_EN low level trigger to the rising edge of SCK[4] for transition of IOUTx from high-Z state to normal operation | 0 | | | ns |
| $t_{SCKH\_OUTENL}$[3] | Delay from the rising edge of SCK[4] to OUT__EN low for transition of IOUTx to high-Z state | 5 | | | ns |
| $t_{OUTEN\_PW}$[3] | OUT_EN pulse width low | 6 | | | µs |
| $t_{DAC\_ACT}$[2] | DAC update activation time (measured on 200mA range)[5] | | 60 | | ns |
| $t_{LDAC\_PW}$ | LDAC/TGP/DCK pulse width (low or high) | 20 | | | ns |
| $t_{LDACT\_SCKH}$ | Delay from the active edge of LDAC/TGP/DCK to the rising edge of SCK[4] in a SPI write transaction during hardware trigger modes | 5 | | | ns |
| $t_{SCKH\_LDACT}$ | Delay from the rising edge of SCK[4] to the following active edge of LDAC/TGP/DCK in a SPI write transaction during hardware trigger modes | 5 | | | ns |
| $t_{HW\_LDAC\_TRIG}$ | Time between the two consecutive active trigger edges of LDAC/TGP/DCK during hardware trigger modes | 80 | | | ns |
| $t_{SW\_LDAC\_TRIG}$ | Delay from one rising edge of SCK[4] to another rising edge of SCK[4] (in the consecutive data phase) in a SPI write transaction during software trigger modes | 16 × $t_{SCK}$ | | | |
| $t_{LDACT\_SCKH\_RD}$ | Delay from active edge of LDAC/TGP/DCK to the rising edge of SCK[4] in a SPI read transaction during hardware trigger modes | 5 | | | ns |
| $t_{SCKH\_LDACT\_RD}$ | Delay from the rising edge of SCK[4] to the next active edge of LDAC/TGP/DCK in a SPI read transaction during hardware trigger modes | 5 | | | ns |
| $t_{FUNC\_MODE\_EN1}$ | Delay from the active trigger edge of LDAC/TGP/DCK to the rising edge of SCK[4] for change in DAC function modes | 5 | | | ns |
| $t_{FUNC\_MODE\_EN2}$ | Delay from the rising edge of SCK[4] to the active trigger edge of LDAC/TGP/DCK for change in DAC function modes | 5 | | | ns |

1 See the DAC Operations section for a list of DAC update modes.

2 The timing does not include the analog IOUT settling time.

3 Guaranteed by design; not production tested.

4 The 8th or 16th SCK, whichever shifts in the final bit of register data in the data phase.

5 This specified activation time applies with the code region from the DAC Code 384 to Code 65535 for AD5706R and from the DAC Code 24 to Code 4095 for AD5705R, transitioning in either direction. The activation time is typical 230ns when updating from a code near zero scale to any code within the specified code ranges. Activation time refers to the delay between applying a new digital input code and the moment when the DAC output current first begins to change from its previous level. It does not represent the time required for the output to reach or settle to the final value.

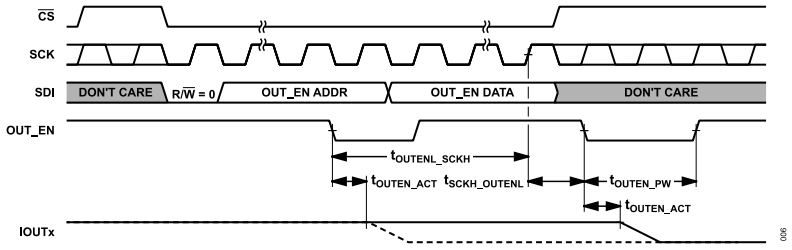



*Figure 6. IOUTx Enable/Disable Timing Diagram Using OUT_EN Pin*

## SPECIFICATIONS

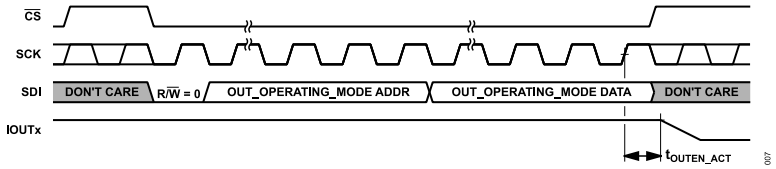



*Figure 7. IOUTx Enable/Disable Timing Diagram Using SPI Write*

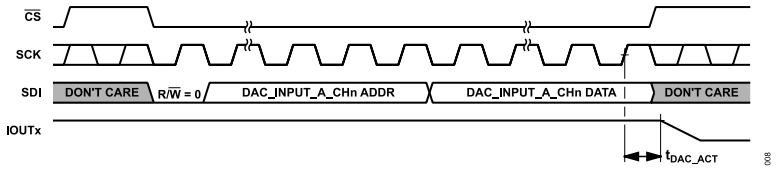



*Figure 8. Asynchronous DAC Update Timing Diagram*

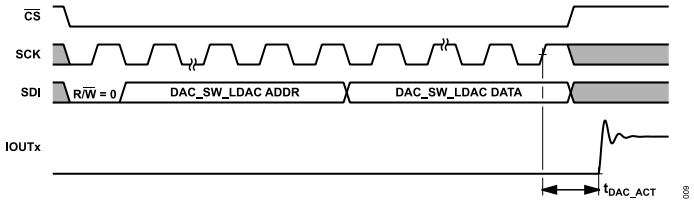



*Figure 9. Software LDAC Timing Diagram*

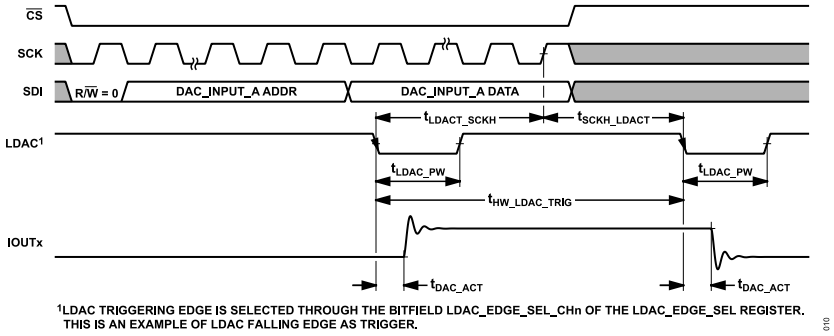



*Figure 10. Hardware LDAC Timing Diagram*

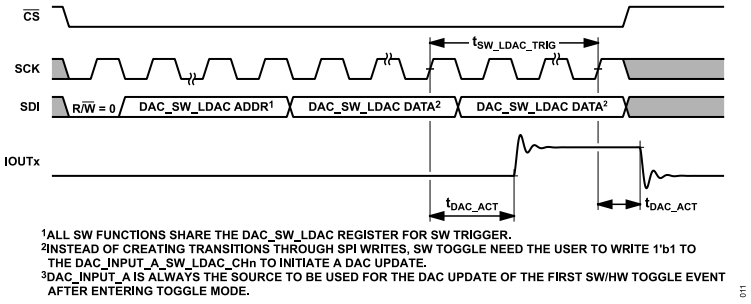
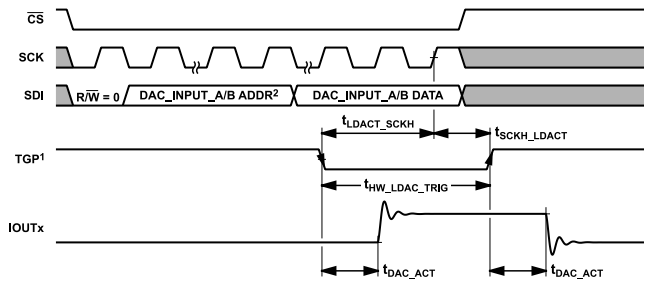



*Figure 11. Software DAC Update Timing Diagram in Streaming Mode*

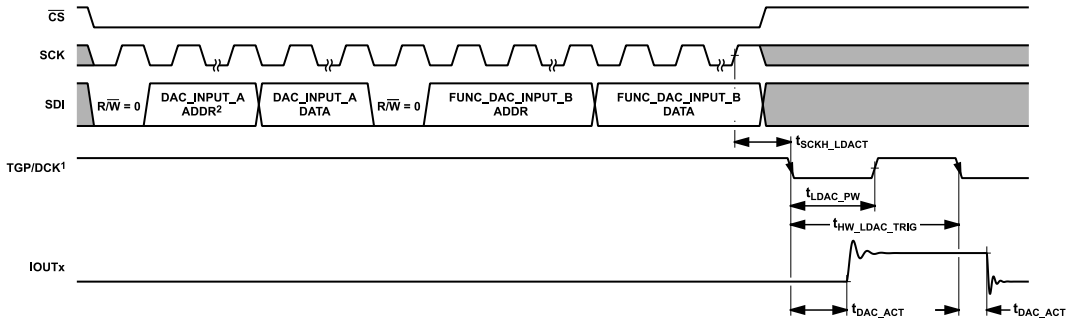
## SPECIFICATIONS




1TGP TRIGGER EDGE IS SELECTED THROUGH THE BITFIELD LDAC_EDGE_SEL_CHn OF THE LDAC_EDGE_SEL REGISTER. THIS IS AN EXAMPLE OF BOTH EDGES AS TRIGGER EVENTS.
2DAC_INPUT_A IS ALWAYS THE FIRST SOURCE TO BE USED FOR DAC UPDATE FOR THE FIRST SW/HW TOGGLE EVENT AFTER ENTERING TOGGLE MODE.

012

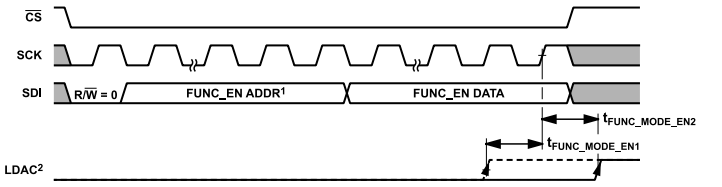
*Figure 12. Hardware Toggle Timing Diagram*




1TGP/DCK TRIGGER EDGE IS SELECTED THROUGH THE BITFIELD LDAC_EDGE_SEL_CHn OF THE LDAC_EDGE_SEL REGISTER. THIS IS AN EXAMPLE OF FALLING EDGE AS TRIGGER EVENT.
2DAC_INPUT_A IS ALWAYS THE FIRST SOURCE TO BE USED FOR DAC UPDATE FOR THE FIRST SW/HW TOGGLE OR DITHER EVENT AFTER ENTERING TOGGLE OR DITHER MODE RESPECTIVELY.

013

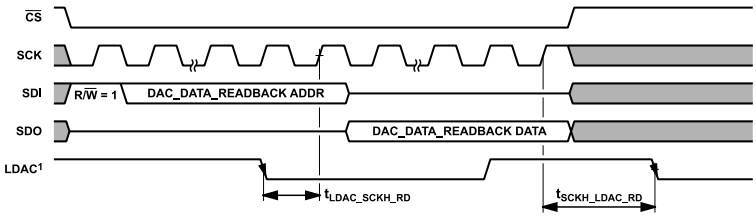
*Figure 13. Hardware Toggle and Dither Timing Diagram in Single Instruction Mode*




1TOGGLE AND DITHER MODES ARE ENABLED AND DISABLED THROUGH THE FUNC_EN REGISTER AFTER SELECTING A MODE IN THE FUNC_MODE_SEL REGISTER.
2LDAC/TGP/DCK TRIGGER EDGE IS SELECTED THROUGH THE BITFIELD LDAC_EDGE_SEL_CHn OF THE LDAC_EDGE_SEL REGISTER. THIS IS AN EXAMPLE OF RISING EDGE AS TRIGGER.

014

*Figure 14. DAC Mode Function Enable Timing Diagram*




1LDAC/TGP/DCK TRIGGER EDGE IS SELECTED THROUGH THE BITFIELD LDAC_EDGE_SEL_CHn OF THE LDAC_EDGE_SEL REGISTER. THIS IS AN EXAMPLE OF LDAC FALLING EDGE AS TRIGGER EVENT.

015

*Figure 15. DAC Data Read Back Timing Diagram*

## ABSOLUTE MAXIMUM RATINGS

$T_A$ = 25°C, unless otherwise noted

*Table 6. Absolute Maximum Ratings*

| Parameter | Rating |
|---|---|
| AVDD to AGND | −0.3 V to +3.96 V |
| PVDDx[1] to GND | −0.3 V to minimum (AVDD + 0.3 V, +3.96 V) |
| IOUTx to GND | −0.3 V to minimum (AVDD + 0.3 V, +4.26 V) |
| VREF[2], NF to REFGND | −0.3 V to minimum (AVDD + 0.3 V, +4.26 V) |
| MUX_IN, MUX_OUT to GND | −0.3 V to minimum (AVDD + 0.3 V, +4.26 V) |
| IOVDD to IOGND[3] | −0.3 V to +2.1 V |
| Digital Inputs[4] to IOGND | −0.3 V to minimum (IOVDD + 0.3 V, +2.4 V) |
| Digital Outputs to IOGND | −0.3 V to minimum (IOVDD + 0.3 V, +2.4 V) |
| Temperature | |
| Operating Range | −40°C ($T_A$) to +125°C ($T_J$) |
| Storage Range | −60°C to +150°C |
| Junction | 150°C |
| Lead, Soldering Reflow | 260°C, as per JEDEC J-STD-020 |

1 PVDDx refers to the PVDD0, PVDD1, PVDD2, and PVDD3 pins. IOUTx refers to the IOUT0, IOUT1, IOUT2, and IOUT3 pins.

2 when in External Reference mode

3 In LFCSP package, IOGND is same as GND.

4 See the Pin Configuration and Function Descriptions section for a list of the digital input and digital output pins.

Stresses at or above those listed under absolute maximum ratings may cause permanent damage to the device. This is a stress rating only; functional operation of the device at these or any other conditions above those indicated in the operational section of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

Absolute maximum ratings apply individually only, not in combination.

## THERMAL PERFORMANCE

The AD5705R/AD5706R can be damaged when the junction temperature ($T_J$) limits are exceeded. See Table 2 for the maximum operating junction temperature specification ($T_{J\_MAX}$). Monitoring ambient temperature does not guarantee that $T_J$ is within the specified maximum temperature limits. In applications with high power dissipation and poor thermal resistance, $T_J$ must be monitored using the internal temperature sensor.

## THERMAL RESISTANCE

Thermal performance is directly linked to printed circuit board (PCB) design and operating environment. Careful attention to PCB thermal design is required, in particular for applications where high maximum power dissipation exists.

Thermal resistance values specified in Table 7 are calculated based on JEDEC specifications and must be used in compliance with JESD51-12. The worst-case junction temperature is reported.

*Table 7. Thermal Resistance*

| Package Type[1] | $\theta_{JA}$ | $\theta_{JB}$ | $\Psi_{JB}$ | $\Psi_{JT}$ | $\Psi_{JC\text{-}TOP}$ | Unit |
|---|---|---|---|---|---|---|
| WLCSP | 42.26 | 7.9 | 8.1 | 0.06 | 0.16 | °C/W |
| LFCSP | 39.02 | 7.88 | 10.28 | 0.56 | 23.32 | °C/W |

1 The values in Table 7 are calculated based on the standard JEDEC 2S2P thermal test board with 4 thermal vias. See JEDEC JESD51 series

$\theta_{JA}$ and $\theta_{JB}$ are mainly used to compare thermal performance of the package of the device with other semiconductor packages when all test conditions listed are similar. $\theta_{JA}$ and $\theta_{JB}$ can be used for first-order approximation of the junction temperature in the system environment.

For WLCSP devices, using $\Psi_{JB}$ or $\Psi_{JT}$ is a more appropriate way to estimate the worst case junction temperature of the device in the system environment if an accurate thermal measurement of the board temperature near the device under test (DUT) or directly on the package top surface operating in the system environment is available.

Using the parameters listed in Table 7 in accordance with JEDEC standards in the JESD51 series is recommended.

## ESD CAUTION

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

## PIN CONFIGURATION AND FUNCTION DESCRIPTIONS

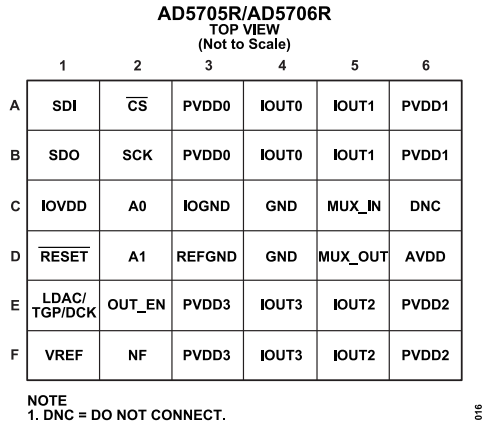



*Figure 16. WLCSP Pin Configuration*

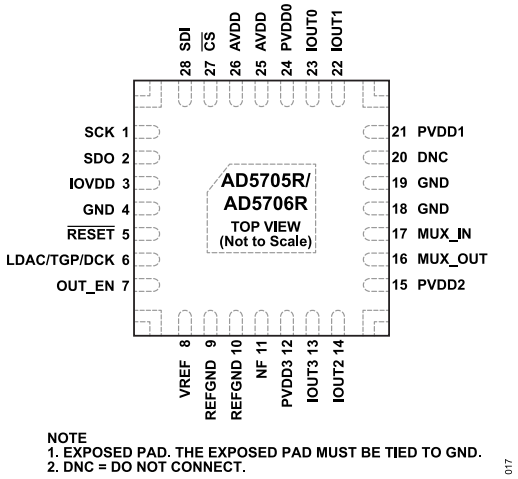



*Figure 17. LFCSP Pin Configuration*

*Table 8. AD5705R/AD5706R Pin Function Descriptions*

| Pin No. | | | | |
|---|---|---|---|---|
| **WLCSP** | **LFCSP** | **Mnemonic** | **Type[1]** | **Description** |
| D5 | 16 | MUX_OUT | AO | Analog Output with Internal Buffer. An external analog-to-digital converter (ADC) reads voltages on this pin for diagnostic purposes. |
| C5 | 17 | MUX_IN | AI | Analog Input. An external input within 0 to VREF range can be fed from another AD5705R/AD5706R MUX_OUT or any other analog output for daisy chaining purposes. |
| A4, B4 | 23 | IOUT0 | AO | Current Output of IDAC0 Channel. |
| A5, B5 | 22 | IOUT1 | AO | Current Output of IDAC1 Channel. |
| E5, F5 | 14 | IOUT2 | AO | Current Output of IDAC2 Channel. |
| E4, F4 | 13 | IOUT3 | AO | Current Output of IDAC3 Channel. |
| A3, B3 | 24 | PVDD0 | S | Power Supply for IDAC0 Channel. |
| A6, B6 | 21 | PVDD1 | S | Power Supply for IDAC1 Channel. |
| E6, F6 | 15 | PVDD2 | S | Power Supply for IDAC2 Channel. |
| E3, F3 | 12 | PVDD3 | S | Power Supply for IDAC3 Channel. |
| D6 | 25, 26 | AVDD | S | Analog Power Supply. |
| C1 | 3 | IOVDD | S | Logic Power Supply. IOVDD must be between 1.14V and 1.89V. This pin supplies power to the serial interface circuit blocks on the device. |
| C4, D4 | 4, 18, 19 | GND | S | Analog Supply Ground Pin. |

## PIN CONFIGURATION AND FUNCTION DESCRIPTIONS

*Table 8. AD5705R/AD5706R Pin Function Descriptions (Continued)*

| Pin No. | | | | |
|---|---|---|---|---|
| **WLCSP** | **LFCSP** | **Mnemonic** | **Type[1]** | **Description** |
| C3 | - | IOGND[2] | S | Digital Supply Ground Pin. |
| D3 | 9, 10 | REFGND | S | Analog Ground for DAC Core(s) and Internal Reference. REFGND and GND must be tied together at 0V using a low impedance connection. |
| F1 | 8 | VREF | AI/O | 2.5V Voltage Reference Input/Output. Connect a 1µF capacitor between this pin and REFGND. |
| F2 | 11 | NF | AI/O | Filter Capacitor for internal regulator. A 0.1µF capacitor connected from the NF pin to REFGND is recommended to achieve the specified performance from the AD5705R/AD5706R. |
| D1 | 5 | $\overline{\text{RESET}}$ | DI | Active Low Device Reset. Tie this pin high for normal operation. Asserting this pin low will reset the device to its default configuration. |
| A2 | 27 | $\overline{\text{CS}}$ | DI | Active Low Chip Select Input. This input is used to frame data during an SPI transaction. |
| B2 | 1 | SCK | DI | Serial Clock Input. Data can be transferred at rates of up to 50MHz when writing to the device. |
| A1 | 28 | SDI | DI | Serial Data Input. Data to be written to the device is provided on this input and is clocked into the register on the rising edge of SCK. |
| B1 | 2 | SDO | DO | Serial Data Output. A read back operation provides data on this output pin as a serial data stream. Data is clocked out on the falling edge of SCK and is valid on the rising edge of SCK. |
| E1 | 6 | LDAC/ TGP/DCK | DI | LDAC: Load DAC pin.<br>TGP: Toggle DAC pin.<br>DCK: Dither Clock Pin.<br>Pulsing this pin low or high based on trigger edge settings allows all selected DAC registers to be updated if the corresponding input registers have new data, allowing all selected DAC outputs to update synchronously. Alternatively, this pin can be tied to either IOVDD or IOGND. |
| E2 | 7 | OUT_EN | DI | Output Enable/Disable. OUT_EN pin is low level triggered for setting the current outputs to high-Z state or to GND. This pin can be tied high to IOVDD if not used. |
| D2 | - | A1 | DI | Logic Input: Programmable Address Bit 1. This pin must be tied to either IOVDD or IOGND. |
| C2 | - | A0 | DI | Logic Input: Programmable Address Bit 0. This pin must be tied to either IOVDD or IOGND. |
| C6 | 20 | DNC | DNC | Do Not Connect. |

[1] AO is analog output, S is supply, AI is analog input, AI/O is analog input or output, DI is digital input, DO is digital output, DI/O is digital input or output, and DNC is do not connect.

[2] IOGND and GND are tied internally in LFCSP package.

## TYPICAL PERFORMANCE CHARACTERISTICS

AVDD = 3.3V, PVDDx = 3.0V, IOVDD = 1.8V, internal reference, and temperature = 25 °C ($T_A$), unless otherwise noted.

### DC PERFORMANCE

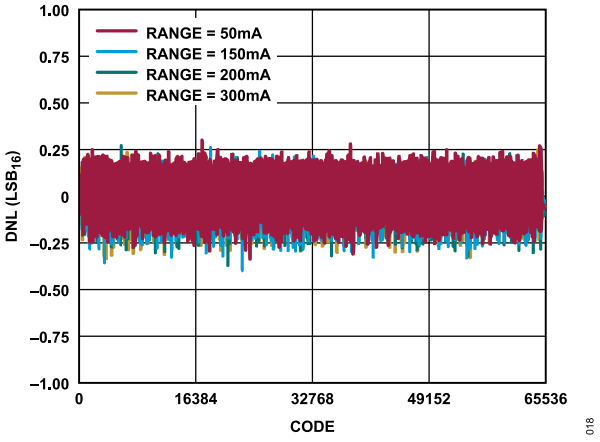



Figure 18. DAC DNL Error vs. DAC Code for AD5706R

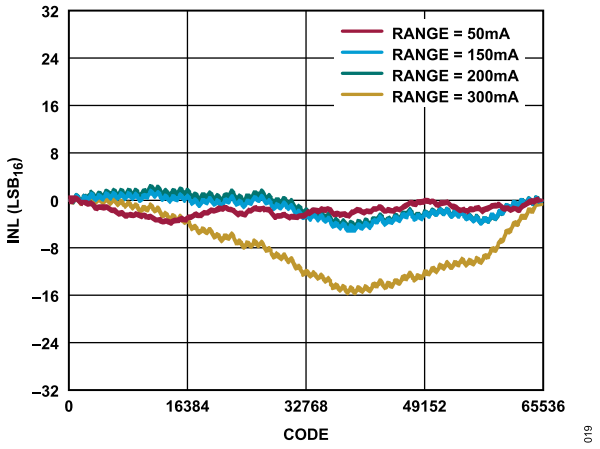



Figure 19. DAC INL Error vs. DAC Code for AD5706R

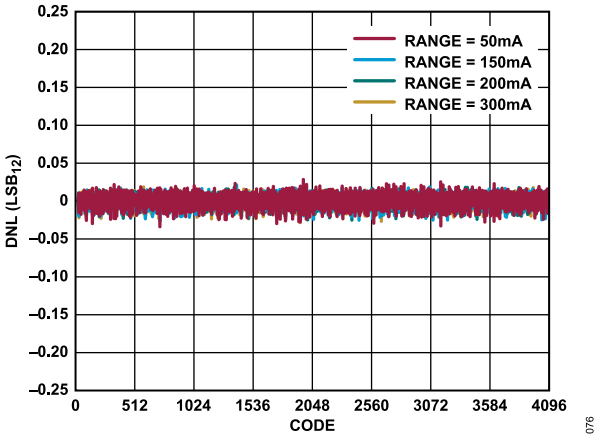



Figure 20. DAC DNL Error vs. DAC Code for AD5705R

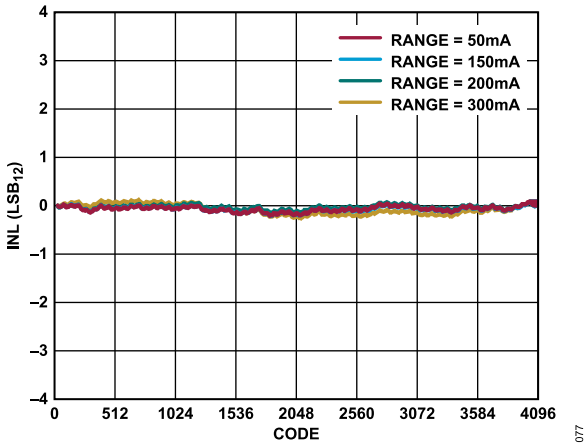



Figure 21. DAC INL Error vs. DAC Code for AD5705R

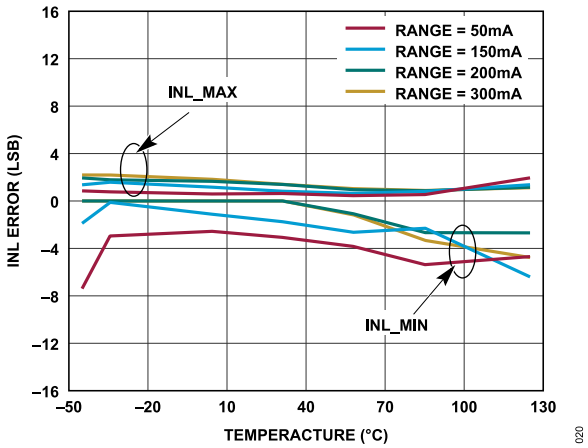



Figure 22. DAC INL Error vs. Temperature

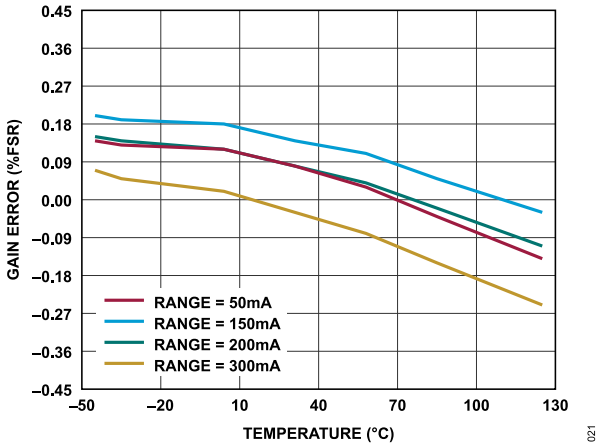



Figure 23. DAC Gain Error vs. Temperature

## TYPICAL PERFORMANCE CHARACTERISTICS

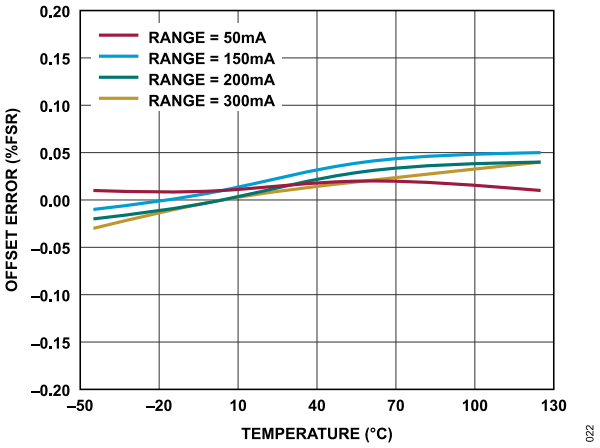



*Figure 24. DAC Offset Error vs. Temperature*

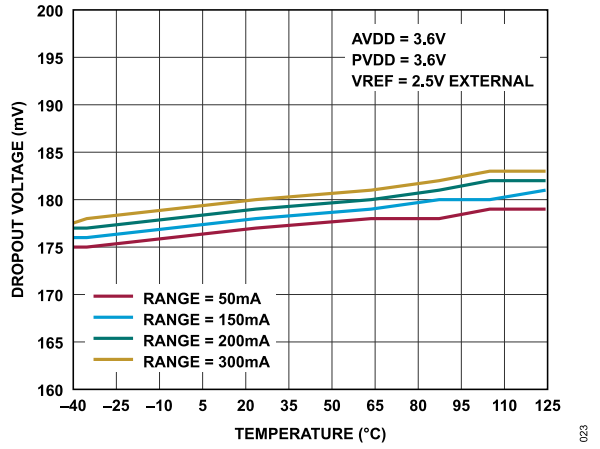



*Figure 25. PVDD Dropout Voltage vs. Temperature at PVDD = 3.6V*

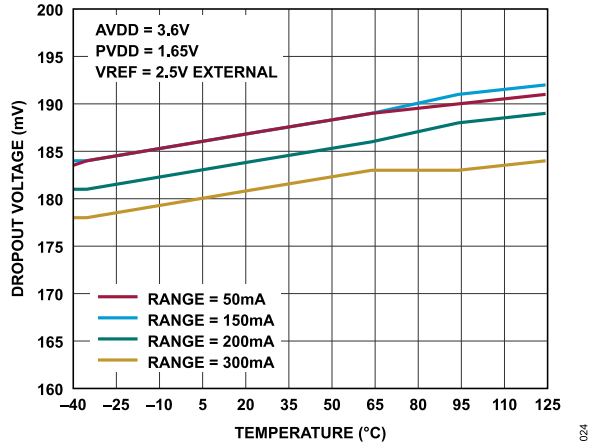



*Figure 26. PVDD Dropout Voltage vs. Temperature at PVDD = 1.65V*

## TYPICAL PERFORMANCE CHARACTERISTICS

### AC PERFORMANCE

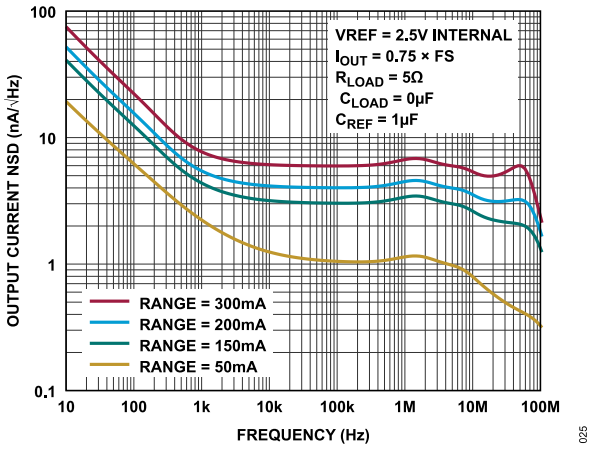



*Figure 27. DAC Output Noise Spectral Density vs. Frequency*

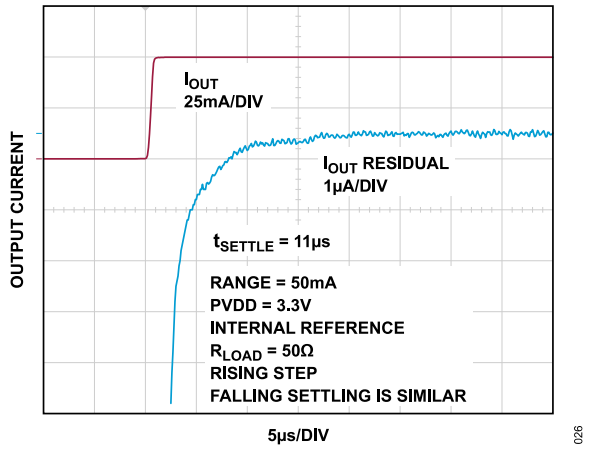



*Figure 28. DAC Output Settling Time, 50mA Range*

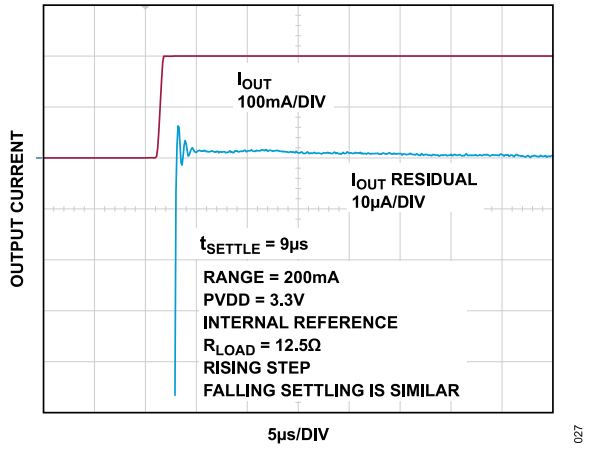



*Figure 29. DAC Output Settling Time, 200mA Range*

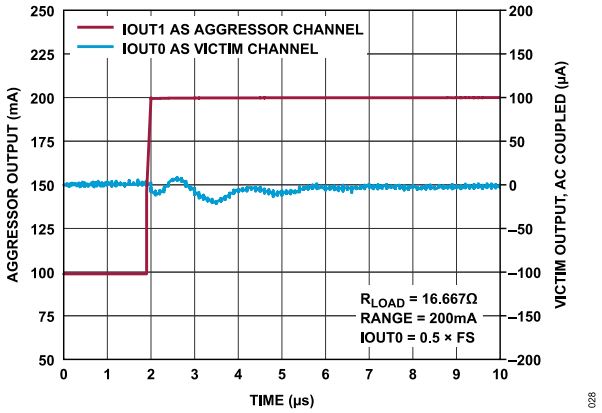



*Figure 30. DAC Channel-to-Channel Crosstalk, Positive Output Transition*

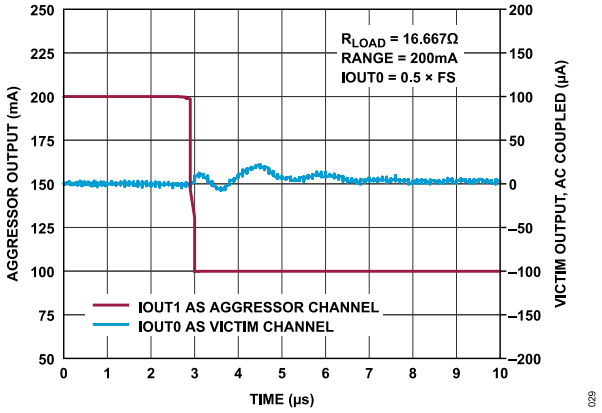



*Figure 31. DAC Channel-to-Channel Crosstalk, Negative Output Transition*

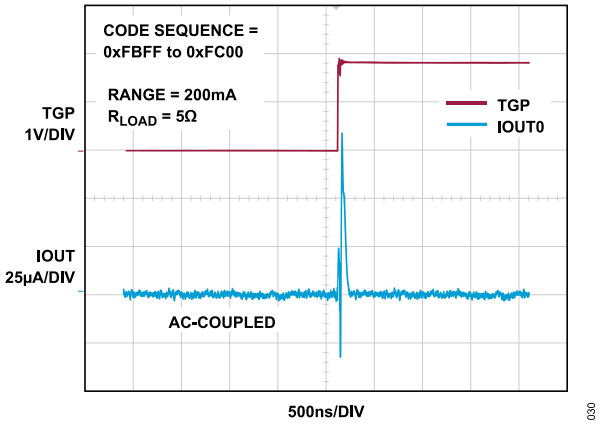



*Figure 32. DAC Glitch on Positive Output Transition, AD5706R*

## TYPICAL PERFORMANCE CHARACTERISTICS

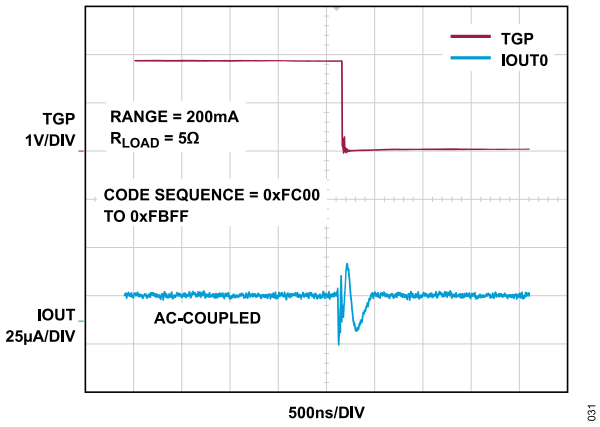



*Figure 33. DAC Glitch on Negative Output Transition, AD5706R*

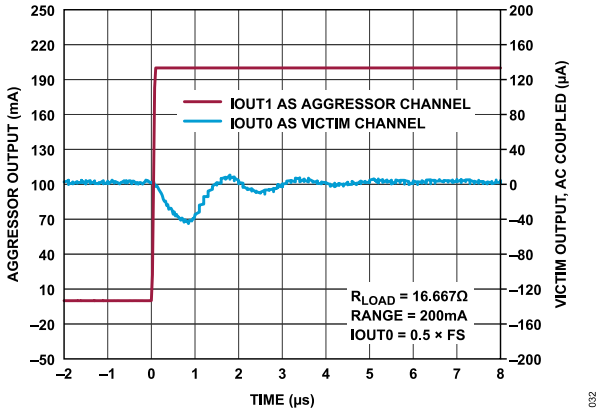



*Figure 34. Analog Crosstalk, Positive Output Transition*

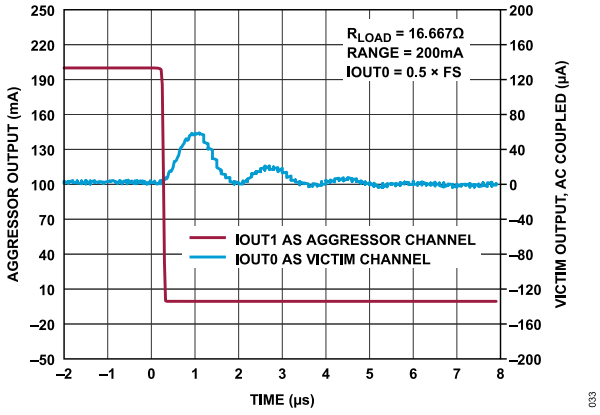



*Figure 35. Analog Crosstalk, Negative Output Transition*

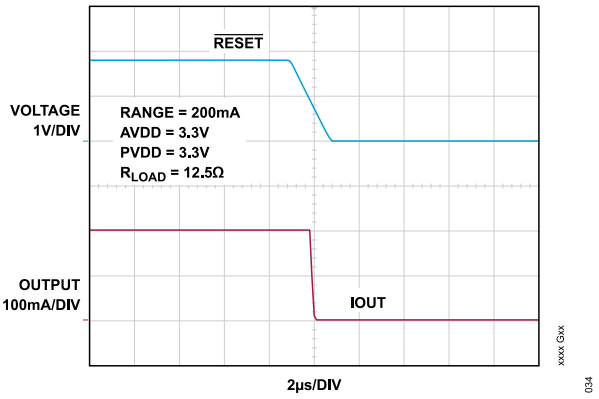



*Figure 36. DAC Response Time on Device Reset*

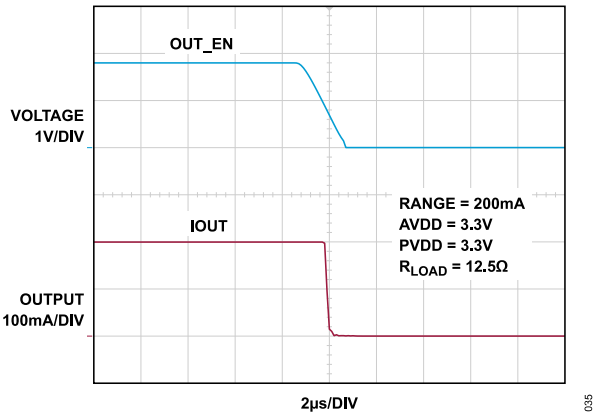



*Figure 37. DAC Response Time on Output Disable*

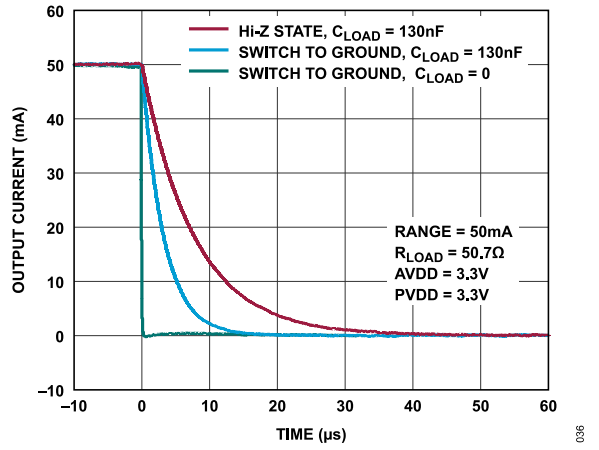



*Figure 38. DAC Response on Output Disable over Time*

## TYPICAL PERFORMANCE CHARACTERISTICS

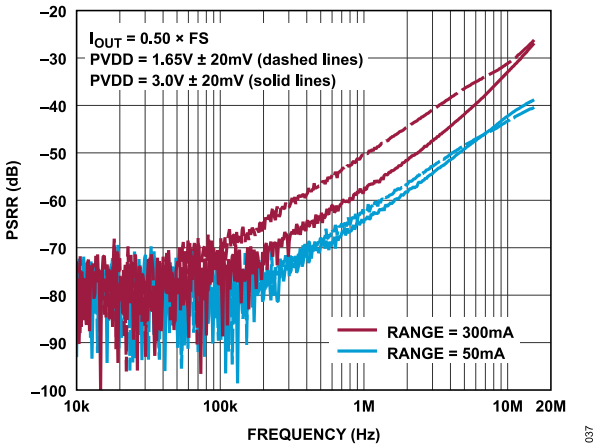



Figure 39. DAC AC PSRR vs. Frequency at DAC Output = 0.50 × FS

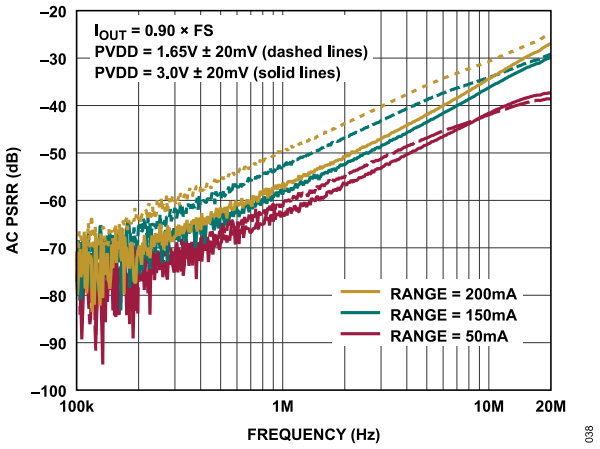



Figure 40. DAC AC PSRR vs. Frequency at DAC Output = 0.90 × FS

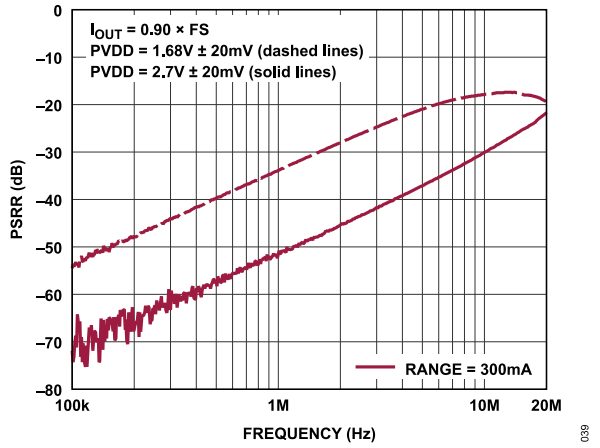



Figure 41. DAC AC PSRR vs. Frequency at DAC Output = 0.90 × FS of 300mA Range

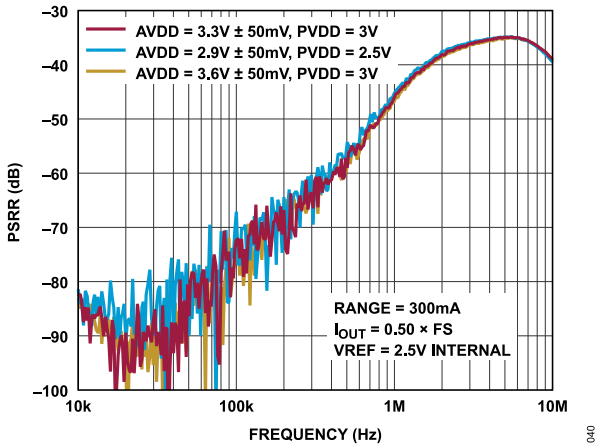



Figure 42. DAC AC PSRR vs. Frequency Across AVDD Supply

## TYPICAL PERFORMANCE CHARACTERISTICS

### MISCELLANEOUS

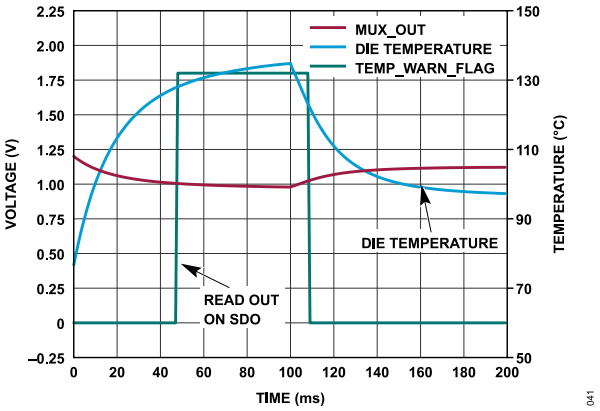



*Figure 43. Temperature Sensor Response over Time*

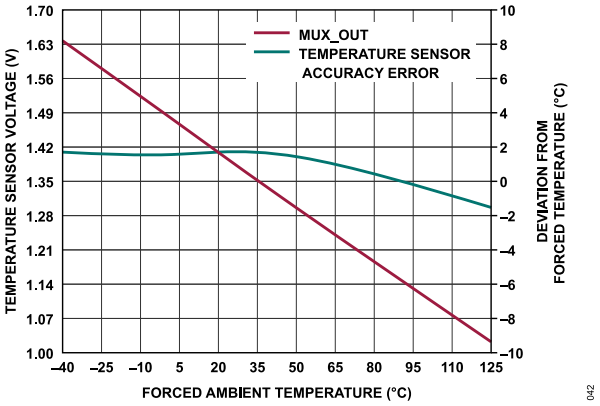



*Figure 44. Temperature Sensor Voltage vs. Forced Ambient Temperature*

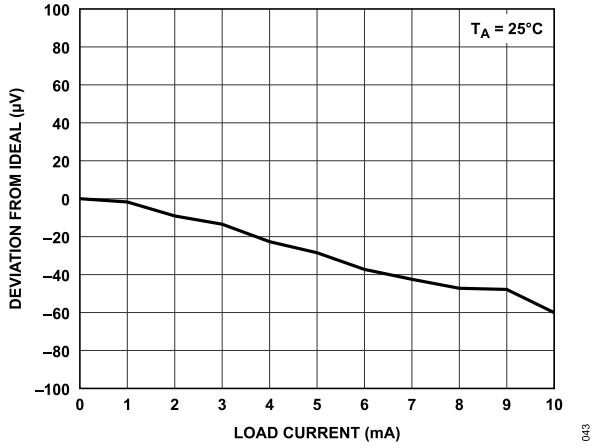



*Figure 45. VREF Accuracy Error vs. Load Current*

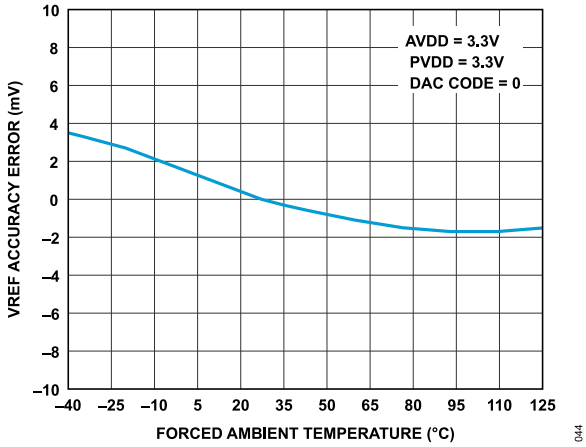



*Figure 46. VREF Accuracy Error vs. Forced Ambient Temperature*

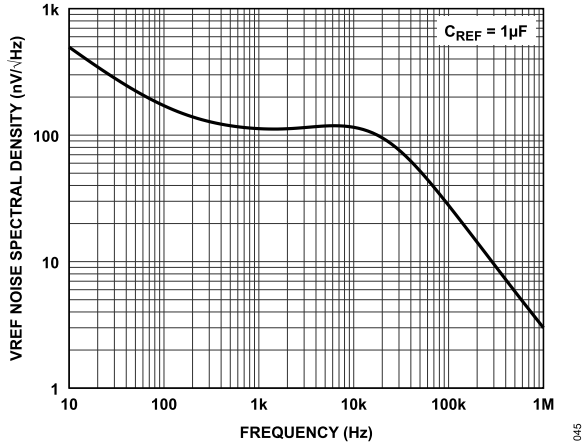



*Figure 47. VREF Noise Spectral Density vs. Frequency*

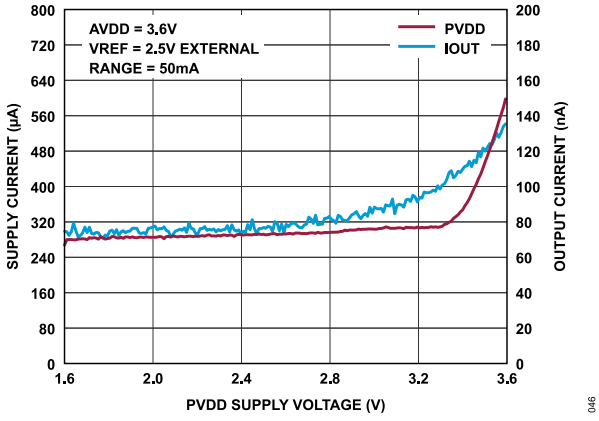



*Figure 48. PVDD and IOUT Response in High-Z vs. PVDD Supply Voltage*

## TYPICAL PERFORMANCE CHARACTERISTICS

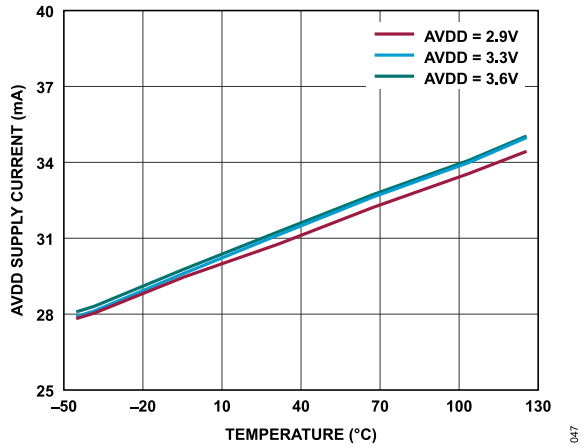



*Figure 49. AVDD Supply Current vs. Temperature*

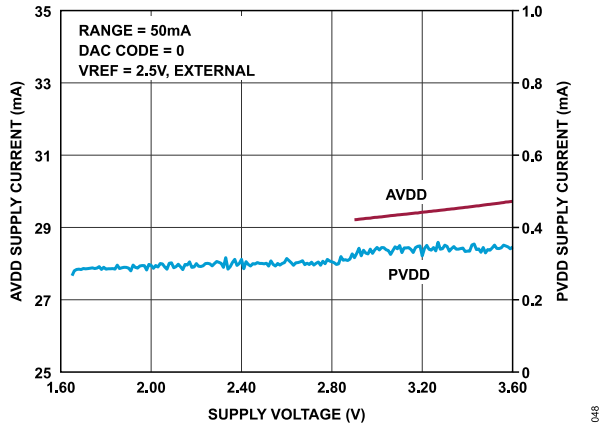



*Figure 50. AVDD Supply Current vs. Supply Voltage*

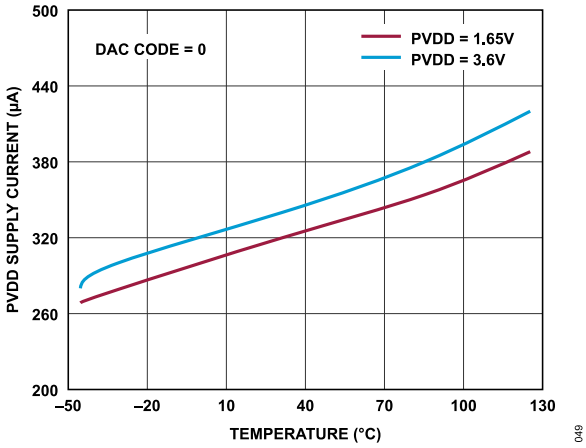



*Figure 51. PVDD Supply Current vs. Temperature*

## TERMINOLOGY

### DAC

### Full-Scale Range (FSR)

FSR is the maximum output of the DAC when the full-scale code (0xFFFF for a 16-bit DAC) is loaded to the DAC register.

### Total Unadjusted Error (TUE)

TUE is a measure of the output error taking all the various errors into account, namely INL error, offset error, gain error, and output drift over supplies, temperature, and time. TUE is expressed as a percentage of full-scale range (% FSR).

### Relative Accuracy or Integral Nonlinearity (INL)

INL is a measurement of the maximum deviation, in LSBs, from a straight line passing through the endpoints of the DAC transfer function.

### Differential Nonlinearity (DNL)

DNL is the difference between the measured change and the ideal 1 LSB change between any two adjacent codes. A specified differential nonlinearity of ±1 LSB maximum ensures monotonicity. This DAC is guaranteed monotonic by design.

### Gain Error

Gain error is a measure of the span error of the DAC. It is the deviation in slope of the DAC transfer characteristic from the ideal expressed as %FSR.

### Gain Error Drift

Gain error drift is a measurement of the change in gain error with changes in temperature. It is expressed in ppm of FSR/°C.

### Offset Error

Offset error is a measurement of the difference between the actual and ideal current output of IOUTx, IOUTx (actual) and IOUTx (ideal), expressed in µA, in the linear region of the transfer function. Offset error can be negative or positive and for a current output DAC, it's expressed in µA.

### Offset Error Drift

Offset error drift is a measurement of the change in offset error with a change in temperature. For a current output DAC, the offset error drift is expressed in µA/°C.

### DC Power Supply Rejection Ratio (PSRR)

PSRR indicates how the output of the DAC is affected by DC changes in the supply voltage. For a current output DAC, PSRR is the ratio of the change in IOUTx to a change in AVDDx for a full-scale output of the DAC and it's measured in µA/V.

### AC Power Supply Rejection Ratio (PSRR)

AC PSRR is a measure of the rejection of the DAC output to AC changes in the power supplies applied to the DAC. AC PSRR is measured for a given amplitude and frequency change in power supply voltage, using the minimum recommended decoupling, and for a current output DAC is expressed in decibels or µA/V.

### Output Settling Time

Output settling time is the amount of time it takes for the output of a DAC channel to settle to a specified level for a specified input change and is measured from the falling or rising edge of LDAC.

### Digital-to-Analog Glitch Energy

Digital-to-analog glitch energy is the energy of the impulse injected into the analog output when the input code in the DAC register changes state. For a current output DAC, the energy is normally specified as the area of the current output glitch in nA-sec and is measured when the digital input code is changed by 1 LSB at the major carry transition (such as 0x7FFF to 0x8000 for a 16-bit DAC).

### Digital Feedthrough

Digital feedthrough is a measure of the energy of the impulse injected into the analog output of the DAC from the digital inputs of the DAC but is measured when the DAC output is not updated. For a current output DAC, digital feedthrough is specified as the area of the current output glitch in nA-sec and measured with a full-scale code change on the data bus, that is, from all 0s to all 1s and vice-versa.

### DC Crosstalk

DC crosstalk is the DC change in the output level of one DAC in response to a change in the output of another DAC. For a current output DAC, DC crosstalk is specified as the current output change in µA and is measured with a full-scale output change on one DAC while monitoring another DAC maintained at midscale.

### Digital Crosstalk

Digital crosstalk is the energy of the glitch impulse transferred to the output of one DAC at midscale in response to a full-scale code change (all 0s to all 1s and vice-versa) in the input register of another DAC. For a current output DAC, digital crosstalk is specified as the area of the current output glitch in nA-sec and measured on one DAC channel at a time.

### DAC to DAC Crosstalk

DAC to DAC crosstalk is the energy of the glitch impulse transferred to the output of one DAC due to a digital code change and

## TERMINOLOGY

subsequent analog output change of another DAC. For a current output DAC, DAC to DAC crosstalk is specified as the area of the current output glitch in nA-sec and measured with a full-scale change on one DAC output, writing directly to the DAC register while monitoring the other DAC output kept at midscale.

### Analog Crosstalk

Analog crosstalk is the glitch impulse transferred to the output of one DAC due to a change in the output of another DAC. It is measured by loading one of the input registers with a full-scale code change (all 0s to all 1s and vice-versa), then executing a software LDAC command and monitoring the output of the DAC whose digital code was not changed. The energy of the glitch is expressed in nA-sec for a current output DAC.

### Output Noise Spectral Density (NSD)

Output NSD is a measurement of the internally generated random noise. Random noise is characterized as a spectral density. NSD is measured by loading the DAC to midscale and measuring noise at the output. For a current output DAC, NSD is expressed in nA/√Hz.

### Multiplexer to DAC Crosstalk

The multiplexer to DAC crosstalk is a measure of the impulse injected into the analog output of the DAC channel when the monitor MUX is changed to the DAC channel from a different channel and vice-versa. This parameter is specified as the area of the current output glitch on the DAC channel and is expressed in nA-sec.

## TERMINOLOGY

### REFERENCE

#### Line Regulation

Line regulation refers to the change in output voltage in response to a given change in supply voltage and is expressed in µV/V.

#### Load Regulation

Load regulation refers to the change in output voltage in response to a given change in load current and is expressed in ppm/mA.

#### Voltage Reference ($V_{REF}$) Temperature Coefficient (TC)

$V_{REF}$ TC is a measure of the change in the reference output voltage with a change in the ambient temperature of the device, normalized by the output voltage at 25°C. $V_{REF}$ TC is specified using the box method that defines TC as the maximum change in the reference output over a given temperature range expressed in ppm/°C, as follows:

$$V_{REF}TC = \left(\frac{V_{REF,\ MAX} - V_{REF,\ MIN}}{V_{REF,\ NOM} \times TEMP\_RANGE}\right) \times 10^6 ppm/°C \quad (1)$$

where:

$V_{REF\_MAX}$ is the maximum reference voltage output measured over the full temperature range.

$V_{REF\_MIN}$ is the minimum reference voltage output measured over the full temperature range.

$V_{REF\_NOM}$ is the nominal reference voltage output at ambient temperature (25°C).

*TEMP_RANGE* is the difference between the maximum and minimum operating temperature of the reference.

#### Voltage Reference ($V_{REF}$) Noise Spectral Density (NSD)

$V_{REF}$ NSD is a measurement of the internally generated random noise characterized as a spectral density nV/√Hz.

## THEORY OF OPERATION

### DEVICE INFORMATION

The AD5705R/AD5706R are 4-channel, 12-/16-bit current output digital-to-analog converters (DACs) with an SPI interface. The devices are optimized for sourcing programmable, low-noise output currents with high power efficiency. Integrated features include a 2.5V on-chip voltage reference, load DAC, A/B toggle functionality, and sinusoidal dither. An internal analog multiplexer enables monitoring of key internal nodes, including supply and output voltages, output currents, and die temperature.

### PRECISION REFERENCE

The AD5705R/AD5706R integrate a low-noise, on-chip 2.5V voltage reference with a typical temperature coefficient of 15ppm/°C. By default, this internal reference is powered down and can be enabled by setting the BGAP_HIZ_B bit in the BANDGAP_CONTROL register. When enabled, the buffered 2.5V reference is available at the VREF pin for external system use.

Alternatively, the device supports the use of an external 2.5V reference applied to the VREF pin, which is the default configuration at power-up. For proper operation, it is recommended that the external reference shares the same supply as the AD5705R/AD5706R to ensure that VREF ≤ AVDD during power-up.

To achieve optimal noise performance, consider the following:

- A 0.1µF capacitor should be connected between the NF pin and REFGND, regardless of the reference source.
- When using the internal reference, an additional 1µF capacitor is recommended between VREF and REFGND.

### ANALOG MULTIPLEXER AND DIAGNOSTIC MONITORING

The device includes an integrated diagnostic feature that enables real-time monitoring of key internal and external parameters via the MUX_OUT pin. The following signals can be routed to the MUX_OUT output:

- IOUTx load voltages
- IOUTx output currents
- Supply rail voltages
- VREF (reference voltage)
- Internal die temperature

All monitored signals are represented as unipolar voltages ranging from 0V to VREF. This allows for straightforward interfacing with external ADCs or monitoring systems.

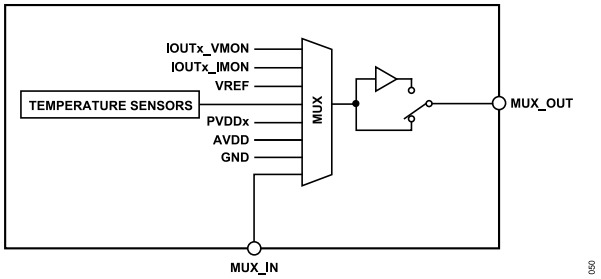



*Figure 52. Analog Multiplexer*

At power-up, the diagnostic monitoring function is disabled, and the MUX_OUT pin is placed in a high-impedance (high-Z) state to prevent unintended signal output.

Complete the following to enable diagnostic monitoring:

- Set the MUX_OUT_EN field in the MUX_OUT_SEL register.
- Use the MUX_PARAM_SEL register to select which internal signal or channel is routed to the MUX_OUT pin.

By default, the GND channel is selected in the MUX block. The available channel selections and their corresponding register field values are listed in Table 9.

*Table 9. Multiplexer Channels*

| MUX_PARAM_SEL Field | MUX_OUT Channel |
|---|---|
| 0 | GND (default) |
| 1 | AVDD supply voltage monitor |
| 2 | VREF voltage monitor |
| 3 | GND |
| 4 | IOUT0 load voltage monitor |
| 5 | IOUT1 load voltage monitor |
| 6 | IOUT2 load voltage monitor |
| 7 | IOUT3 load voltage monitor |
| 8 | IOUT0 output current monitor |
| 9 | IOUT1 output current monitor |
| 10 | IOUT2 output current monitor |
| 11 | IOUT3 output current monitor |
| 12 | PVDD0 supply voltage monitor |
| 13 | PVDD1 supply voltage monitor |
| 14 | PVDD2 supply voltage monitor |
| 15 | PVDD3 supply voltage monitor |
| 16 | IOUT0 temperature sensor monitor |
| 17 | IOUT1 temperature sensor monitor |
| 18 | IOUT2 temperature sensor monitor |
| 19 | IOUT3 temperature sensor monitor |
| 20 to 23 | MUX_IN |
| 24 to 31 | Reserved |

The MUX_OUT pin is internally buffered to support direct interfacing with external ADCs. This buffer can be bypassed by clearing the BUFFER_EN bit (that is, setting it to 0). The internal buffer can be bypassed by setting the BUFFER_EN bit to 0. Note that when

## THEORY OF OPERATION

the internal buffer is disabled, the MUX switch impedance can vary between 1kΩ and 20kΩ. In this configuration, the MUX_OUT pin should only be connected to high-impedance inputs to ensure signal integrity. The available MUX_OUT configurations are summarized in Table 10, and the corresponding full-scale output voltages for each monitored channel are provided in Table 11.

*Table 10. MUX_OUT Configuration*

| MUX_OUT_EN Bit | BUFFER_EN Bit | MUX_OUT Status |
|---|---|---|
| 1'b0 | X | High-Z |
| 1'b1 | 1'b0 | Buffer disabled and bypassed |
| 1'b1 | 1'b1 | Buffer enabled |

*Table 11. Full Scale Output Voltages*

| MUX_OUT Channel | Nominal Full Scale Output (V) |
|---|---|
| Current Monitor Channels | 1.2 |
| Temperature Sensor Channels | 1.6 |
| Other Channels | 2.5 |

### IOUT Output Current Monitoring

When the appropriate MUX channel is selected via the MUX_PARAM_SEL register, the current at the IOUTx pin is represented as a voltage on the MUX_OUT pin. The output voltage range for this representation is 0V to 1.25V.

The output current can be calculated using the following formula:

$$I_{OUT} = \frac{300mA \times V_{MUX\_OUT}}{1.25V} \quad (2)$$

where:

$I_{OUT}$ is the output current at the IOUTx pin.
$V_{MUX\_OUT}$ is the voltage at the MUX_OUT pin.

The voltage output at MUX_OUT maintains the same optimal linearity as the current outputs. However, due to potential slope error (up to ±20% of full-scale range), calibration is recommended for accurate current measurement.

### Two-Point Calibration

For improved accuracy (up to ±1% FSR), a two-point calibration method can be applied, assuming the current output is reasonably linear over the measurement range. The calibrated output current is calculated as:

$$I_{OUT,X} = \frac{I_{OUT,A} - I_{OUT,B}}{DAC\_CODE_A - DAC\_CODE_B} \times \left(DAC\_CODE_X - DAC\_CODE_B\right) + I_{OUT,B} \quad (3)$$

where:

$I_{OUT,A}$, $I_{OUT,B}$ = Measured output currents at respective DAC codes $DAC\_CODE_A$ and $DAC\_CODE_B$.
$I_{OUT,X}$ = Expected output currents at DAC code $DAC\_CODE_X$.

### IOUT Output Voltage Monitoring

By selecting the appropriate MUX channel in the MUX_PARAM_SEL register, a scaled-down representation of the IOUTx pin voltage can be routed to the MUX_OUT pin. The output voltage range for this representation is 0V to VREF, with a typical accuracy of ±10mV.

The conversion from MUX_OUT voltage to the actual IOUTx pin voltage follows the equation:

$$V_{OUT} = 2/3 \times V_{MUX\_OUT} \quad (4)$$

where:

$V_{OUT}$ is the actual voltage at the IOUTx pin.
$V_{MUX\_OUT}$ is the voltage measured at the MUX_OUT pin.

### Internal Die Temperature Monitoring

The device features four integrated temperature sensors, each located near a DAC channel. These sensors operate by sourcing a precision current through diode-connected bipolar junction transistors (BJTs), generating a voltage that correlates with the local junction temperature.

By selecting the appropriate MUX channel in the MUX_PARAM_SEL register, the output of a temperature sensor can be routed to the MUX_OUT pin for external measurement.

The junction temperature $T_J$ is proportional to the voltage $V_{OUT}$ measured at the MUX_OUT pin and can be estimated using the following equation:

$$T_J = 25\ ^\circ C + \frac{V_{OUT} - 1.395V}{-3.8\ mV/^\circ C} \quad (5)$$

where:

$T_J$ is the junction temperature in units of °C.
$V_{OUT}$ is the voltage at the MUX_OUT pin.

### Power Supplies Monitoring

By selecting the appropriate MUX channels AVDD, PVDDx, VREF, and GND in the MUX register, a scaled down representation of the selected rail voltage is multiplexed to the MUX_OUT pin.

$$V_{AVDD} = 1.5 \times V_{MUX\_OUT}$$
$$V_{PVDD} = 1.5 \times V_{MUX\_OUT} \quad (6)$$
$$V_{REF} = V_{MUX\_OUT}$$

It is recommended to bypass the internal buffer for GND channel measurement at the MUX_OUT.

Note that all monitoring equations are valid only when the IOUTx pin is properly loaded and operating within its specified compliance range.

## THEORY OF OPERATION

### MULTIPLEXER INPUT AND DAISY CHAINING

The device includes an external input to the analog multiplexer, accessible via the MUX_IN pin (see Figure 52). This feature enables daisy-chaining of the MUX_OUT paths when multiple devices are connected to the same SPI bus.

By connecting the MUX_OUT of one device to the MUX_IN of the next, a single ADC channel can be shared across multiple AD5705R/AD5706R devices. This configuration simplifies system design and reduces the number of required ADC inputs.

An example of a two-device daisy-chain configuration is shown in Figure 53 for two devices, and the concept can be extended to support more than two devices.

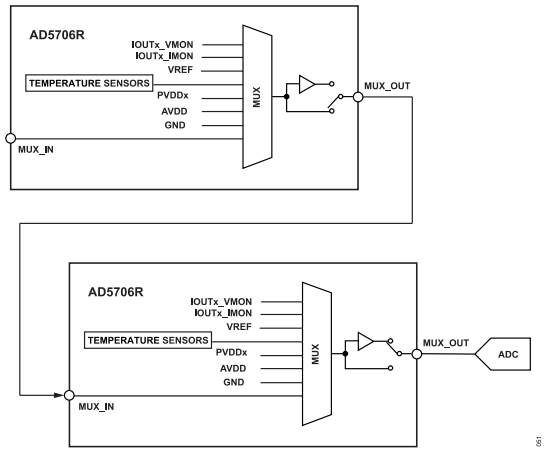



*Figure 53. Input Daisy Chaining for Multiple AD5706R Devices*

### POWER SUPPLIES

The AD5705R/AD5706R devices require the following supply rails:

- AVDD: Positive analog supply for the core analog circuitry.
- IOVDD: Digital interface supply for SPI and control logic.
- PVDDx: Independent positive supply for each DAC output stage (IOUTx).

Each IOUTx channel is powered by its corresponding PVDDx supply, allowing for flexible output stage configuration. To ensure proper operation and maintain output compliance, all PVDDx voltages must be selected such that the IOUTx outputs remain within their specified compliance voltage range.

These PVDDx supplies can be individually set within the range of 1.65V to AVDD.

- If PVDDx is intended to operate at the same voltage as AVDD, it must be directly tied to AVDD.
- If PVDDx is sourced from an independent supply, it must be maintained at the same or lower voltage than AVDD to ensure proper operation and avoid back-biasing internal circuitry.

## DIGITAL INTERFACE

The AD5705R/AD5706R utilize a 4-wire digital interface ($\overline{CS}$, SCK, SDI, and SDO) that is compatible with SPI, QSPI, and MICRO-WIRE standards as well as most digital signal processors (DSPs).

Figure 2 demonstrates the timing for register read and write transactions. These transactions are framed by the $\overline{CS}$ signal. While $\overline{CS}$ is high, SCK edges are ignored, and SDO is in a high impedance state. A falling edge on $\overline{CS}$ activates the digital interface and initiates an SPI frame. Data is shifted MSB first for all SPI transactions.

The SPI interface of the AD5705R/AD5706R supports the transmission mode with CPOL = 0 and CPHA = 0. In this mode, the data on SDI is latched on the rising edges of SCK, while the data is shifted out on SDO on the falling edges of SCK. The data on SDO is valid after the time $t_{SCKL\_SDO}$ from the falling edge of SCK and remains valid for the duration $t_{SDO\_VALID}$ after the next falling edge of SCK (refer to Table 4 and Figure 2).

Communication with the device is divided into two distinct phases.

1. Instruction phase: This phase initiates an action of the device with a 15-bit wide register address.
2. Data phase: In this phase, data is either sent to the device or received from it in response to the instruction phase.

The maximum SCK frequency is 50MHz for writing transactions and 25MHz for reading transactions.

Figure 54 illustrates an SPI transaction phase with a 15-bit address instruction phase. An SPI frame can contain one or more register read and write transactions.

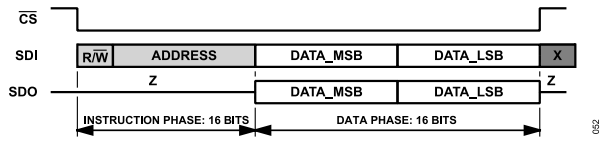



*Figure 54. SPI Transaction Phases*

### INSTRUCTION PHASE

The instruction phase begins immediately after the falling edge of $\overline{CS}$, which initiates the SPI transaction. This phase includes a read/write bit (R/$\overline{W}$) followed by an address word. Setting R/$\overline{W}$ high selects a read instruction (see Figure 56), while setting R/$\overline{W}$ low selects a write instruction (see Figure 55). By default, the address word is 15 bits long (15-bit addressing). The register address sent during the instruction phase serves as the starting address for subsequent read or write operations.

### DATA PHASE

The data phase follows directly after the instruction phase, as illustrated in Figure 55 and Figure 56. During a write instruction, data is clocked into the device via the SDI pin on the rising edges of SCK and is written to the selected register location. Conversely, during a read instruction, data from the selected register location is shifted out via the SDO pin on the falling edges of SCK.

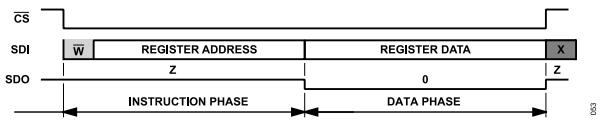



*Figure 55. Basic SPI Write Frame*

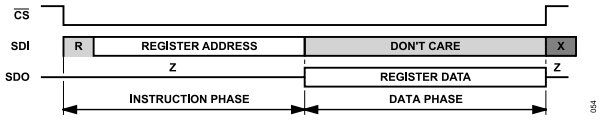



*Figure 56. Basic SPI Read Frame*

The data phase can include the data for a single-byte register, a multibyte register, or multiple registers depending on the selected registers and access modes. For details on how these modes configure the read and write data in the data phase, refer to the sections on Single Instruction Mode, Streaming Mode, and Register Map Address Direction.

The user configuration register contents are updated immediately after the last SCK rising edge, which shifts in the final bit of the register data. For a single-byte register, this occurs on the eighth SCK rising edge of the data phase. For a two-byte register, it happens on the 16th SCK rising edge of the data phase. Refer to the Multibyte Registers section for more information on when multibyte register data is updated.

Data must be written to the configuration registers in full bytes to ensure they are updated. If the data phase of an SPI write transaction does not include the entire byte of data for the register being updated, the register contents are not updated, and the CLOCK_COUNT_ERR bit in the INTERFACE_STATUS_A register is set.

### SPI FRAME SYNCHRONIZATION

The $\overline{CS}$ pin frames data during SPI transactions in both streaming mode and single instruction mode. A falling edge on $\overline{CS}$ initiates an SPI transaction. If $\overline{CS}$ is deasserted (returned high) during an SPI transaction, it terminates part or all of the data transfer.

- If $\overline{CS}$ is deasserted before the instruction phase is complete, the transaction aborts, and the device returns to the ready state.
- If $\overline{CS}$ is deasserted before the first data word is written, the transaction aborts, and the device returns to the ready state.
- If $\overline{CS}$ is deasserted after one or more data words have been written or read, those completed data words are written or read, but any partially written data words are aborted.

Figure 2 illustrates the detailed timing diagram for executing register read and write via the SPI interface. For the specific timing specifications, refer to Table 4.

## DIGITAL INTERFACE

### MULTIBYTE REGISTERS

Some configuration registers in the AD5705R/AD5706R consist of multiple bytes of data stored in adjacent addresses, known as multibyte registers. These registers are all two bytes wide and include the following:

- DAC_INPUT_A_CHn
- FUNC_DAC_INPUT_B_CHn
- MULTI_DAC_INPUT_A_CHn
- DAC_DATA_READBACK_CHn

For the 12-bit parts, the 4 most significant bits (MSBs) are ignored and are set to 0 internally (refer to Table 12).

When writing to a multibyte register, all bytes must be accessed in a single SPI transaction. Consequently, the STRICT_REGISTER_ACCESS bit in the INTERFACE_CONFIG_C register is read only and set to 1. If an SPI write transaction to a multibyte register is attempted on a per byte basis, the register contents are not updated, and the REGISTER_PARTIAL_ACCESS_ERR bit in the INTERFACE_STATUS_A register is set. A write transaction to a multibyte register is completed after the 16th SCK rising edge of the data phase, which shifts in the final bit of the register data.

***Table 12. Multibyte Registers***

| | MSB | | | | | | | | LSB | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Bit** | **15** | **14** | **13** | **12** | **11** | **10** | **9** | **8** | **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| 16 | DATA [15:14] [1] | | DATA [13:8] | | | | | | DATA [7:0] | | | | | | | |
| 12 | 0 | 0 | 0 | 0 | DATA [11:10] [1] | | DATA [9:8] | | DATA [7:0] | | | | | | | |

[1] These bits are don't care in the dither mode when assigned for FUNC_DAC_INPUT_B_CHn registers. Refer to the Dither Function section.

## DIGITAL INTERFACE

### REGISTER MAP ADDRESS DIRECTION

The address direction options control whether the address automatically decrements (address descending) or increments (address ascending) when accessing multiple bytes of data in a single data phase, such as when accessing multibyte registers (see Figure 57 and Figure 58) or multiple registers with streaming mode.

Address direction is selected using the ADDR_ASCENSION bit in the INTERFACE_CONFIG_A register. If ADDR_ASCENSION is set to 0, the address decrements after each byte is accessed. If ADDR_ASCENSION is set to 1, the address increments after each byte is accessed.

The order of the bytes in the SPI frames depends on the ADDR_ASCENSION bit as follows:

- Address descending: The address used must be the address of the most significant byte. The first byte accessed in the data phase is the most significant byte of the multibyte register, with each subsequent byte corresponding to the next lower address.
- Address ascending: The address used must be the address of the least significant byte. The first byte accessed in the data phase is the least significant byte of the multibyte register, with each subsequent byte corresponding to the next higher address.

For example, the DAC_INPUT_A_CH0 register is two bytes long, with the least significant byte at Address 0x60 and the most significant byte at Address 0x61. If the ADDR_ASCENSION bit is set to decrement, the address must be 0x61, and the first byte must be the most significant one. Refer to the Registers section for a complete list of registers.

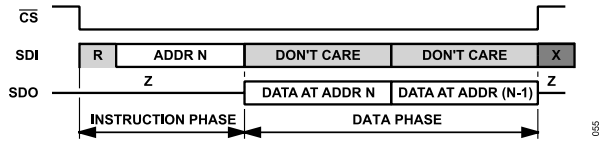



*Figure 57. Multibyte Register Address Descending Read Access*

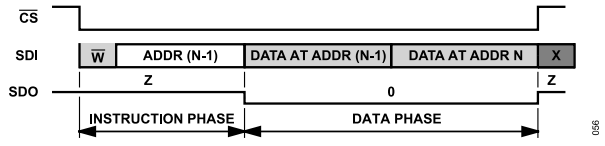



*Figure 58. Multibyte Register Address Ascending Write Access*

When accessing multibyte registers, use descending addresses to shift in most significant byte first. The address direction is set to descending by default.

### STREAMING MODE

When the SINGLE_INST bit in the INTERFACE_CONFIG_B register is set to 0, single instruction mode is disabled, and streaming mode is enabled. In streaming mode, multiple registers with adjacent addresses can be accessed with a single instruction and data phase, allowing efficient access to contiguous memory regions (for example, during initial device configuration). The device is configured in streaming mode by default.

When in streaming mode, each SPI frame consists of a single instruction phase followed by a data phase containing data for multiple registers with adjacent addresses. The digital host specifies a starting register address in the instruction phase, and this address is automatically decremented or incremented (based on the ADDR_ASCENSION bit) after each byte of data is accessed. Consequently, the data phase can be multiple bytes long, with each consecutive byte of read or write data corresponding to the next lower or higher register address (for descending and ascending address direction, respectively).

When writing to a multibyte register in streaming mode with the address descending, the user must start by addressing the most significant byte of the register in the instruction phase and provide data starting from the most significant byte in the data phase. Conversely, when writing to a multibyte register in streaming mode with the address ascending, the user must start by addressing the least significant byte of the register in the instruction phase and provide data starting from the least significant byte in the data phase.

Figure 59 shows the instruction and data phase when using streaming mode with address descending to write to all the DAC input registers. In this example, the starting address is the most significant byte of the DAC_INPUT_A_CH3 register at Address 0x67. To determine the number of data bytes to transfer to consecutive addresses, $\overline{CS}$ must be brought high at the end of the write transaction (in Figure 59, the end of the write transaction occurs after Address 0x60).

When reading from a multibyte register in streaming mode with address descending, read back data starting from the most significant byte. When reading from a multibyte register in streaming mode with address ascending, read back data starting from the least significant byte.

#### Looping in Streaming Mode

The LOOP_COUNT bits in the STREAM_MODE register determine the length of registers addresses to increment or decrement when streaming multiple bytes of data before looping back to the first register address. When streaming mode is active and this register is cleared, register addresses increment or decrement until the end of the address space, then loop to the last or first address in the memory map and continue incrementing or decrementing. Looping allows the digital host to efficiently read from or write to a set of registers repeatedly.

If LOOP_COUNT is set to 0, looping is disabled, and the following occurs:

- Address descending: The address decrements until it reaches Address 0x00. On the subsequent byte access, the address is set to the highest valued byte address available (0x77).

## DIGITAL INTERFACE

- Address ascending: The address increments until it reaches the highest valued byte address available (0x77). On the subsequent byte access, the address is reset to 0x00.

If LOOP_COUNT is set to a nonzero value, looping is enabled, and the value in LOOP_COUNT sets the number of bytes to be accessed in a single data phase before the byte address resets to the one specified in the instruction phase. The loop count value must correspond to the number of actual bytes involved in the loop, which may differ from the number of registers addressed if some are multibyte registers.

Figure 60 shows an example where data is written to the two DAC input registers in streaming mode with the address descending.

The address in the instruction phase is 0x67 and LOOP_COUNT is set to four, but $\overline{CS}$ is held low for six bytes, which means that the DAC_INPUT_A_CH3 register is written at the start and end of the transaction. The user must carefully determine when to end the transaction by deasserting $\overline{CS}$.

Avoid using streaming mode with looping when writing to a range of registers that includes unused and reserved registers.

By default, LOOP_COUNT automatically clears itself after the instruction where it is applied. If the user wants the programmed value to hold indefinitely, the user should first set KEEP_STREAM_LENGTH_VAL bit to 1 before programming the LOOP_COUNT bits.

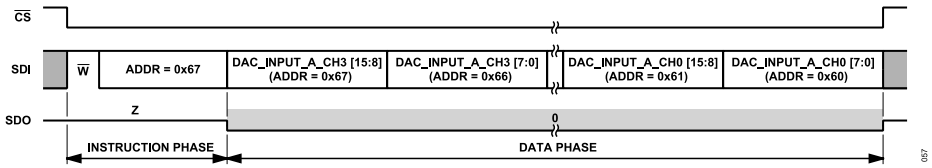



*Figure 59. Streaming Mode Write Register Access with Address Descending (CRC Error Detection Disabled)*

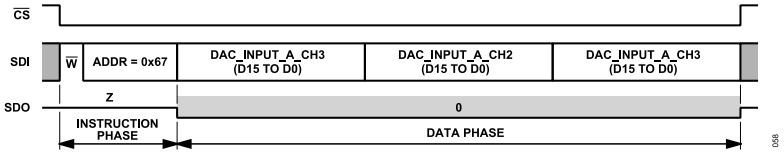



*Figure 60. Streaming Mode Write Register Access with Looping and Address Descending (LOOP_COUNT = 4)*

## DIGITAL INTERFACE

### SINGLE INSTRUCTION MODE

When the SINGLE_INST bit in the INTERFACE_CONFIG_B register is set to 1, streaming mode is disabled, and single instruction mode is enabled. In single instruction mode, the data phase consists of data for a single register, and each data phase must be followed by a new instruction phase even if $\overline{CS}$ remains low.

Single instruction mode allows the digital host to quickly read from and write to registers with non-adjacent addresses in a single SPI frame. In contrast, streaming mode only allows reading or writing to contiguous registers without pulsing $\overline{CS}$ high to initiate a new instruction phase. When multiple registers are accessed in a single synchronization frame in single instruction mode, then the frame width of the data phase matches the byte size of the registers. Thus, the SPI transaction is 24-bit wide for a single byte register, and 32-bit wide for a two-byte register, assuming 15-bit addressing mode.

Figure 61 illustrates an example of an SPI transaction in single instruction mode with the following register accesses:

1. Enables internal reference (BANDGAP_CONTROL register).
2. Enables IOUT0 output (OUT_OPERATING_MODE register).
3. Sets the output range of IOUT0 (OUT_RANGE_CH0 register).
4. Reads the TEMP_WARN_STAT register.

In the Figure 61, BANDGAP_CONTROL is a single byte register, while OUT_OPERATING_MODE, OUT_RANGE_CH0, and TEMP_WARN_STAT are two-byte registers. Therefore, the data phases containing OUT_OPERATING_MODE, OUT_RANGE_CH0, and TEMP_WARN_STAT have two bytes, making the SPI transactions for these data phases 32 bits wide. In contrast, the SPI transaction for the one-byte register BANDGAP_CONTROL is 24 bits wide.

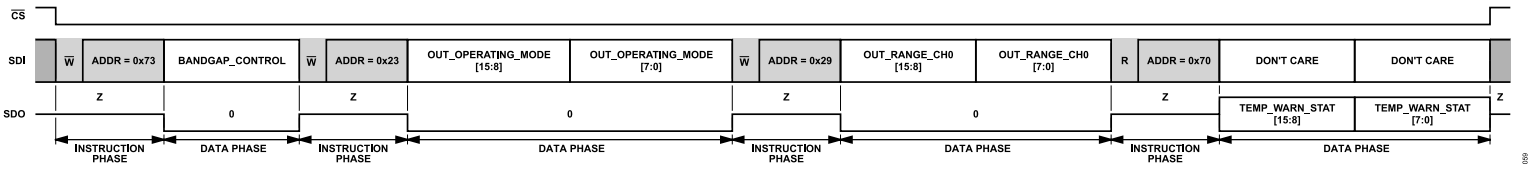



*Figure 61. Single Instruction Mode Register Access Example with Address Descending (CRC Error Detection Disabled)*

## DIGITAL INTERFACE

### DEVICE ADDRESSING

The AD5705R/AD5706R have two address pins, A (1:0), in the WLCSP package. With four unique addresses, it is possible to keep a maximum of four devices on the same SPI bus as shown in Figure 62. Care should be taken to ensure that every device on the same SPI bus has a unique device address on the address pins.

To communicate with any AD5705R/AD5706R, the instruction phase in the SPI frame has three MSB bits, A (14:12), which correspond to the device addresses as indicated in Figure 2 and Table 4. This allows for selecting and writing to a specific device using the corresponding address bits in the instruction phase. An SPI read transaction must address a specific device to read out data on the SDO line.

Registers with Address 0x00 through Address 0x11 can be accessed independent of the device address pins. All other registers require matching address bits in the instruction phase.

By default, the SHORT_INSTRUCTION bit is set to 1'b0 for 15-bit addressing. In case of 7-bit addressing, the device address pins A[1:0] must be set to 00 and in the instruction phase, only the address bits A[7:0] are valid for use in Figure 2.

*Table 13. Device Addressing Truth Table*

| Address Pins, A[1:0] | Device Identity | Address Bits, A[14:12]), in Instruction Phase |
|---|---|---|
| 0 0 | 0 | 0 0 0 |
| 0 1 | 1 | 0 0 1 |
| 1 0 | 2 | 0 1 0 |
| 1 1 | 3 | 0 1 1 |

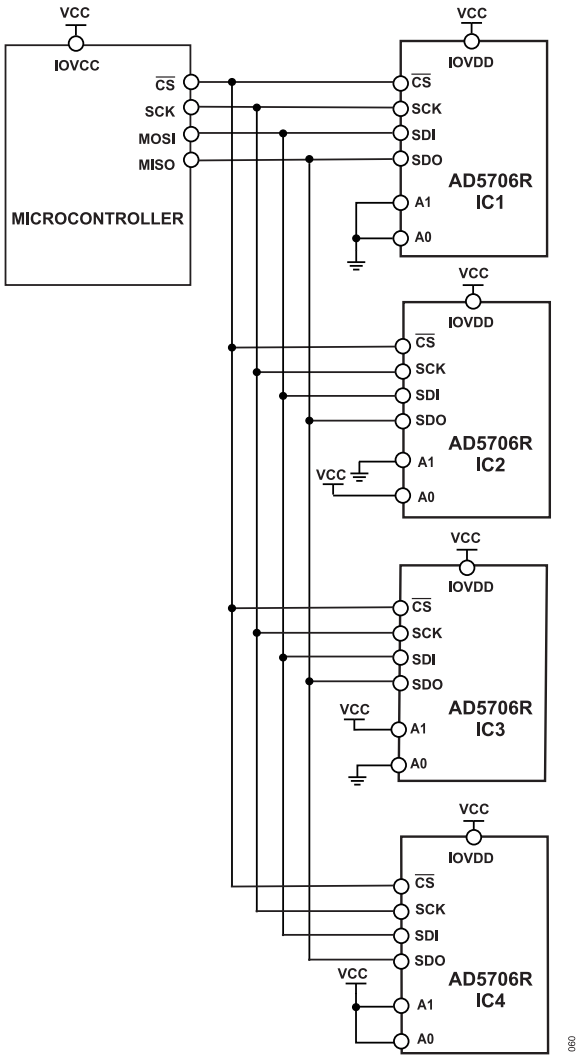



*Figure 62. Communication with Multiple AD5705R/AD5706R Devices on a Single SPI Bus*

## DIGITAL INTERFACE

When multiple AD5705R/AD5706R are connected to the same SPI bus, especially on the SDO line, the capacitive load increases due to the input capacitance of each device connected on the bus and the PCB trace capacitance. This increased capacitance slows down the signal transitions (rise and fall times), which can cause the data on the SDO line to not settle in time before the host samples it on the next clock edge. It is therefore recommended to reduce the SCK frequency to allow more time for the SDO to settle after a transition.

## ERROR DETECTION

### Interface_Status_A Register

The INTERFACE_STATUS_A register contains all the status bits related to the digital interface. These status bits are set based on the actions related to the SPI transactions and remain set until explicitly cleared by writing a 1 to the bit location. Each status bit in the INTERFACE_STATUS_A register is enabled by default.

### Register Does Not Exist Error

When the host attempts to access an invalid register address, the ADDRESS_INVALID_ERR bit is set in the INTERFACE_STATUS_A register. To clear this error, write 1 to this bit.

### Partial Register Access Error

The REGISTER_ PARTIAL_ACCESS_ERR bit in the INTERFACE_STATUS_A register is set when a multibyte register is partially accessed for read or write, meaning that the transaction ends before all bytes of a multibyte register have been accessed. To clear this error, write 1 to the REGISTER_ PARTIAL_ACCESS_ERR bit.

### Write to Read only Register Error

If the host tries to write to a read only register, the WR_TO_RD_ONLY_REG_ERR bit field is asserted in the INTERFACE_STATUS_A register. To clear this error, write 1 to the WR_TO_RD_ONLY_REG_ERR bit.

### Invalid/No CRC Received Error

The CRC_ERR bit in the INTERFACE_STATUS_A register is set when the CRC is enabled and the CRC byte in the SPI transaction is missing or does not match the calculated value. To clear this error, write 1 to this bit. Note that because the CRC is enabled, the SPI transaction must have a valid CRC code to succeed.

### Clock Counting Error

The error reported in the CLOCK_COUNT_ERR bit is produced when the number of SCK cycles does not match the amount required to shift a multiple of eight bits. To clear this error, write 1 to this bit.

### Interface Not Ready Status

The NOT_READY_ERR bit in the INTERFACE_STATUS_A register is a status bit, not an error. This bit can be polled to determine when the device is ready to receive data from the host. This bit is a read or write one to clear (R/W1C) type of bit and can be cleared by writing 1 to it.

### Device Status During Instruction Phase

The device has an internal status register that contains all its major status bits, and it can be appended to SPI frames during transactions. The status bits in the internal status register are mapped to the INTERFACE_STATUS_A and TEMP_WARN_STATUS register bits. When the SEND_STATUS bit in INTERFACE_CONFIG_C register is enabled, the contents of the status register are read out on the SDO pin as shown in Figure 63. It is recommended to use this feature on a single device to avoid issuing separate SPI commands to read the mapped registers. This feature should not be used for multiple devices sharing the same SPI bus.

The TEMP_WARN_INT_EN register can be used to block individual status bits from affecting the status register. All bits in the status register are read only and represent the AND-ed value of the actual status bit and its corresponding mask field, if any. Note that the bits in the INTERFACE_STATUS_A register do not have corresponding mask bits. Actual status bits are R/W1C type bits and can only be cleared if the event that triggered their assertion is no longer true.

The internal status register is 16-bit wide and is described in Table 14. The readout contents of the status register depend on the SHORT_INSTRUCTION bit of the INTERFACE_CONFIG_B register. If the SHORT_INSTRUCTION is set to 1, the Bits[7:0] of the status register are read out on the SPI SDO. If the SHORT_INSTRUCTION is set to 0 (15-bit addressing), the Bits[15:0] of the status register are read out on the SPI SDO.

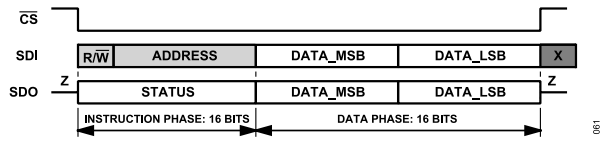



*Figure 63. SPI Transaction Phases with SEND_STATUS Bit Enabled and 15-Bit Addressing Mode*

*Table 14. Internal Status Register Bit Details*

| Bit Position | Status Bit | Register Name |
|---|---|---|
| 0 | TEMP_WARN_FLAG_CH0 | TEMP_WARN_STAT |
| 1 | TEMP_WARN_FLAG_CH1 | TEMP_WARN_STAT |
| 2 | TEMP_WARN_FLAG_CH2 | TEMP_WARN_STAT |
| 3 | TEMP_WARN_FLAG_CH3 | TEMP_WARN_STAT |
| 4 | ORed logic of interface error bits | INTERFACE_STATUS_A |
| 5 to 15 | 0 | Not applicable |

## DIGITAL INTERFACE

### CYCLIC REDUNDANCY CHECK (CRC) ERROR DETECTION

The device features an optional CRC to provide error detection for SPI transactions between the digital host and the AD5705R/AD5706R.

CRC error detection allows the SPI host and its subordinates to detect bit transfer errors with significant reliability. The CRC algorithm involves using a seed value and polynomial division to generate a CRC checksum. Both the host and subordinates calculate the CRC checksum independently to determine the validity of the transferred data.

The device uses the CRC-8 standard with the following polynomial as the divisor:

$$x^8 + x^2 + x + 1 \quad (7)$$

CRC error detection is enabled with the CRC_ENABLE and CRC_ENABLEB bits in the INTERFACE_CONFIG_C register. The value of CRC_ENABLE is only updated if CRC_ENABLEB is set to the CRC_ENABLE inverted value in the same register write instruction. To enable the CRC, CRC_ENABLE must therefore be set to 2'b01 while CRC_ENABLEB is set to 2'b10 in the same write transaction. Note that the CRC error detection must not be enabled when there is more than one device sharing the same SPI bus.

To disable the CRC, CRC_ENABLE must be set to 2'b00 while CRC_ENABLEB is set to 2'b11 in the same write transaction. Writing inverted values to two separate fields reduces the chances of CRC being enabled in error. $\overline{CS}$ must be brought high at the end of the enable/disable write. The first CRC code must be included after the register write/read data, immediately following the register write transaction enabling the CRC. A register write transaction that disables CRC must still include the CRC code on SDI, but the following transaction does not require the CRC code.

Figure 64 and Figure 65 show how a CRC code is appended to the write or read, respectively, for the digital host or the device to validate the data. For register writes, the host must generate the CRC using the calculation described in Equation 7. For register reads, the host must also send the correct CRC byte that is checked by the device. The first byte of data sent contributes to the CRC calculation; therefore, a value of 0x00 is recommended. In the same read transaction, the device provides the CRC code for the digital host to verify.

When accessing multibyte registers with CRC error detection enabled, the CRC code is placed after all bytes of register data.

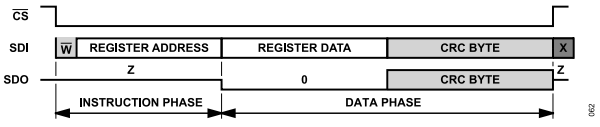



*Figure 64. Basic SPI Write Frame (CRC Error Detection Enabled)*

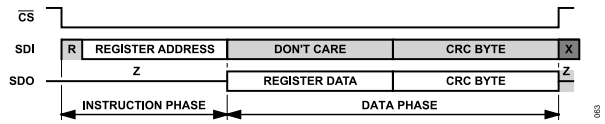



*Figure 65. Basic SPI Read Frame (CRC Error Detection Enabled)*

When CRC error detection is enabled, the device does not update the register contents in response to a register write transaction unless it receives a valid CRC code at the end of the register data on SDI. If the CRC code is invalid or the digital host fails to transmit the CRC code, the device does not update its register contents, and the CRC_ERR flag in the INTERFACE_ STATUS_A register is set. The CRC_ERR flag is R/W1C type of bit, and the correct CRC is required for the write to clear to take effect.

Table 15 shows the seed value used in the CRC code calculation and how it is transmitted for both single instruction mode and streaming mode. When using single instruction mode, every CRC code in a SPI frame uses 0xA5 as the seed value to prevent stuck-at fault conditions for Address 0x0000.

When using streaming mode, the first CRC code in an SPI frame also uses 0xA5 as the seed value, but subsequent CRC codes in the same frame are calculated using the least significant byte of the register address being accessed in the SPI transaction as the seed value depending on the address ascending or descending modes.

Figure 66 shows an example where CRC is enabled in the first SPI transaction, which demonstrates how the CRC codes are provided by the device during a read transaction and by the host during a write transaction. In this example, the device is configured in single instruction mode with the address descending. This example shows the following sequence:

1. Register write of 0x62 to the INTERFACE_CONFIG_C register to enable the CRC.
   - Instruction phase: Address 0x10 (write)
   - Data phase: Value 0x62
   - CRC code: Calculated using the seed value 0xA5 and the data 0x62
2. $\overline{CS}$ is pulsed high.
3. In the address descending mode, the register write of the OUT_RANGE_CH0 register (a multibyte register with the most significant byte at Address 0x29).
   - Instruction phase: Address 0x29 (write)
   - Data phase: Value for OUT_RANGE_CH0 (most significant byte first)
   - CRC code: Calculated using the seed value 0xA5 and the data for OUT_RANGE_CH0
4. Register write of 0x23 to the INTERFACE_CONFIG_C register to disable the CRC while still providing the CRC code.
   - Instruction phase: Address 0x10 (write)
   - Data phase: Value 0x23

## DIGITAL INTERFACE

- CRC code: Calculated using the seed value 0xA5 and the data 0x23

5. $\overline{CS}$ is brought high.

When CRC error detection is enabled, do not use streaming mode, including looping, if the range of registers to be addressed includes unused and reserved registers. See Figure 67 and Figure 68 for examples of write and read SPI transactions in streaming mode, particularly for multibyte registers.

*Table 15. CRC Seed Values*

| | | | Streaming Mode | |
|---|---|---|---|---|
| **SPI Transaction Type** | **Pin** | **Single Instruction Mode** | **First Data Phase** | **Subsequent Data Phases** |
| Read | SDI | CRC seed = 0xA5, instruction phase, write data = 0x00 | CRC seed = 0xA5, instruction phase, write data = 0x00 | Write data = 0x00 to the least significant byte of address, CRC byte sent after each multibyte register |
| | SDO | CRC seed = 0xA5, instruction phase, read data | CRC seed = 0xA5, instruction phase, read data | Read data from the least significant byte of address, CRC byte received after each multibyte register |
| Write | SDI | CRC seed = 0xA5, instruction phase, write data | CRC seed = 0xA5, instruction phase, write data | Write data to the least significant byte of address, CRC byte sent after each multibyte register |
| | SDO | CRC seed = 0xA5, instruction phase, read data = 0x00 | CRC seed = 0xA5, instruction phase, read data = 0x00 | Read data = 0x00, CRC byte received after each multibyte register |

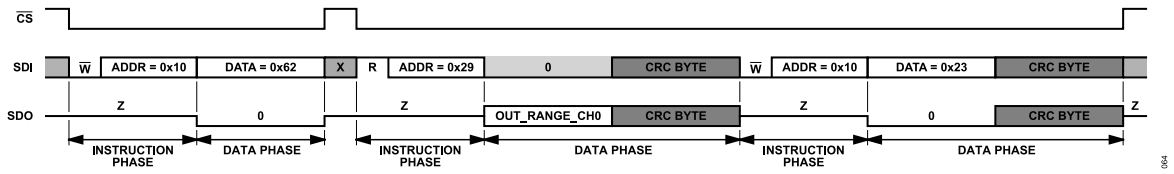



*Figure 66. CRC Code SPI Transactions Example (Address Descending, Single Instruction Mode)*

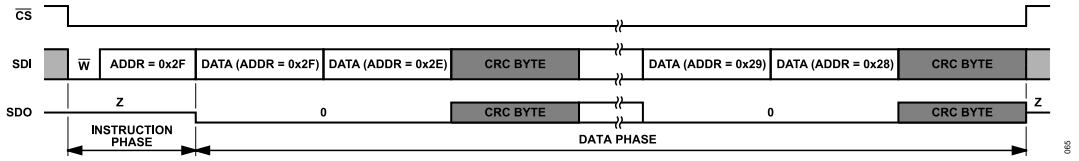



*Figure 67. SPI Write Transaction Example in Streaming Mode for Multibyte Registers with CRC Enabled (Address Descending)*

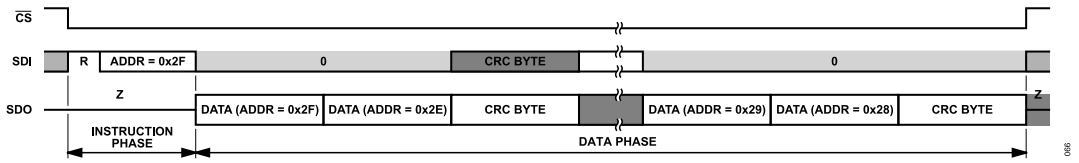



*Figure 68. SPI Read Transaction Example in Streaming Mode for Multibyte Registers with CRC Enabled (Address Descending)*

## DIGITAL INTERFACE

### OVERTEMPERATURE WARNING ALERT

To protect the device from damage due to overtemperature occurrences during operation, the device features an overtemperature warning alert for each IDAC channel. When the internal die temperature reaches approximately 125°C, the TEMP_WARN_FLAG_CHx bit in the TEMP_WARN_STAT register is set to 1. The TEMP_WARN_FLAG_CHx bits in the TEMP_WARN_STAT register are R/W1C type bits and can be cleared by writing 1 to them. To ensure proper data downloads from the internal memory, a reset function must not be performed when the TEMP_WARN_FLAG_CHx bit in the status register is set to 1.

Mask bits in the TEMP_WARN_INT_EN register can be used to mask the corresponding TEMP_WARN_FLAG_CH_x bits in the TEMP_WARN_STAT register. By default, the TEMP_WARN_FLAG_CHx bits are disabled.

These status bits can also be routed out during the instruction phase of any SPI transaction through the SEND_STATUS bit of the INTERFACE_CONFIG_C register. For more details, refer to the Device Status During Instruction Phase section.

There is no automatic shutdown functionality based on the overtemperature warning alert. The system controller must monitor the temperature warning flags through the TEMP_WARN_STAT register and reduce load or disable outputs if required.

### DEVICE RESET

The device offers the following three options for performing a reset:

1. Hardware reset: Resets all user configuration registers to their default values as listed in the Registers section.
2. Software reset via an SPI write: Resets all user configuration registers except for the INTERFACE_CONFIG_A register.
3. Power-on reset (POR): A POR signal is generated when the AVDD supply voltage first exceeds the nominal threshold of 2.4V. The POR resets the user programmable registers if the AVDD supply drops below this threshold.

A hardware reset is initiated by a falling edge on the $\overline{\text{RESET}}$ pin and the device remains in a reset state for as long as the $\overline{\text{RESET}}$ pin is held low. Figure 4 shows the hardware reset timing diagram, with the minimum $\overline{\text{RESET}}$ pulse width, represented by $t_{RESET\_PW}$ in Figure 4 and Table 4.

A software reset is initiated by setting both the SW_RESET and RESET_SW bits in the INTERFACE_CONFIG_A register to 1 in the same data phase. Figure 5 shows the software reset timing diagram.

Whenever any reset event, including a POR event, occurs, the RESET_OCCURRED bit in the DIGITAL_STATUS register is set to 1. This bit is a R/W1C type and can be cleared by writing 1 to it.

The device requires a minimum of 200µs between any reset event and a register read/write transaction as shown in Figure 4 and Figure 5 and represented by $t_{RESET\_DELAY}$ in Table 4. If an SPI transaction is attempted before the device is ready, it may not succeed and the NOT_READY_ERR bit in the INTERFACE_STATUS_A register is set. This is also R/W1C type .

To verify complete initialization, check the NOT_READY_ERR bit in the INTERFACE_STATUS_A register and the POWERUP_COMPLETED bit in the DIGITAL_STATUS register. If any error bit is flagged, perform a device reset.

### SCRATCHPAD AND USER SPARE REGISTERS

The device provides a dedicated SCRATCH_PAD register (Address 0x0A) and four USER_SPARE registers (Address 0x74 to Address 0x77), each 8-bit (one byte) wide. These registers are intended for temporary data storage, allowing flexible handling of intermediate values and reducing reliance on external memory. Users can write to and read from these locations without any device side effects.

### DEVICE IDENTIFICATION

The following registers contain identification information about the AD5705R/AD5706R:

- VENDOR register to identify Analog Devices as the vendor of the device
- CHIP_TYPE register to identify the category of Analog Devices products the device belongs to
- PRODUCT_ID register to be used in conjunction with CHIP_TYPE to identify a device
- CHIP_GRADE register to record the device revision and performance grade
- SPI_REVISION register for the information on the SPI interface revision

The device identifies as follows:

- VENDOR_ID = 0x0456
- CHIP_TYPE = 0x08
- PRODUCT_ID = 0x4130 (AD5706R), 0x4131 (AD5705R)
- CHIP_GRADE = 0x0
- SPI_REVISION = 0x85

## DAC OPERATIONS

The DAC core of each channel of the AD5705R/AD5706R has double buffered registers for the DAC code. Data for each channel can be written directly to the DAC register for an asynchronous DAC update, or to the input registers (DAC_INPUT_A_CHn or FUNC_DAC_INPUT_B_CHn) without changing the DAC outputs. A load DAC command issued to the device via the LDAC pin or the DAC_SW_LDAC bit transfers input register content into the DAC register, updating the DAC output.

Each DAC channel has a user programmable enable bit OUT_OP_MODE_CHn. When the enable bit for any channel is set to 0, the corresponding channel output is shutdown to high-Z state. These enable bits are found in the OUT_OPERATING_MODE register. All DAC channel cores are enabled on power-up.

### DAC REGISTERS (DATA AND INPUT)

Each DAC channel has an associated 12-/16-bit DAC_DATA_READBACK_CHn, DAC_INPUT_A_CHn, and FUNC_DAC_INPUT_B_CHn registers.

- The DAC_DATA_READBACK_CHn is a read-only register that contains the undecoded DAC code that is loaded to the DAC core.
- The DAC_INPUT_A_CHn register is used for storing DAC data that can be loaded to the DAC_DATA_READBACK_CHn register at a later time through the load DAC functions. It is also utilized for the toggle and dither tone generation functionalities.
- Lastly, the FUNC_DAC_INPUT_B_CHn register is used for storing DAC data that is used for the toggle DAC and dither tone generation functionalities.

### IOUT OUTPUT STAGES

The device is a quad-channel current output DAC with selectable output ranges. The full set of current output ranges is only available through SPI programming. Each DAC channel has a programmable current output stage that supports four current ranges: 50mA, 150mA, 200mA, and 300mA. On power-up, the output stages of IOUT0 to IOUT3 are in shutdown mode.

The full-scale current range is selected via the bits OUT_RANGE_CH, on a per channel basis. At power-up, the device initializes with all channel outputs (IOUT0 to IOUT3) in off mode (high-Z mode). The range and code of each channel are then fully programmable through SPI write to registers OUT_RANGE_CHn (Address 0x28 to Address 0x2F) as shown in Table 16 and Figure 69.

*Table 16. Output Ranges*

| OUT_RANGE_CH Bits | Output Current Range |
|---|---|
| 2'b00 | 50mA |
| 2'b01 | 150mA |
| 2'b10 | 200mA |
| 2'b11 | 300mA |

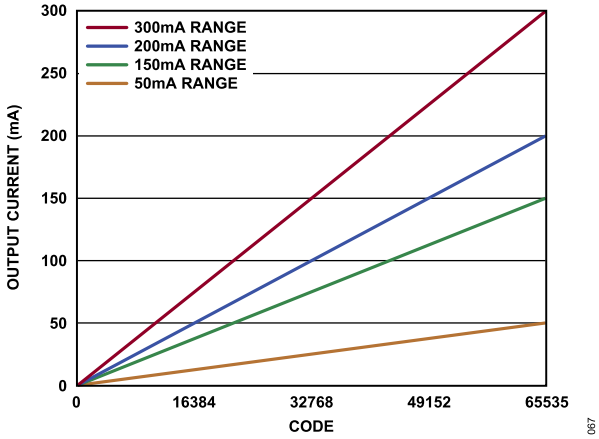



*Figure 69. 16-bit IDAC Transfer Function*

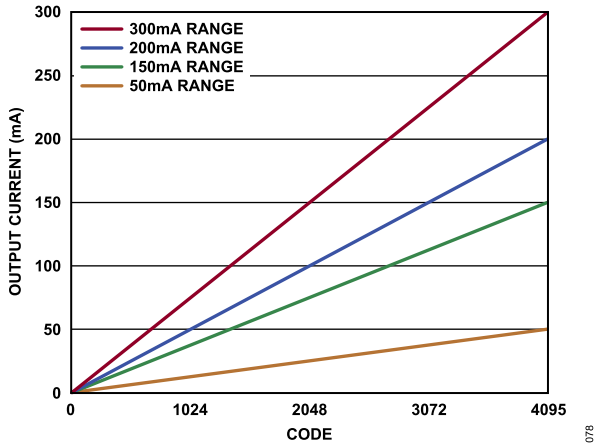



*Figure 70. 12-bit IDAC Transfer Function*

### IOUT in High-Z Mode

On power-up or on device reset, the outputs of each channel are in high-Z (off) mode (by default) and the DAC core is active. Additionally, each channel includes an internal ground switch with an on-resistance of typical 10Ω. The output always transitions to a high-Z state before the ground switch is enabled, ensuring break-before-make operation and preventing transient short-circuit currents. However, the bias circuitry for each IDAC channel remains powered up, and only the output is either in off state or is connected to GND.

The user can control the DAC output stage through the SPI Write or through OUT_EN pin. When changing between output modes on a DAC channel, the output stage of the channel is set to

## DAC OPERATIONS

high-Z to prevent glitches on the output. The output enable functionality for each channel is controlled by specific register bits: OUT_OP_MODE_CHn, OUT_SWITCH_EN_CHn, and HW_SHUTDOWN_EN_CHn. These bits configure the output stage to either high-Z state or connect to ground. In specific use cases, the host controller may read the alarm and other fault status of the AD5705R/AD5706R or other devices on the system and accordingly decides upon status of the DAC output stages through the control of OUT_EN pin or through SPI write. Alternatively, the user can tie the OUT_EN pin to a resistor divider junction between the AVDD and GND, and any fault on the AVDD can disable the DAC output. There are certain bit field names for the output states per channel, as mentioned in the Table 17.

Figure 6 and Figure 7 show the timings associated with the shutdown functionality for a given IOUTx channel.

*Table 17. Output in High-Z Mode and Ground Switch Control*

| Mode | Register Configuration | OUT_EN Pin | DAC_DATA_Readback_CHn Register | DAC Channel GND Switch | DAC Channel Output |
|---|---|---|---|---|---|
| Normal (SW) | OUT_OP_MODE_CHn = 1<br>OUT_SWITCH_EN_CHn = x<br>HW_SHUTDOWN_EN_CHn = 0 (default) | x | Retained | Open | Normal |
| SW Shutdown | OUT_OP_MODE_CHn = 0 (default)<br>OUT_SWITCH_EN_CHn = 0 (default)<br>HW_SHUTDOWN_EN_CHn = x | x | Retained | Open | High-Z |
| SW Shutdown with GND Switch | OUT_OP_MODE_CHn = 0 (default)<br>OUT_SWITCH_EN_CHn = 1<br>HW_SHUTDOWN_EN_CHn = x | x | Retained | Closed | High-Z |
| Normal (HW) | OUT_OP_MODE_CHn = 1<br>OUT_SWITCH_EN_CHn = x<br>HW_SHUTDOWN_EN_CHn = 1 | 1 | Retained | Open | Normal |
| HW Shutdown | OUT_OP_MODE_CHn = 1<br>OUT_SWITCH_EN_CHn = 0 (default)<br>HW_SHUTDOWN_EN_CHn = 1 | 0 (default) | Retained | Open | High-Z |
| HW Shutdown with GND Switch | OUT_OP_MODE_CHn = 1<br>OUT_SWITCH_EN_CHn = 1<br>HW_SHUTDOWN_EN_CHn = 1 | 0 (default) | Retained | Closed | High-Z |

## DAC OPERATIONS

### IOUT Voltage Compliance

When sourcing current, each IOUTx pin must comply with the voltage headroom required by its respective positive supply PVDDx. The minimum headroom voltage when sourcing at full scale is listed in Table 2 as output compliance voltage.

## DAC UPDATE MODES

The AD5705R/AD5706R allows the user to update the DAC channels with different methods. Most DAC update operations require an SPI write to the device. Additionally, it is possible to use LDAC pin functionality to update the DACs, after having loaded the respective DAC data registers in register mode. DAC updates via LDAC pin include the hardware LDAC, the hardware toggle (TGP) and the sinusoidal dither mode (DCK) functions.

Table 24 summarizes the DAC update modes described in this section and succeeding subsections. All DAC update modes described assume the respective output stages are enabled.

### DAC Functionality Control

The Bitfield FUNC_EN_CHn and Bitfield FUNC_MODE_CHn of the FUNC_EN and FUNC_MODE_SEL registers, respectively, are used to control which DAC update features are active at a time for each DAC channel.

When FUNC_EN_CH is 1'b0, only the load DAC function is enabled. The user can select a DAC function through the FUNC_MODE_CH bitfield. However, only by writing 1'b1 to the bitfield FUNC_EN_CH enables the selected DAC function.

Also, when FUNC_EN_CH is 1'b1, writing to the corresponding FUNC_MODE_CH does not update the value of FUNC_MODE_CH, effectively locking the current mode to the function selected. For example, if the channel is configured for toggle mode and needs to be switched to dither tone generation mode, then the user has to first write 1'b0 to FUNC_EN_CH bit, 1'b1 to FUNC_MODE_CH bit, and 1'b1 to FUNC_EN_CH bit sequentially to switch from toggle DAC function to dither tone generation function. Omitting the last write to FUNC_EN_CH bit in the previous sequence retains the toggle DAC function.

The functions, namely, load DAC, toggle DAC, and dither tone generation, are logically exclusive with each other. This means that when one function is active for a particular channel, the other two are not. These functions can be executed either using LDAC pin or with an SPI write command. All DAC functions assume that the output stages of respective DAC channels are enabled. See the definition of FUNC_EN_CH and FUNC_MODE_CH bits in Table 18 below.

*Table 18. DAC Functionality Control bits*

| FUNC_EN_CH bit | FUNC_MODE_CH bit | Function |
|---|---|---|
| 1'b0 (default) | x | Load DAC |
| 1'b1 | 1'b0 (default) | Toggle DAC |
| 1'b1 | 1'b1 | Dither tone generation |

### Global Software and Hardware Triggers

All channels share the same triggers for each of their respective software and hardware modes. For instance, SW LDAC, SW toggle, and SW dither share the trigger of SPI write of 1'b1 to the DAC_SW_LDAC_CHn bitfield of the DAC_SW_LDAC register. This means that the channels are independent of each other and trigger their corresponding SW modes functionalities with just one SPI write. This is the same for the HW triggers: LDAC, TGP, and DCK. This is true in AD5705R/AD5706R since the three mentioned signals share the same pin.

Consequently, a SW trigger cannot trigger a channel on HW mode, and vice-versa.

## LOAD DAC FUNCTION

DAC channels can be updated with the load DAC function. The load DAC functionality can be divided into three modes: asynchronous load, hardware LDAC (through LDAC pin), and software LDAC (through SPI write to the DAC_SW_LDAC register) as illustrated in Figure 71. For the AD5705R/AD5706R, the LDAC pin is shared with the TGP (toggle) and DCK (dither clock) pins.

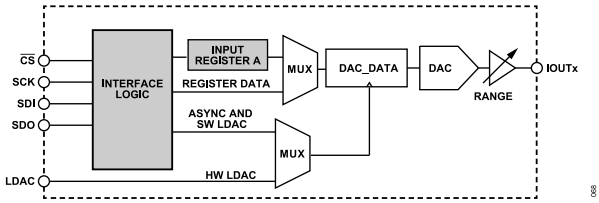



*Figure 71. Load DAC Function*

*Table 19. Load DAC Function Configuration*

| FUNC_EN_CHn bit | FUNC_MODE_CHn bit | LDAC_SYNC_ASYNC_CHn bit | LDAC_HW_SW_CHn bit | DAC Function |
|---|---|---|---|---|
| 1'b1 | x | x | x | None are active |
| 1'b0 (default) | x | 0 (default) | x | Only asynchronous load is active |
| 1'b0 (default) | x | 1 | 0 (default) | Only HW LDAC is active |
| 1'b0 (default) | x | 1 | 1 | Only SW LDAC is active |

The load DAC function is enabled by default or by writing 1'b0 to the Bitfield FUNC_EN_CH assigned to each channel on their corresponding FUNC_EN register. During load DAC mode, the user can choose which mode is active through the Bitfield

## DAC OPERATIONS

LDAC_SYNC_ASYNC_CHn and Bitfield LDAC_HW_SW_CHn on the LDAC_SYNC_ASYNC and LDAC_HW_SW registers, respectively. See Table 19 for reference.

The following subsections explain the different modes of the load DAC function.

### Asynchronous Load DAC

During asynchronous load DAC update, writing to a DAC_INPUT_A_CHn register immediately sends the written data to the corresponding DAC_DATA_READBACK_CHn register of that channel and triggers a DAC update. The DAC output is updated after the last SCK rising edge (16th rising edge of SCK in the data phase) as shown in Figure 8. It is to noted that writing to the FUNC_DAC_INPUT_B_CHn register has no effect on the DAC_DATA_READBACK_CHn register or the DAC output of any respective channel in the asynchronous load DAC mode.

### Software LDAC (SPI Write)

The software LDAC (SW LDAC) mode utilizes a SPI write of 1'b1 to the Bitfield DAC_SW_LDAC_CHn on the DAC_SW_LDAC register as trigger to the load DAC operations. This can be enabled per channel by writing 1'b1 to the LDAC_SYNC_ASYNC_CHn and 1'b1 to the LDAC_HW_SW_CHn bitfields of the corresponding LDAC_SYNC_ASYNC and LDAC_HW_SW registers of each channel, respectively.

Writing 1'b1 to the DAC_SW_LDAC_CHn bitfield of a channel in the DAC_SW_LDAC register transfers the data that is currently written to its respective DAC_INPUT_A_CHn register, given that it is in SW LDAC mode, to their corresponding DAC_DATA_READBACK_CHn register. This triggers a DAC update. See Figure 9.

Writing 1'b1 to DAC_SW_LDAC_CHn of multiple channels also triggers their respective SW LDAC function, individually. It is to be noted that the channels that are on other SW modes (SW toggle and SW dither) also trigger their corresponding SW functions when writing 1'b1 to the Bitfield DAC_SW_LDAC_CHn of the DAC_SW_LDAC register.

The Bitfield DAC_SW_LDAC_CHn auto clears upon write of 1'b1 to the register. Thus, reading back from the DAC_SW_LDAC register only returns 0s.

### Hardware LDAC (LDAC Pin)

The hardware LDAC (HW LDAC) mode utilizes the LDAC pin as trigger for the load DAC operations. This can be enabled per channel by writing 1'b1 to the LDAC_SYNC_ASYNC_CHn and 1'b0 to the LDAC_HW_SW_CHn bitfields of the corresponding LDAC_SYNC_ASYNC and LDAC_HW_SW registers of each channel, respectively. The functionality of the modes follows:

- Idle high/low: While the LDAC pin is held high or low, the data can be written to the DAC_INPUT_A_CHn registers without triggering a DAC update.
- Active edge: The active edge for HW LDAC is selected through the LDAC_HW_EDGE_SEL_CHn bitfields on the LDAC_EDGE_SEL register. This is also true for other HW triggers (toggle, dither). See Table 20 for reference.

*Table 20. HW Active Edge Configuration*

| LDAC_HW_EDGE_SEL_CHn bit | Active Edge |
|---|---|
| 2'b00 (default) | Rising edge |
| 2'b01 | Falling edge |
| 2'b10 | Both edges |
| 2'b11 | No edges |

When the LDAC transitions to an active edge, the data that are currently written in the DAC_INPUT_A_CHn registers for all channels in HW LDAC mode are transferred to their corresponding DAC_DATA_READBACK_CHn registers and trigger a DAC update. See Figure 10. It is to be noted that the channels that are on other HW modes (HW toggle and HW dither) also trigger their corresponding HW functions when triggering on their set active edges.

## TOGGLE DAC FUNCTION

The toggle DAC function is used to generate a square wave at the DAC output. In this mode, each channel has an additional input register, FUNC_DAC_INPUT_B, and the user can switch between these two input registers on where the data is transferred from to the corresponding DAC_DATA_READBACK_CHn register.

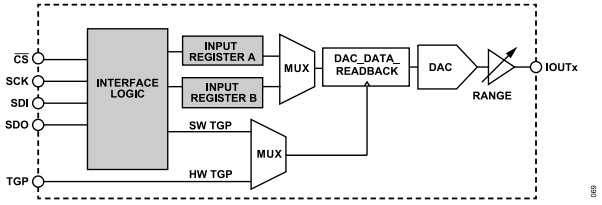



*Figure 72. Toggle DAC Function*

*Table 21. Toggle DAC Function Configuration*

| FUNC_EN_CH_x Bit | FUNC_MODE_CH_x Bit | LDAC_HW_SW_CH_x Bit | Function |
|---|---|---|---|
| 1'b0 (Default) | x | x | None are active |
| 1'b1 | 1'b0 (default) | 1'b0 (default) | Only HW toggle is active |
| 1'b1 | 1'b0 (default) | 1'b1 | Only SW toggle is active |

The toggle DAC function is enabled by writing 1'b0 to the Bitfield FUNC_MODE_CHn on the FUNC_MODE_SEL register, and then, writing 1'b1 to the FUNC_EN_CHn bitfield in the FUNC_EN register. During toggle DAC mode, the user can choose which trigger

## DAC OPERATIONS

is active (HW/SW) by writing to the corresponding LDAC_HW_SW register. See Table 21 for reference.

The multi-DAC update functions are active during toggle DAC Mode.

The following sections detail the functionality for both the HW toggle and SW toggle modes.

### Software Toggle (SPI Write)

The software toggle (SW toggle) function utilizes a SPI write to the DAC_SW_LDAC as trigger to the toggle DAC operations. This can be enabled per channel by writing 1'b1 to the LDAC_HW_SW_CHn bitfield of each channel's corresponding LDAC_HW_SW register. The functionality is detailed in the following sections.

### First Write

After enabling SW toggle mode, writing 1'b1 to a Bitfield DAC_SW_LDAC_CHn of the DAC_SW_LDAC register transfers the data that is currently written to its corresponding DAC_INPUT_A_CHn register, given that it is in SW toggle mode, to its corresponding DAC_DATA_READBACK_CHn registers. This triggers a DAC update. See Figure 11.

Writing 1'b1 to multiple DAC_SW_LDAC_CHn bitfields at the same time is also possible. It is to be noted that the channels that are on other SW modes (SW LDAC and SW dither), also trigger their corresponding SW functions when writing 1'b1 to the Bitfield DAC_SW_LDAC_CHn of the DAC_SW_LDAC register.

The Bitfield DAC_SW_LDAC_CHn auto clears upon write of 1'b1 to the register. Thus, reading back from the DAC_SW_LDAC register only returns 0s.

Entering SW toggle mode from another DAC function (other than HW toggle) resets the sourced data to transfer from DAC_INPUT_A_CHn register for the first SPI write after switching to SW toggle mode. However, switching to SW toggle from HW toggle and vice-versa, does not reset the sourced data to DAC_INPUT_A_CHn register.

### Succeeding Writes

Succeeding writes of 1'b1 to the Bitfield DAC_SW_LDAC_CHn transfer alternately between the DAC_INPUT_A_CHn and FUNC_DAC_INPUT_B_CHn registers (starting with FUNC_DAC_INPUT_B_CHn for the second write). For instance, the second write transfers data from FUNC_DAC_INPUT_B_CHn to the corresponding DAC_DATA_READBACK_CHn register, from DAC_INPUT_A_CHn to the third write, and then from FUNC_DAC_INPUT_B_CHn again to the next.

### Hardware Toggle (TGP Pin)

The hardware toggle (HW toggle) function uses the TGP pin as trigger for the toggle DAC operations. This can be enabled per channel by writing 1'b0 to the LDAC_HW_SW_CHn bitfield of corresponding LDAC_HW_SW register of each channel.

Same as the HW LDAC, the active edge for HW toggle is selected through the LDAC_HW_EDGE_SEL_CHn bitfields on the LDAC_EDGE_SEL_CHn register. See Table 20 again as reference.

The HW toggle functionality is detailed in the following sections.

### Idle High/Low

While the TGP pin is held high or low, the data can be written to the DAC_INPUT_A_CHn registers without triggering a DAC update.

### First Edge

After enabling HW toggle mode, the first active edge of TGP pin transfers the data that is currently written to all DAC_INPUT_A_CHn registers of channels currently in HW toggle mode to their corresponding DAC_DATA_READBACK_CHn registers. This triggers a DAC update. See Figure 12. It is to be noted that the channels that are on other HW modes (HW LDAC and HW dither) also trigger their corresponding HW functions when triggering on their set active edges.

Entering HW toggle from another DAC function (other than SW toggle) resets the sourced data to transfer from DAC_INPUT_A_CHn register for the first TGP trigger after switching to HW toggle. Switching to HW toggle from SW toggle, and vice-versa, does not reset the sourced data to DAC_INPUT_A_CHn register.

### Succeeding Edges

Succeeding active edges transfers alternately between the DAC_INPUT_A_CHn and FUNC_DAC_INPUT_B_CHn registers (starting with FUNC_DAC_INPUT_B_CHn for the second write). For example, the second write transfers data from FUNC_DAC_INPUT_B_CHn to the corresponding DAC_DATA_READBACK_CHn register, from DAC_INPUT_A_CHn to the third write, and then from FUNC_DAC_INPUT_B_CHn again to the next. See Figure 73.

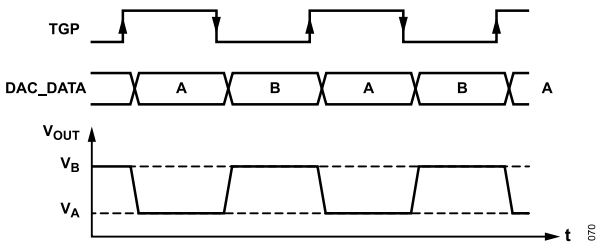



*Figure 73. Toggle Timing with Both Edges Selected as Active Trigger*

### Recommended Flow for Toggle DAC Function

The recommended flow for the toggle DAC function is as follows:

1. Select the toggle DAC function for the desired channels by writing 1'b0 (default) to their respective FUNC_MODE_CHn bitfields on the FUNC_MODE_SEL register. If FUNC_EN_CHn

## DAC OPERATIONS

bit for the corresponding channel is 1'b1, then it must be reset to 1'b0 before enabling the FUNC_MODE_CHn for the toggle DAC function.

2. Enable the toggle DAC function for the desired channels by writing 1'b1 to the corresponding FUNC_EN_CHn bitfields on the FUNC_EN register.
3. Select whether the trigger to be used by the toggle DAC function is via the TGP pin (HW toggle) or via SPI write (SW toggle) by writing to the LDAC_HW_SW_CHn bitfields in the LDAC_HW_SW register. Write 1'b0 for HW toggle or 1'b1 for SW toggle.
4. For HW toggle mode, select which edges are going to be used as active edges for the TGP pin by writing to the corresponding LDAC_HW_EDGE_SEL_CHn bitfields on the LDAC_EDGE_SEL register.
5. Write to the target DAC_INPUT_A_CHn and FUNC_DAC_INPUT_B_CHn registers.
6. For HW toggle mode, apply an edge transition to the TGP pin for the set active edges. For SW toggle mode, write to the corresponding DAC_SW_LDAC_CHn bitfield on the DAC_SW_LDAC register.
7. Repeat Step 6 to functionally output a square wave.
8. Disable the toggle DAC function for the desired channels by writing 1'b0 to the corresponding FUNC_EN_CHn bitfields on the FUNC_EN register.

### DITHER FUNCTION

Dithering is a digital signal processing technique that involves injection of AC noise to the signal path to reduce system nonlinearities. In AD5705R/AD5706R, this adds a small sinusoidal wave approximation to the DAC output.

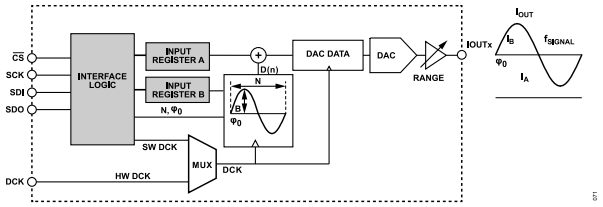



*Figure 74. Dither Tone Function*

A sinusoidal dither signal, D(n), is generated by Equation 8, where N is the signal period, $\phi_0$ is the initial phase offset, and n = 0, 1, 2, … (N-1).

$$D(n) = B \times sin\left(\frac{2\pi n}{N} + \varphi_0\right) \quad (8)$$

where:

n = {0, 1, 2, …, (N-1)}.

N = dither period = {4, 8, 16, 32, 64} samples. Refer to register FUNC_DITHER_PERIOD_CHn in Table 22.

$\phi_0$ = initial phase offset = {0°, 90°, 180°, 270°}. Refer to register FUNC_DITHER_PHASE_CHn in Table 22.

B = peak dither amplitude = [0, $2^{(M-2)}$ - 1]. It is configured through the FUNC_DAC_INPUT_B_CHn register. Refer to Table 12. The 2 MSB bits of data are disregarded on the register FUNC_DAC_INPUT_B_CHn while configuring the dither amplitude.

M = DAC resolution = 12 (for AD5705R) or 16 (for AD5706R).

D(n) = output range = [−B, B].

The signal parameters, N and $\phi_0$, can also be selected for each individual channel through the FUNC_DITHER_PERIOD_CHn and FUNC_DITHER_PHASE_CHn registers, respectively. This allows the precise control of signal frequencies and phase relationships between dithered DAC channels.

The frequency of the sine wave generated corresponds to Equation 9, where $f_{OUT}$ is the dither signal frequency, and $f_{DCK}$ is the dither clock frequency.

$$f_{OUT} = \frac{f_{DCK}}{N} \quad (9)$$

Each channel also utilizes the DAC_INPUT_A_CHn and FUNC_DAC_INPUT_B_CHn registers, where the DAC_INPUT_A_CHn register is used for the offset of the sine wave, while the FUNC_DAC_INPUT_B_CHn register is used for scaling the amplitude of the sine wave.

$$DAC\_OUTPUT\_CODE = A + D(n) \quad (10)$$

where:

A is an M-bit unsigned data obtained from register DAC_INPUT_A_CHn with range [0, $2^M$ −1].

D(n) is represented as a (M-2)-bit signed binary in 2's complement notation with range of only [-$2^{(M-2)}$, $2^{(M-2)}$–1].

M is DAC resolution = 12 (for AD5705R) or 16 (for AD5706R).

At phases 90° and 270°, the DAC_OUTPUT_CODE is same as DAC_INPUT_A_CHn for the first (N/4) samples. It is to be noted that the DAC_OUTPUT_CODE range is bounded to [0, $2^M$ – 1], where M is the DAC resolution.

*Table 22. Dither Tone Generation Module Registers*

| Register | Bitfield | Reset Value | Description |
|---|---|---|---|
| FUNC_DITHER_PHASE_CHn | DITHER_PHASE_CHn | 2'b00 | Configures the initial phase offset ($\phi_0$) in terms of degrees of the sine wave:<br>▶ 2'b00 : 0° (default)<br>▶ 2'b01 : 90° |

## DAC OPERATIONS

*Table 22. Dither Tone Generation Module Registers (Continued)*

| Register | Bitfield | Reset Value | Description |
|---|---|---|---|
| | | | ▶ 2'b10 : 180°<br>▶ 2'b11 : 270° |
| FUNC_DITHER_PERIOD_CHn | DITHER_PERIOD_CHn | 3'b000 | Configures the period (N) in terms of samples of the sine wave:<br>▶ 3'b000 : 4 samples (default)<br>▶ 3'b001 : 8 samples<br>▶ 3'b010 : 16 samples<br>▶ 3'b011 : 32 samples<br>▶ 3'b100 to 3'b111 : 64 samples |

## DAC OPERATIONS

### Example of Dither Operation

As an example, set up channel IDAC0 in dither mode with the following specifications: DC value of 100mA in a 300mA output range, a sinusoidal amplitude of 10mA, sinusoidal frequency of 10kHz, a signal period of N = 64 samples, and use the LDAC/TGP/DCK pin as the dither clock input.

For the AD5706R, the DC value to be written to DAC_INPUT_A_CH0 is d21845 = 0x5555.

The allowable range for dither amplitude is limited to one-fourth of the output current range. FUNC_DAC_INPUT_B_CH0 is loaded with

$$\frac{10mA \quad (Amplitude)}{300mA \quad (Range)/4} \times 2^{14} = d2184 = 0x0888 \qquad (11)$$

For N = 64, set the DCK frequency to 64 × 10kHz = 640kHz. Set the initial phase value to 0 in the register DITHER_PHASE_CH0. See Table 22.

### Dither Tone Generation

The dither mode can be enabled per channel by writing 1'b1 to the respective FUNC_MODE_CHn bitfield in the FUNC_MODE_SEL register, and then, writing 1'b1 to the FUNC_EN_CHn bitfield in the FUNC_EN register.

The dither clock (DCK), that is responsible for sending the dither code output DAC_OUTPUT_CODE to the DAC_DATA_READBACK_CHn register, is controlled/generated via the DCK pin (HW dither) or via SPI writes (SW dither). This option is selected through the LDAC_HW_SW_CHn bitfield on the LDAC_HW_SW register. See the Table 23 for reference. The multi-DAC function is also active during dither mode.

*Table 23. Dither Tone Generation Function Configuration*

| FUNC_EN_CHn Bit | FUNC_MODE_CHn Bit | LDAC_HW_SW_CHn Bit | DAC Function |
|---|---|---|---|
| 1'b0 (default) | x | x | None are active |
| 1'b1 | 1'b1 | 1'b0 (default) | Only HW dither is active |
| 1'b1 | 1'b1 | 1'b1 | Only SW dither is active |

The following sections detail the functionality for both the HW dither and SW dither modes.

### First Trigger

The DAC_OUTPUT_CODE value of the DAC update of the first SW/HW trigger event after entering dither mode is always DAC_INPUT_A_CHn register. This is because A + sin (0) is A (in reference to Equation 10).

Entering either SW dither or HW dither from another DAC function resets the sine wave generated. However, switching to HW dither from SW dither, and vice-versa, does not.

### Software Dither (SPI Write)

Consistent with the other SW triggers, the software dither (SW dither) mode utilizes a SPI Write of 1'b1 to the DAC_SW_LDAC_CHn bitfields in the DAC_SW_LDAC register as triggers to the load DAC operations. This can be enabled per channel by writing 1'b1 to the LDAC_HW_SW_CHn bitfields of the corresponding LDAC_HW_SW register of each selected channel.

Writing 1'b1 to the bitfield DAC_SW_LDAC_CHn of the DAC_SW_LDAC register transfers the data that is currently written to a DAC_INPUT_A_CHn register to their corresponding DAC_DATA_READBACK_CHn register. This triggers a DAC update. See Figure 11. It is to be noted that the channels that are on other SW modes (SW LDAC and SW toggle) also trigger their corresponding SW functions when writing 1'b1 to the bitfield DAC_SW_LDAC_CHn of the DAC_SW_LDAC register.

The bitfield DAC_SW_LDAC_CHn auto clears upon write of 1'b1 to the register. Thus, reading back from the DAC_SW_LDAC register only returns 0s.

### Hardware Dither (DCK Pin)

The hardware dither (HW dither) mode utilizes the DCK pin as trigger for the load DAC operations. This can be enabled per channel by writing 1'b0 to the LDAC_HW_SW_CHn bitfields of each channel's corresponding LDAC_HW_SW register. The mode's functionality is detailed as in the following sections.

### Idle High/Low

While the DCK pin is held high or low, no change in the value of the DAC_OUTPUT_CODE output happens.

### Active Edge

Same as the HW LDAC and HW toggle, the active edge for HW dither is selected through the LDAC_HW_EDGE_SEL_CHn bitfields on the LDAC_EDGE_SEL register. See Table 20 again as reference.

An active edge on the DCK pin updates the DAC_DATA_READBACK_CHn register (with their respective DCK active edge) with the calculated data based on the DAC_INPUT_A_CHn and FUNC_DAC_INPUT_B_CHn registers. It is to be noted that the channels that are on other HW modes (HW LDAC and HW toggle) also trigger their corresponding HW functions when triggering on their set active edges.

### Disabling the Dither Operation

Dither operation can be disabled by resetting the FUNC_EN_CHn bit to 0. After this bit is reset, the dither operation is effectively

## DAC OPERATIONS

disabled and regular DAC codes can then be written to the DAC channel.

### Recommended Flow for Dither Tone Generation Function

The following is the recommended flow for the dither generation:

1. Select the dither tone generation function for the desired channels by writing 1'b1 to their respective FUNC_MODE_CHn bitfields on the FUNC_MODE_SEL register. If FUNC_EN_CHn bit for the corresponding channel is 1'b1, then it must be reset to 1'b0 before enabling the FUNC_MODE_CHn for the dither tone generation function.
2. Enable the dither tone generation function for the desired channels by writing 1'b1 to the corresponding FUNC_EN_CHn bitfields on the FUNC_EN register.
3. Select whether the trigger to be used by the dither tone generation function is via the DCK pin (HW dither) or via SPI writes (SW dither) by writing to the LDAC_HW_SW_CHn bitfields in the LDAC_HW_SW register. Write 1'b0 for HW dither or 1'b1 for SW dither.
4. For HW DCK, select which edges are going to be used as active edges for the DCK pin by writing to the corresponding LDAC_HW_EDGE_SEL_CHn bitfields on the LDAC_EDGE_SEL register.
5. Set the initial phase offset and period in the DITHER_PERIOD and DITHER_PHASE bits on the FUNC_DITHER_PERIOD_CHn and FUNC_DITHER_PHASE_CHn registers, respectively.
6. Write to the target DAC_INPUT_A_CHn registers to set the vertical offset of the sine wave.
7. Write to the target FUNC_DAC_INPUT_B_CHn registers to set the amplitude of the sine wave.
8. For HW dither, apply an edge transition to the DCK pin for the set active edge; or, for SW dither, write 1'b1 to the corresponding DAC_SW_LDAC_CHn bitfield on the DAC_SW_LDAC register.
9. Repeat Step 8 to functionally output a sine wave.
10. Disable the dither generation for the desired channels by writing 1'b0 through their corresponding FUNC_EN_CHn bitfields on the FUNC_EN register.

## DAC OPERATIONS

### Normalized Waveforms

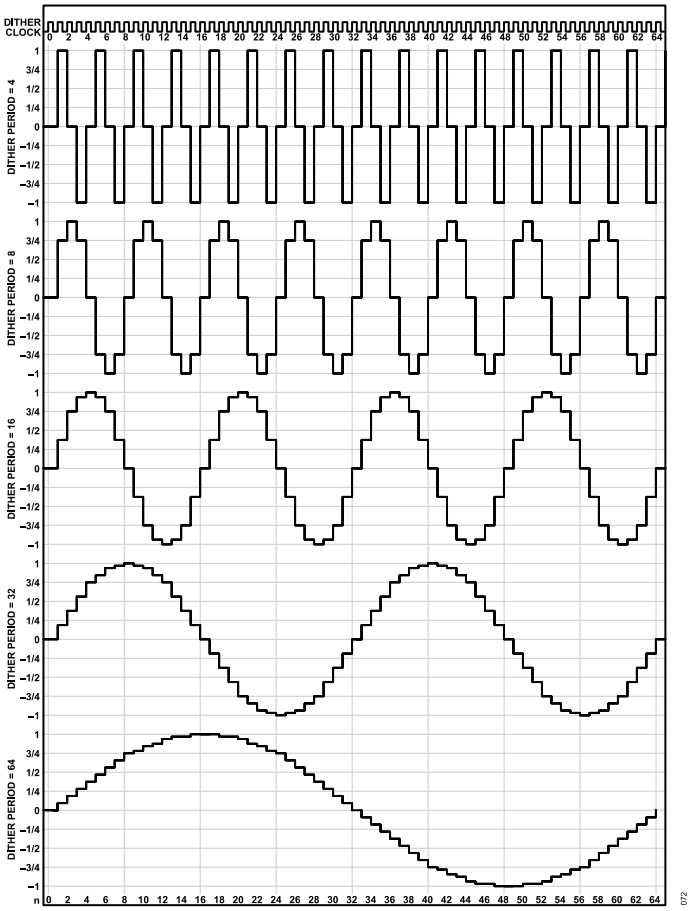



*Figure 75. Normalized Sinusoidal Waveforms for Various Values of Dither Period N*

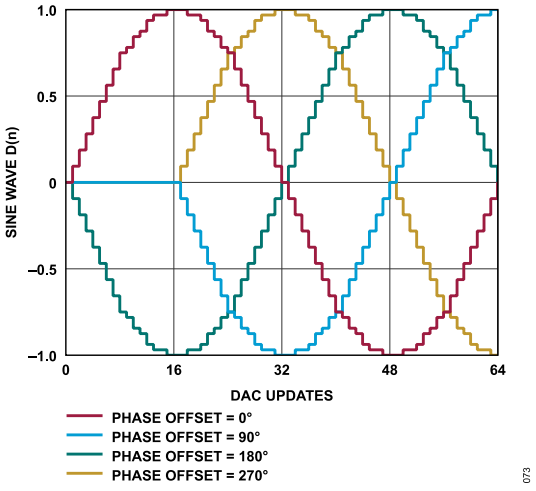



*Figure 76. Normalized Sinusoidal Waveforms for Various Values of Dither Phase Offset*

## DAC OPERATIONS

### MULTI-DAC FUNCTION

The multi-DAC function allows simultaneous writing/updating of all specified DAC channels. A MULTI_DAC_INPUT_A register and a MULTI_DAC_CH_SEL register are available for the multi-DAC function for the DAC_INPUT_A_CHn registers. A bit is assigned per DAC channel in the MULTI_DAC_CH_SEL register. Setting a bit to 1'b1 enables the multi-DAC function for that channel. All channels are disabled (1'b0) by default in the MULTI_DAC_CH_SEL register.

Data written to the MULTI_DAC_INPUT_A register is also written to the DAC_INPUT_A_CHn registers whose MULTI_DAC_CH_SEL bits are asserted. For example, when bits 0 and 1 in the MULTI_DAC_CH_SEL register are set to 1'b1 (channel 0 and channel 1 are selected), and then the data is written to MULTI_DAC_INPUT_A register, the same data is written to the DAC_INPUT_A_CH0 and DAC_INPUT_A_CH1 registers concurrently.

The function of transferring written data from the MULTI_DAC_INPUT_A register to the selected DAC_INPUT_A_CHn registers is active for all modes.

#### Asynchronous Load DAC Mode

Writing to the MULTI_DAC_INPUT_A register transfers the written data to the corresponding DAC_INPUT_A_CHn registers of channels selected in the MULTI_DAC_CH_SEL register. Additionally, the selected channels that are in asynchronous load DAC mode also transfers the written data to their respective DAC_DATA_READBACK_CHn registers, and this triggers a DAC update.

The functionality and timing is same as referenced in the Asynchronous Load DAC section.

#### Software LDAC Mode

Writing 1'b1 to the Bitfield MULTI_DAC_SW_LDAC of the MULTI_DAC_SW_LDAC register triggers a SW LDAC for all channels that are in SW LDAC mode and are selected in the MULTI_DAC_CH_SEL register. This transfers the contents of the MULTI_DAC_INPUT_A register to the respective DAC_INPUT_A_CHn registers and their corresponding DAC_DATA_READBACK_CHn registers and triggers a DAC update for each selected channel.

The sequence is as follows:

1. For all desired DAC channels to be triggered with a SW LDAC update, set the corresponding DAC function mode to any SW modes (SW LDAC, SW toggle, and SW dither).
2. Write 1'b1 to all the bits assigned to the desired channels in the MULTI_DAC_CH_SEL register.
3. Set the value of the DAC_INPUT_A_CHn registers of selected channels by writing to the MULTI_DAC_INPUT_A register. Note that if there are channels that are in asynchronous load DAC mode and are also selected in the MULTI_DAC_CH_SEL register, this triggers a DAC update.
4. Write 1'b1 to the bitfield MULTI_DAC_SW_LDAC of the MULTI_DAC_SW_LDAC to trigger a corresponding SW LDAC to the selected channels depending on their SW modes as follows:
   a. SW LDAC: transfer the content of the DAC_INPUT_A_CHn registers to their corresponding DAC_DATA_READBACK_CHn registers and triggers a DAC update.
   b. SW toggle: toggle between transferring data from DAC_INPUT_A_CHn and FUNC_DAC_INPUT_B_CHn registers to the corresponding DAC_DATA_READBACK_CHn register and triggers a DAC update.
   c. SW dither: output a step of a sine wave and trigger a DAC update.
   d. Other modes: no updates to the DAC_DATA_READBACK_CHn registers and the DAC outputs.

The Bitfield MULTI_DAC_SW_LDAC_CHn auto clears upon write of 1'b1 to the register. Thus, reading back from the MULTI_DAC_SW_LDAC register only returns 0s.

The functionality and timing is same as referenced in the Software LDAC (SPI Write) section.

#### Hardware LDAC Mode

When the HW LDAC pin transitions to an active edge (selected in the LDAC_EDGE_SEL_CHn register), a HW LDAC is triggered for all the channels that are in HW LDAC mode and are selected in the MULTI_DAC_CH_SEL register. This transfers the contents of the MULTI_DAC_INPUT_A register to the respective DAC_INPUT_A_CHn registers and their corresponding DAC_DATA_READBACK_CHn registers and triggers a DAC update for each selected channel.

The sequence is as follows:

1. Ensure that there is no active HW DAC function modes. Set the LDAC pin in an inactive position to the intended active edge. Set the active edge accordingly in the LDAC_EDGE_SEL_CHn registers of the desired channels.
2. For all desired DAC channels to be triggered with a HW LDAC update, set the corresponding DAC function mode to any HW modes (HW LDAC, HW toggle, HW dither).
3. Write 1'b1 to all the bits assigned to the desired channels in the MULTI_DAC_CH_SEL register.
4. Set the value of the DAC_INPUT_A_CHn registers of the selected channels by writing to the MULTI_DAC_INPUT_A register. Note that if there are channels that are in asynchronous load DAC mode and are also selected in the MULTI_DAC_CH_SEL register, this triggers a DAC update.
5. Transition the LDAC pin to an active edge in order to trigger a corresponding HW LDAC to the selected channels depending on their HW modes as follows:

## DAC OPERATIONS

a. HW LDAC: transfer the content of the DAC_INPUT_A_CHn register to their corresponding DAC_DATA_READBACK_CHn register and trigger a DAC update.
b. HW toggle: toggle between transferring data from DAC_INPUT_A_CHn and FUNC_DAC_INPUT_B_CHn registers to the corresponding DAC_DATA_READBACK_CHn register and trigger a DAC update.
c. HW DCK: output a step of a sine wave and trigger a DAC update.
d. Other modes: no updates to the DAC_DATA_READBACK_CHn registers and the DAC outputs.

The functionality and timing is same as referenced in the Hardware LDAC (LDAC Pin) section.

### SUMMARY OF DAC MODES

*Table 24. DAC Modes in Terms of DAC Configuration Bits*

| Mode | FUNC_EN_CHx Bit | FUNC_MODE_CHx Bit | LDAC_SYNC_ASYNC_CHx Bit | LDAC_HW_SW_CHx Bit |
|---|---|---|---|---|
| Asynchronous Load | 0 (default) | x | 0 (default) | x |
| SW LDAC | 0 (default) | x | 1 | 1 |
| HW LDAC | 0 (default) | x | 1 | 0 (default) |
| SW Toggle | 1 | 0 (default) | x | 1 |
| HW Toggle | 1 | 0 (default) | x | 0 (default) |
| SW Dither | 1 | 1 | x | 1 |
| HW Dither | 1 | 1 | x | 0 (default) |

## APPLICATIONS INFORMATION

### RECOMMENDED EXTERNAL COMPONENTS

The target application module solution is highly space constrained, requiring the minimization of external component count and size while adhering to functional and specification requirements.

Typically, capacitor values are considered with a ±5% tolerance unless otherwise specified.

*Table 25. External Passive Components*

| Pin Name | Passive Components Required |
|---|---|
| $\overline{RESET}$ | 10kΩ to IOVDD |
| NF | 0.1µF capacitor to AGND |
| VREF | 1µF capacitor to AGND |
| AVDD | 1µF capacitor per pin to AGND |
| PVDDx | 1µF capacitor per pin to AGND |
| IOVDD | 1µF capacitor to AGND |

### POWER SUPPLY INITIALIZATION

For a more robust, user-controlled power-on, it is recommended to hold the active-low RESET signal low until all relevant power supplies and signals are powered up and settled. Additionally, keep the digital lines $\overline{CS}$, SCK, SDI, and SDO in high impedance or low during the power-on sequence.

The sequence described in the Device Reset section requires a specific duration identified as $t_{RESET\_DELAY}$ in Table 4. During $t_{RESET\_DELAY}$, avoid using the digital interface to communicate with the device, including toggling any digital asynchronous pins.

If the digital host attempts an SPI transaction before the device is ready, the transaction is invalid and the NOT_READY_ERR bit in the INTERFACE_STATUS_A register is set. The NOT_READY_ERR bit is a R/W1C type of bit. Check the NOT_READY_ ERR bit in the INTERFACE_STATUS_A register and the POWERUP_COMPLETED bit in the DIGITAL_STATUS register to verify initialization. If any error bit is flagged, perform a device reset.

After the device initializes, the digital interface can be accessed to configure the device, including selecting the reference scheme according to the application. Regardless of the voltage reference scheme used, it is recommended to let the DAC voltage reference settle after configuring the device to ensure it achieves its specifications.

### COMBINING CHANNELS TO INCREASE CURRENT RANGE

The maximum current that can be sourced from IDACx is 300mA. It is possible to increase the current source capability by directly connecting the two or more channels together. For example combining IOUT0 with IOUT1 can create a full scale output current of 400mA as shown in Figure 77. When combining channels, ensure the following:

- The output compliance voltage stays within the range specified in Table 2.
- The output voltage stays within the absolute maximum ratings specified in Table 6.

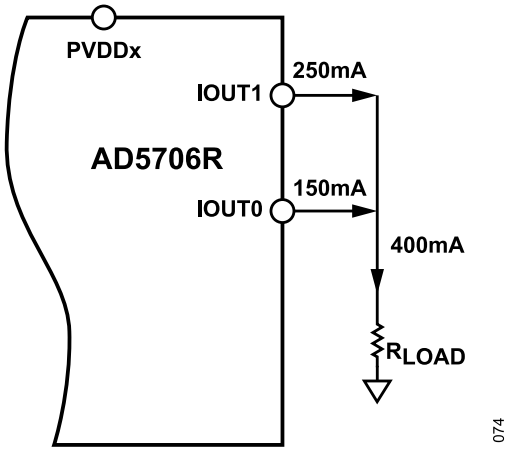



*Figure 77. Increasing the Current Range by Summing Channels*

### COMBINING CHANNELS TO REDUCE OUTPUT NOISE

Another benefit associated with combining channels is noise reduction, especially at higher frequencies. High frequency noise is uncorrelated between channels, so it can be estimated in a root-sum-square fashion when tying the channels together.

For example, the specified noise spectral density (NSD) for one channel at 200mA is 4.49nA/√Hz at 10kHz. If two channels are tied together generating 100mA each (200mA total) with an NSD of 2.25nA/√Hz at 10kHz per channel at 100mA, the estimated NSD at 10kHz of both channels combined (200mA) can be calculated using the root-sum-square approach as shown in the following equation.

$$NSD_{IOUT} \approx \sqrt{NSD_{IOUT0}^2 + NSD_{IOUT1}^2} = \sqrt{2.25^2 + 2.25^2} \approx 3.18\text{nA}/\sqrt{\text{Hz}} \quad (12)$$

Thus, the NSD is improved from 4.49nA/√Hz to an estimation of 3.18nA/√Hz by combining both channels.

### MICROPROCESSOR INTERFACING

Microprocessor interfacing to the AD5705R/AD5706R is via a serial bus that uses a standard protocol compatible with DSPs and microcontrollers. The communication channel requires a 4-wire serial interface consisting of a clock signal (SCLK), a data input signal (SDI/MOSI), a data output signal (SDO/MISO), and a synchronization signal (SS).

The SPI interface of the AD5705R/AD5706R is designed for easy connection to industry-standard DSPs and microcontrollers. Figure 78 shows the AD5706R connected to the ADuCM320. The ADuCM320 has an integrated SPI port that can be connected directly to the SPI pins of the AD5705R/AD5706R.

## APPLICATIONS INFORMATION

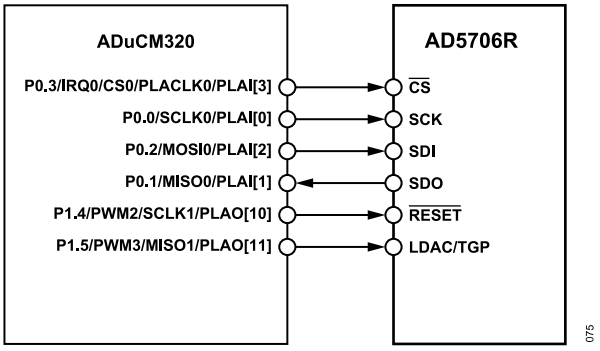



*Figure 78. ADuCM320 SPI Interface*

## LAYOUT AND ASSEMBLY GUIDELINES

Take careful consideration of the power supply and ground return layout to ensure the rated performance. Ensure that currents flowing from the ground end of each decoupling capacitor to its associated ground pin share as little track as possible with other ground currents on the PCB. Design the PCB so that the device lies on the analog plane.

Place the decoupling capacitors as close to the package as possible (ideally directly against the device).

Ensure that the power supply line has as large a trace as possible to provide a low impedance path and reduce glitch effects on the supply line. Shield clocks and other fast switching digital signals from other parts of the PCB by using a separate ground. Avoid crossover of digital and analog signals if possible. When traces cross on opposite sides of the board, ensure that they run at right angles to each other to reduce feedthrough effects through the PCB.

Because the AD5705R/AD5706R can dissipate a large amount of power, it is recommended to provide some heat sinking capability to allow power to dissipate easily. For the WLCSP package, heat is transferred through the solder balls to the PCB. Thermal impedance is dependent on PCB construction. More copper layers and ground vias enable heat to be removed more effectively.

The PCB level reliability of the device is directly linked to the PCB type and design used. Using a PCB material that matches the coefficient of thermal expansion (CTE) of the silicon (for example, ceramic) provides the optimal mechanical performance. For organic material PCBs (for example, FR4) where the CTE is different from that of the silicon, the use of underfill can increase the mechanical performance. For organic PCB thickness >0.8mm, consider using underfill. Particular attention must be given to the underfill material selection to match the material properties with the application use conditions.

Consider using low alpha material in the system assembly to reduce the soft error rate (SER).

The AN-617 Application Note provides information on PCB layout and assembly for the WLCSP.

## THERMAL CONSIDERATIONS

The device has an absolute maximum junction temperature of 150°C (see Table 6) and a maximum operating junction temperature ($T_{J\_MAX}$) of 125°C (see Table 2). To meet the specified performance, the AD5705R/AD5706R must be operated at a junction temperature no greater than 125°C. The junction temperature is directly affected by the power dissipated across the AD5705R/AD5706R and the ambient temperature.

Most of the power dissipation of the AD5705R/AD5706R comes from the IOUT channels. Table 2 specifies the output current ranges for each IOUT channel and the maximum power supply voltages. It is important to understand the effects of power dissipation on the package and the effects the package has on the junction temperature. The part is packaged in a 36-ball 2.55mm × 2.55mm WLCSP. The thermal resistance values are specified in Table 7. Table 26 provides examples of the maximum allowed power dissipation, the maximum allowed ambient temperature, and the maximum board temperature under certain specified conditions.

*Table 26. Thermal Considerations for AD5705R/AD5706R WLCSP Package*

| Parameter | Description |
|---|---|
| Maximum Power Dissipation | When the operating ambient temperature is known ($T_A$), $\theta_{JA}$ can be used to estimate the maximum allowed power dissipation ($P_{DISS\_MAX}$) starting from the maximum operating junction temperature ($T_{J\_MAX}$). For example, if $T_A$ = 85°C,<br>$P_{DISS\_MAX} = \frac{T_{J\_MAX} - T_A}{\theta_{JA}} = \frac{125°C - 85°C}{42.26°C/W} = 0.95W$ (13)<br>When the board temperature near the DUT is known ($T_B$ measured using a temperature sensor near the DUT on the same board), $\Psi_{JB}$ can be used to better estimate the maximum allowed power dissipation ($P_{DISS\_MAX}$). For example, if $T_B$ = 95°C,<br>$P_{DISS\_MAX} = \frac{T_{J\_MAX} - T_B}{\Psi_{JB}} = \frac{125°C - 95°C}{8.1°C/W} = 3.7W$ (14)<br>Using the preceding equation, calculate the AD5705R/AD5706R $P_{DISS}$ as follows, and confirm $P_{DISS}$ is lower than the allowed $P_{DISS\_MAX}$:<br>Power dissipation calculation example:<br>$R_{LOAD}$ = 8.5Ω per channel,<br>AVDD = PVDD = IOVDD = 3.3V |

## APPLICATIONS INFORMATION

*Table 26. Thermal Considerations for AD5705R/AD5706R WLCSP Package (Continued)*

| Parameter | Description |
|---|---|
| | AD5705R/AD5706R quiescent power dissipation = 100mW<br>IOUT0 current: 300mA, power dissipation = 225mW<br>IOUT1 current: 250mA, power dissipation = 294mW<br>IOUT2, IOUT3 current: 275mA, power dissipation = 264mW each<br>$P_{DISS}$ = 100mW (quiescent) + 1.047W (active)= 1.147W (Total) |
| Maximum Ambient Temperature | Maximum allowed ambient temperature, when dissipating 1.147W across the AD5705R/AD5706R is<br>$T_{A\_MAX} = T_{J\_MAX} - \left(P_{DISS} \times \theta_{JA}\right) = 125°C - \left(1.147 \;\; W \times 42.26°C/W\right) \cong 76°C$ (15) |
| Maximum Board Temperature | Maximum allowed board temperature, when dissipating 1.147W across the AD5705R/AD5706R is<br>$T_{A\_MAX} = T_{J\_MAX} - \left(P_{DISS} \times \Psi_{JB}\right) = 125 \;\; °C - \left(1.147 \;\; W \times 8.1 \;\; °C/W\right) \cong 116 \;\; °C$ (16) |

## REGISTERS

The AD5705R/AD5706R has programmable user configuration registers that are used to configure the device. The Register Summary section contains the complete list of the user configuration registers.

The access column specifies whether the register comprises only of read only bits (R) or a mix of read only and read/write bits (R/W). Read only bits cannot be overwritten by an SPI write transaction, whereas read/write bits can. Attempting to write to a register that contains only read only bits sets the WR_TO_RD_ONLY_REG_ERR flag in the INTERFACE_STATUS_A register. Attempting to access an invalid register address sets the ADDR_INVALID_ERR flag in the INTERFACE_STATUS_A register. The error flags in the INTERFACE_STATUS_A register are also read or write 1 to clear (R/W1C) and are only reset when an SPI write transaction writes a 1 to their location.

The Register Summary section and the Register Bitwise Summary section show the size of the register and its bitfields. See the Multibyte Registers section for a detailed description of how multibyte registers can be accessed.

### REGISTER SUMMARY

*Table 27. Register Summary*

| Address | Name | Description | Reset | Access |
|---|---|---|---|---|
| 0x00 | INTERFACE_CONFIG_A | Interface Configuration A. | 0x10 | R/W |
| 0x01 | INTERFACE_CONFIG_B | Interface Configuration B. | 0x00 | R/W |
| 0x02 | DEVICE_CONFIG | Device Configuration. | 0x00 | R |
| 0x03 | CHIP_TYPE | Chip Type. | 0x08 | R |
| 0x04 | PRODUCT_ID_L | Product ID Low. | 0x30 or 0x31 | R |
| 0x05 | PRODUCT_ID_H | Product ID High. | 0x41 | R |
| 0x06 | CHIP_GRADE | Chip Grade. | 0x00 | R |
| 0x0A | SCRATCH_PAD | Scratch Pad. | 0x00 | R/W |
| 0x0B | SPI_REVISION | SPI Revision. | 0x85 | R |
| 0x0C | VENDOR_L | Vendor ID Low. | 0x56 | R |
| 0x0D | VENDOR_H | Vendor ID High. | 0x04 | R |
| 0x0E | STREAM_MODE | Stream Mode. | 0x00 | R/W |
| 0x0F | TRANSFER_CONFIG | Transfer Configuration. | 0x00 | R/W |
| 0x10 | INTERFACE_CONFIG_C | Interface Configuration C. | 0x2F | R/W |
| 0x11 | INTERFACE_STATUS_A | Interface Status A. | 0x00 | R/W |
| 0x14 | MULTI_DAC_CH_SEL | Multi-DAC Channel Select. | 0x0000 | R/W |
| 0x16 | LDAC_SYNC_ASYNC | Sync-Async Load. | 0x0000 | R/W |
| 0x18 | LDAC_HW_SW | Hardware-Software Trigger. | 0x0000 | R/W |
| 0x1A to 0x20 by 2 | LDAC_EDGE_SEL_CHn | LDAC Edge Select. | 0x0000 | R/W |
| 0x22 | OUT_OPERATING_MODE | Output Operating Mode. | 0x0000 | R/W |
| 0x24 | OUT_SWITCH_EN | Output Switch Enable. | 0x0000 | R/W |
| 0x26 | OUT_EN | OUT_EN Pin Enable. | 0x0000 | R/W |
| 0x28 to 0x2E by 2 | OUT_RANGE_CHn | Output Range. | 0x0000 | R/W |
| 0x30 | FUNC_EN | Function Enable. | 0x0000 | R/W |
| 0x32 to 0x38 by 2 | FUNC_MODE_SEL_CHn | Function Mode Selector. | 0x0000 | R/W |
| 0x3A to 0x40 by 2 | FUNC_DAC_INPUT_B_CHn | DAC Input B. | 0x0000 | R/W |
| 0x42 to 0x48 by 2 | FUNC_DITHER_PERIOD_CHn | Dither Settings. | 0x0000 | R/W |
| 0x4A to 0x50 by 2 | FUNC_DITHER_PHASE_CHn | Dither Settings. | 0x0000 | R/W |
| 0x54 | MUX_OUT_SEL | MUX Out Select. | 0x0000 | R/W |
| 0x56 | MUX_OUT_CONTROL | MUX Out Control. | 0x0001 | R/W |
| 0x58 | TEMP_WARN_INT_EN | Temperature Warning Interrupt Enabled. | 0x0000 | R/W |

## REGISTERS

*Table 27. Register Summary (Continued)*

| Address | Name | Description | Reset | Access |
|---|---|---|---|---|
| 0x5A | MULTI_DAC_SW_LDAC | Software LDAC for Multiple DAC Input. | 0x0000 | R/W |
| 0x5C | MULTI_DAC_INPUT_A | Multiple DAC Input. | 0x0000 | R/W |
| 0x5E | DAC_SW_LDAC | Software LDAC for DAC Input. | 0x0000 | R/W |
| 0x60 to 0x66 by 2 | DAC_INPUT_A_CHn | DAC Input. | 0x0000 | R/W |
| 0x68 to 0x6E by 2 | DAC_DATA_READBACK_CHn | DAC Data Readback. | 0x0000 | R |
| 0x70 | TEMP_WARN_STAT | Temperature Warning Flags Status. | 0x0000 | R/W |
| 0x72 | DIGITAL_STATUS | Digital Status. | 0x01 | R/W |
| 0x73 | BANDGAP_CONTROL | Bandgap Control. | 0x00 | R/W |
| 0x74 | USER_SPARE_0 | User Spare 0. | 0x00 | R/W |
| 0x75 | USER_SPARE_1 | User Spare 1. | 0x00 | R/W |
| 0x76 | USER_SPARE_2 | User Spare 2. | 0x00 | R/W |
| 0x77 | USER_SPARE_3 | User Spare 3. | 0x00 | R/W |

## REGISTER BITWISE SUMMARY

*Table 28. Register Bitwise Summary*

| Reg | Name | Bits | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Reset | RW |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x00 | INTERFACE_CONFIG_A | [7:0] | SW_RESET | RESERVED | ADDR_ASCENSION | SDO_ENABLE | RESERVED | | | RESET_SW | 0x10 | R/W |
| 0x01 | INTERFACE_CONFIG_B | [7:0] | SINGLE_INST | RESERVED | | | SHORT_INSTRUCTION | RESERVED | | | 0x00 | R/W |
| 0x02 | DEVICE_CONFIG | [7:0] | RESERVED | | | | | | OPERATING_MODE | | 0x00 | R |
| 0x03 | CHIP_TYPE | [7:0] | RESERVED | | | | CHIP_TYPE | | | | 0x08 | R |
| 0x04 | PRODUCT_ID_L | [7:0] | PRODUCT_ID[7:0] | | | | | | | | 0x30 or 0x31 | R |
| 0x05 | PRODUCT_ID_H | [7:0] | PRODUCT_ID[15:8] | | | | | | | | 0x41 | R |
| 0x06 | CHIP_GRADE | [7:0] | GRADE | | | | DEVICE_REVISION | | | | 0x00 | R |
| 0x0A | SCRATCH_PAD | [7:0] | SCRATCH_VALUE | | | | | | | | 0x00 | R/W |
| 0x0B | SPI_REVISION | [7:0] | SPI_TYPE | | VERSION | | | | | | 0x85 | R |
| 0x0C | VENDOR_L | [7:0] | VID[7:0] | | | | | | | | 0x56 | R |
| 0x0D | VENDOR_H | [7:0] | VID[15:8] | | | | | | | | 0x04 | R |
| 0x0E | STREAM_MODE | [7:0] | LOOP_COUNT | | | | | | | | 0x00 | R/W |
| 0x0F | TRANSFER_CONFIG | [7:0] | RESERVED | | | | | KEEP_STREAM_LENGTH_VAL | RESERVED | | 0x00 | R/W |
| 0x10 | INTERFACE_CONFIG_C | [7:0] | CRC_ENABLE | | STRICT_REGISTER_ACCESS | SEND_STATUS | ACTIVE_INTERFACE_MODE | | CRC_ENABLEB | | 0x2F | R/W |
| 0x11 | INTERFACE_STATUS_A | [7:0] | NOT_READY_ERR | RESERVED | | CLOCK_COUNT_ERR | CRC_ERR | WR_TO_RD_ONLY_REG_ERR | REGISTER_PARTIAL_ACCESS_ERR | ADDRESS_INVALID_ERR | 0x00 | R/W |
| 0x14 | MULTI_DAC_CH_SEL | [7:0] | RESERVED | | | | MULTI_DAC_SEL_CH_3 | MULTI_DAC_SEL_CH_2 | MULTI_DAC_SEL_CH_1 | MULTI_DAC_SEL_CH_0 | 0x00 | R/W |
| 0x15 | | [15:8] | RESERVED | | | | | | | | 0x00 | |
| 0x16 | LDAC_SYNC_ASYNC | [7:0] | RESERVED | | | | LDAC_SYNC_ASYNC_CH_3 | LDAC_SYNC_ASYNC_CH_2 | LDAC_SYNC_ASYNC_CH_1 | LDAC_SYNC_ASYNC_CH_0 | 0x00 | R/W |

## REGISTERS

*Table 28. Register Bitwise Summary (Continued)*

| Reg | Name | Bits | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Reset | RW |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x17 | | [15:8] | RESERVED | | | | | | | | 0x00 | |
| 0x18 | LDAC_HW_SW | [7:0] | RESERVED | | | | LDAC_HW_SW_CH_3 | LDAC_HW_SW_CH_2 | LDAC_HW_SW_CH_1 | LDAC_HW_SW_CH_0 | 0x00 | R/W |
| 0x19 | | [15:8] | RESERVED | | | | | | | | 0x00 | |
| 0x1A to 0x20 by 2 | LDAC_EDGE_SEL_CHn | [7:0] | RESERVED | | | | | | LDAC_HW_EDGE_SEL_CH | | 0x00 | R/W |
| 0x1B to 0x21 by 2 | | [15:8] | RESERVED | | | | | | | | 0x00 | |
| 0x22 | OUT_OPERATING_MODE | [7:0] | RESERVED | | | | OUT_OP_MODE_CH_3 | OUT_OP_MODE_CH_2 | OUT_OP_MODE_CH_1 | OUT_OP_MODE_CH_0 | 0x00 | R/W |
| 0x23 | | [15:8] | RESERVED | | | | | | | | 0x00 | |
| 0x24 | OUT_SWITCH_EN | [7:0] | RESERVED | | | | OUT_SWITCH_EN_CH_3 | OUT_SWITCH_EN_CH_2 | OUT_SWITCH_EN_CH_1 | OUT_SWITCH_EN_CH_0 | 0x00 | R/W |
| 0x25 | | [15:8] | RESERVED | | | | | | | | 0x00 | |
| 0x26 | OUT_EN | [7:0] | RESERVED | | | | HW_SHUTDOWN_EN_CH_3 | HW_SHUTDOWN_EN_CH_2 | HW_SHUTDOWN_EN_CH_1 | HW_SHUTDOWN_EN_CH_0 | 0x00 | R/W |
| 0x27 | | [15:8] | RESERVED | | | | | | | | 0x00 | |
| 0x28 to 0x2E by 2 | OUT_RANGE_CHn | [7:0] | RESERVED | | | | | | OUT_RANGE_CH | | 0x00 | R/W |
| 0x29 to 0x2F by 2 | | [15:8] | RESERVED | | | | | | | | 0x00 | |
| 0x30 | FUNC_EN | [7:0] | RESERVED | | | | FUNC_EN_CH_3 | FUNC_EN_CH_2 | FUNC_EN_CH_1 | FUNC_EN_CH_0 | 0x00 | R/W |
| 0x31 | | [15:8] | RESERVED | | | | | | | | 0x00 | |
| 0x32 to 0x38 by 2 | FUNC_MODE_SEL_CHn | [7:0] | RESERVED | | | | | | | FUNC_MODE_CH | 0x00 | R/W |
| 0x33 to 0x39 by 2 | | [15:8] | RESERVED | | | | | | | | 0x00 | |
| 0x3A to 0x40 by 2 | FUNC_DAC_INPUT_B_CHn | [7:0] | DAC_INPUT_B_CH[7:0] | | | | | | | | 0x00 | R/W |
| 0x3B to 0x41 by 2 | | [15:8] | DAC_INPUT_B_CH[15:8] | | | | | | | | 0x00 | |
| 0x42 to | FUNC_DITHER_PERIOD_CHn | [7:0] | RESERVED | | | | | DITHER_PERIOD_CH | | | 0x00 | R/W |

## REGISTERS

*Table 28. Register Bitwise Summary (Continued)*

| Reg | Name | Bits | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Reset | RW |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x48 by 2 | | | | | | | | | | | | |
| 0x43 to 0x49 by 2 | | [15:8] | RESERVED | | | | | | | | 0x00 | |
| 0x4A to 0x50 by 2 | FUNC_DITHER_PHASE_CHn | [7:0] | RESERVED | | | | | | DITHER_PHASE_CH | | 0x00 | R/W |
| 0x4B to 0x51 by 2 | | [15:8] | RESERVED | | | | | | | | 0x00 | |
| 0x54 | MUX_OUT_SEL | [7:0] | MUX_OUT_EN | RESERVED | | | MUX_PARAM_SEL | | | | 0x00 | R/W |
| 0x55 | | [15:8] | RESERVED | | | | | | | | 0x00 | |
| 0x56 | MUX_OUT_CONTROL | [7:0] | RESERVED | | | | | | | BUFFER_EN | 0x01 | R/W |
| 0x57 | | [15:8] | RESERVED | | | | | | | | 0x00 | |
| 0x58 | TEMP_WARN_INT_EN | [7:0] | RESERVED | | | | TEMP_WARN_INT_EN_CH_3 | TEMP_WARN_INT_EN_CH_2 | TEMP_WARN_INT_EN_CH_1 | TEMP_WARN_INT_EN_CH_0 | 0x00 | R/W |
| 0x59 | | [15:8] | RESERVED | | | | | | | | 0x00 | |
| 0x5A | MULTI_DAC_SW_LDAC | [7:0] | RESERVED | | | | | | | MULTI_DAC_SW_LDAC | 0x00 | R/W |
| 0x5B | | [15:8] | RESERVED | | | | | | | | 0x00 | |
| 0x5C | MULTI_DAC_INPUT_A | [7:0] | MULTI_DAC_INPUT_A[7:0] | | | | | | | | 0x00 | R/W |
| 0x5D | | [15:8] | MULTI_DAC_INPUT_A[15:8] | | | | | | | | 0x00 | |
| 0x5E | DAC_SW_LDAC | [7:0] | RESERVED | | | | DAC_SW_LDAC_CH_3 | DAC_SW_LDAC_CH_2 | DAC_SW_LDAC_CH_1 | DAC_SW_LDAC_CH_0 | 0x00 | R/W |
| 0x5F | | [15:8] | RESERVED | | | | | | | | 0x00 | |
| 0x60 to 0x66 by 2 | DAC_INPUT_A_CHn | [7:0] | DAC_INPUT_A_CH[7:0] | | | | | | | | 0x00 | R/W |
| 0x61 to 0x67 by 2 | | [15:8] | DAC_INPUT_A_CH[15:8] | | | | | | | | 0x00 | |
| 0x68 to 0x6E by 2 | DAC_DATA_READBACK_CHn | [7:0] | DAC_DATA_READBACK_CH[7:0] | | | | | | | | 0x00 | R |
| 0x69 to 0x6F by 2 | | [15:8] | DAC_DATA_READBACK_CH[15:8] | | | | | | | | 0x00 | |
| 0x70 | TEMP_WARN_STAT | [7:0] | RESERVED | | | | TEMP_WARN_FLAG_CH_3 | TEMP_WARN_FLAG_CH_2 | TEMP_WARN_FLAG_CH_1 | TEMP_WARN_FLAG_CH_0 | 0x00 | R/W |
| 0x71 | | [15:8] | RESERVED | | | | | | | | 0x00 | |

## REGISTERS

*Table 28. Register Bitwise Summary (Continued)*

| Reg | Name | Bits | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Reset | RW |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x72 | DIGITAL_STATUS | [7:0] | RESERVED | | | | FUSE_ERROR | INTERFACE_ERROR | POWERUP_COMPLETED | RESET_OCCURRED | 0x01 | R/W |
| 0x73 | BANDGAP_CONTROL | [7:0] | RESERVED | | | | | | | BGAP_HIZ_B | 0x00 | R/W |
| 0x74 | USER_SPARE_0 | [7:0] | USER_SPARE[7:0] | | | | | | | | 0x00 | R/W |
| 0x75 | USER_SPARE_1 | [7:0] | USER_SPARE[15:8] | | | | | | | | 0x00 | R/W |
| 0x76 | USER_SPARE_2 | [7:0] | USER_SPARE[23:16] | | | | | | | | 0x00 | R/W |
| 0x77 | USER_SPARE_3 | [7:0] | USER_SPARE[31:24] | | | | | | | | 0x00 | R/W |

## REGISTER DETAILS

### Interface Configuration A Register

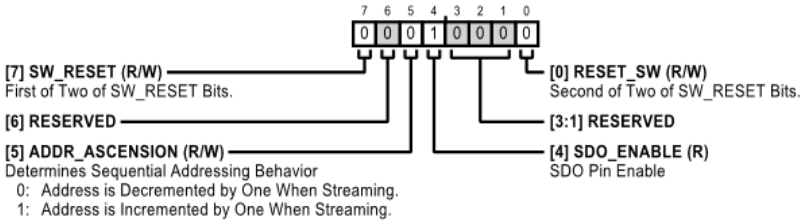



*Figure 79. Address: 0x00, Reset: 0x10, Name: INTERFACE_CONFIG_A*

*Table 29. Bit Descriptions for INTERFACE_CONFIG_A*

| Bits | Bit Name | Description | Reset | Access |
|---|---|---|---|---|
| 7 | SW_RESET | First of Two of SW_RESET Bits. This bit appears in two locations in this register. Both locations must be written at the same time to trigger a software reset of the part. All registers except for this register are reset to their default values. | 0x0 | R/W |
| 6 | RESERVED | Reserved. | 0x0 | R |
| 5 | ADDR_ASCENSION | Determines sequential addressing behavior.<br>0: Address is decremented by one when streaming.<br>1: Address is incremented by one when Streaming. | 0x0 | R/W |
| 4 | SDO_ENABLE | SDO Pin Enable. | 0x1 | R |
| [3:1] | RESERVED | Reserved. | 0x0 | R |
| 0 | RESET_SW | Second of Two of SW_RESET Bits. This bit appears in two locations in this register. Both locations must be written at the same time to trigger a software reset of the part. All registers except for this register are reset to their default values. | 0x0 | R/W |

### Interface Configuration B Register

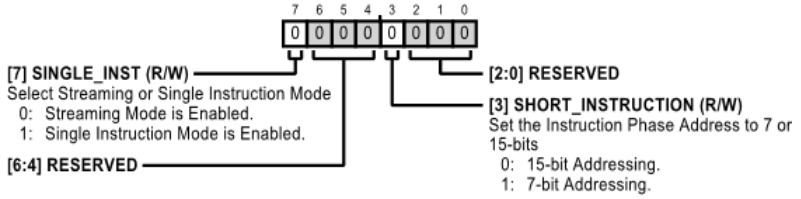



*Figure 80. Address: 0x01, Reset: 0x00, Name: INTERFACE_CONFIG_B*

## REGISTERS

*Table 30. Bit Descriptions for INTERFACE_CONFIG_B*

| Bits | Bit Name | Description | Reset | Access |
|---|---|---|---|---|
| 7 | SINGLE_INST | Select Streaming or Single Instruction Mode.<br>0: Streaming mode is enabled. The address increments/decrements as successive data bytes are received.<br>1: Single instruction mode is enabled. | 0x0 | R/W |
| [6:4] | RESERVED | Reserved. | 0x0 | R |
| 3 | SHORT_INSTRUCTION | Set the Instruction Phase Address to 7 or 15 bits.<br>0: 15-bit addressing.<br>1: 7-bit addressing. | 0x0 | R/W |
| [2:0] | RESERVED | Reserved. | 0x0 | R |

### Device Configuration Register

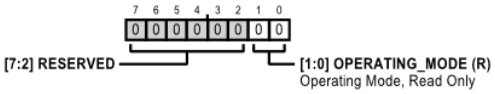



*Figure 81. Address: 0x02, Reset: 0x00, Name: DEVICE_CONFIG*

*Table 31. Bit Descriptions for DEVICE_CONFIG*

| Bits | Bit Name | Description | Reset | Access |
|---|---|---|---|---|
| [7:2] | RESERVED | Reserved. | 0x0 | R |
| [1:0] | OPERATING_MODE | Operating Mode; read only. | 0x0 | R |

### Chip Type Register

The chip type is used to identify the family of ADI devices a given device belongs to. It should be used in conjunction with the Product ID to uniquely identify a given product.

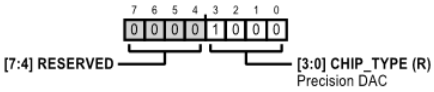



*Figure 82. Address: 0x03, Reset: 0x08, Name: CHIP_TYPE*

*Table 32. Bit Descriptions for CHIP_TYPE*

| Bits | Bit Name | Description | Reset | Access |
|---|---|---|---|---|
| [7:4] | RESERVED | Reserved. | 0x0 | R |
| [3:0] | CHIP_TYPE | Precision DAC. | 0x8 | R |

### Product ID Low Register

Low byte of the Product ID

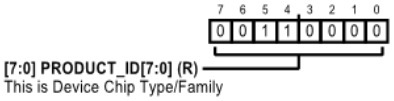



*Figure 83. Address: 0x04, Reset: 0x30, Name: PRODUCT_ID_L*

## REGISTERS

*Table 33. Bit Descriptions for PRODUCT_ID_L*

| Bits | Bit Name | Description | Reset | Access |
|---|---|---|---|---|
| [7:0] | PRODUCT_ID[7:0] | This is Device Chip Type/Family. The product ID should be used in conjunction with chip type to identify a product.<br>▶ 0x30: AD5706R<br>▶ 0x31: AD5705R | 0x30 (AD5706R), 0x31 (AD5705R) | R |

### Product ID High Register

High byte of the Product ID

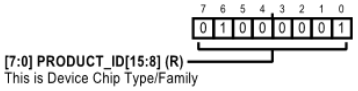



*Figure 84. Address: 0x05, Reset: 0x41, Name: PRODUCT_ID_H*

*Table 34. Bit Descriptions for PRODUCT_ID_H*

| Bits | Bit Name | Description | Reset | Access |
|---|---|---|---|---|
| [7:0] | PRODUCT_ID[15:8] | This is Device Chip Type/Family. The product ID should be used in conjunction with chip type to identify a product. | 0x41 | R |

### Chip Grade Register

Identifies product variations and device revisions

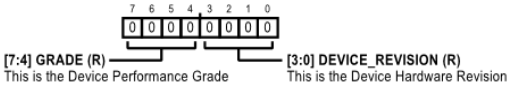



*Figure 85. Address: 0x06, Reset: 0x00, Name: CHIP_GRADE*

*Table 35. Bit Descriptions for CHIP_GRADE*

| Bits | Bit Name | Description | Reset | Access |
|---|---|---|---|---|
| [7:4] | GRADE | This is the Device Performance Grade. | 0x0 | R |
| [3:0] | DEVICE_REVISION | This is the Device Hardware Revision. | 0x0 | R |

### Scratch Pad Register

This may be used to test writes and reads

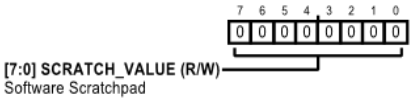



*Figure 86. Address: 0x0A, Reset: 0x00, Name: SCRATCH_PAD*

*Table 36. Bit Descriptions for SCRATCH_PAD*

| Bits | Bit Name | Description | Reset | Access |
|---|---|---|---|---|
| [7:0] | SCRATCH_VALUE | Software scratchpad. Software can write to and read from this location without any device side effects. | 0x0 | R/W |

## REGISTERS

### SPI Revision Register

Indicates the SPI interface revision

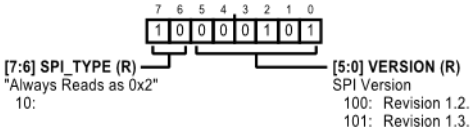



*Figure 87. Address: 0x0B, Reset: 0x85, Name: SPI_REVISION*

*Table 37. Bit Descriptions for SPI_REVISION*

| Bits | Bit Name | Description | Reset | Access |
|---|---|---|---|---|
| [7:6] | SPI_TYPE | "Always Reads as 0x2".<br>10: | 0x2 | R |
| [5:0] | VERSION | SPI Version.<br>100: Revision 1.2.<br>101: Revision 1.3. | 0x5 | R |

### Vendor ID Low Register

Low byte of the Vendor ID

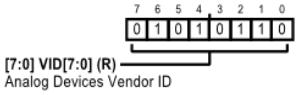



*Figure 88. Address: 0x0C, Reset: 0x56, Name: VENDOR_L*

*Table 38. Bit Descriptions for VENDOR_L*

| Bits | Bit Name | Description | Reset | Access |
|---|---|---|---|---|
| [7:0] | VID[7:0] | Analog Devices Vendor ID. | 0x56 | R |

### Vendor ID High Register

High byte of the Vendor ID

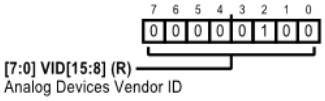



*Figure 89. Address: 0x0D, Reset: 0x04, Name: VENDOR_H*

*Table 39. Bit Descriptions for VENDOR_H*

| Bits | Bit Name | Description | Reset | Access |
|---|---|---|---|---|
| [7:0] | VID[15:8] | Analog Devices Vendor ID. | 0x4 | R |

### Stream Mode Register

Defines the length of the loop when streaming data

## REGISTERS

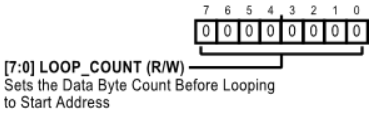



*Figure 90. Address: 0x0E, Reset: 0x00, Name: STREAM_MODE*

*Table 40. Bit Descriptions for STREAM_MODE*

| Bits | Bit Name | Description | Reset | Access |
|---|---|---|---|---|
| [7:0] | LOOP_COUNT | Sets the Data Byte Count Before Looping to the Start Address. When streaming data, a non-zero value sets the number of data bytes written before the address loops back to the start address. A maximum of 255 bytes may be written using this approach. A value of 0x00, disables the loop back, so that addressing wraps around at the upper/lower limits of memory. | 0x0 | R/W |

### Transfer Configuration Register

Controls how data moves between controller and target registers

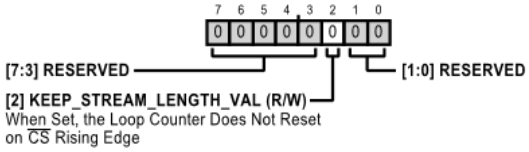



*Figure 91. Address: 0x0F, Reset: 0x00, Name: TRANSFER_CONFIG*

*Table 41. Bit Descriptions for TRANSFER_CONFIG*

| Bits | Bit Name | Description | Reset | Access |
|---|---|---|---|---|
| [7:3] | RESERVED | Reserved. | 0x0 | R |
| 2 | KEEP_STREAM_LENGTH_VAL | When set, the loop counter does not reset on $\overline{CS}$ rising edge. | 0x0 | R/W |
| [1:0] | RESERVED | Reserved. | 0x0 | R |

### Interface Configuration C Register

Additional Interface Configuration Settings

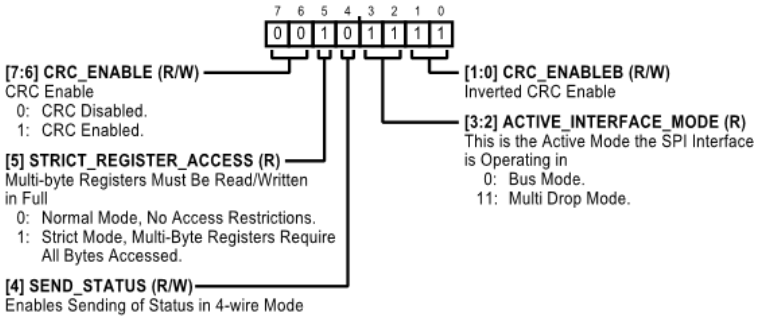



*Figure 92. Address: 0x10, Reset: 0x2F, Name: INTERFACE_CONFIG_C*

*Table 42. Bit Descriptions for INTERFACE_CONFIG_C*

| Bits | Bit Name | Description | Reset | Access |
|---|---|---|---|---|
| [7:6] | CRC_ENABLE | CRC Enable. This is written to enable/disable the use of CRC on the interface. The CRC_ENABLEB bit field must also be written with the inverted value of this bit field for the CRC to be enabled.<br>0: CRC disabled. | 0x0 | R/W |

## REGISTERS

*Table 42. Bit Descriptions for INTERFACE_CONFIG_C (Continued)*

| Bits | Bit Name | Description | Reset | Access |
|---|---|---|---|---|
| | | 1: CRC enabled. | | |
| 5 | STRICT_REGISTER_ACCESS | Multibyte Registers Must Be Read/Written in Full. When this mode is enabled, all bytes of a multi-byte register must be read/written in full.<br>0: Normal mode, No Access Restrictions.<br>1: Strict mode, multibyte registers require all bytes accessed. | 0x1 | R |
| 4 | SEND_STATUS | Enables Sending of Status in 4-Wire Mode. When set, status information is sent by the device on SDO during the instruction phase. When clear, no status is sent during the instruction phase. | 0x0 | R/W |
| [3:2] | ACTIVE_INTERFACE_MODE | This is the Active Mode the SPI Interface is Operating in.<br>00: Bus mode.<br>11: Multi drop mode. | 0x3 | R |
| [1:0] | CRC_ENABLEB | Inverted CRC Enable. This must be written with the inverted value of the CRC_ENABLE. | 0x3 | R/W |

### Interface Status A Register

Status bits are set to '1' to indicate an active condition. They may be cleared by writing a '1' to the corresponding bit location.

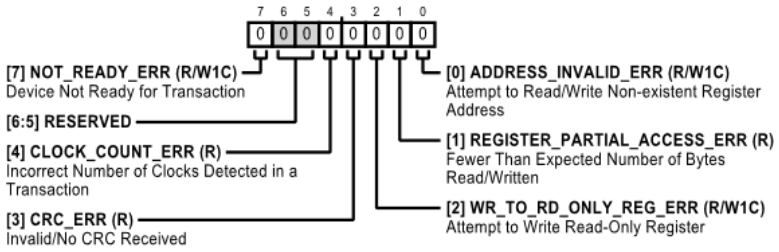



*Figure 93. Address: 0x11, Reset: 0x00, Name: INTERFACE_STATUS_A*

*Table 43. Bit Descriptions for INTERFACE_STATUS_A*

| Bits | Bit Name | Description | Reset | Access |
|---|---|---|---|---|
| 7 | NOT_READY_ERR | Device Not Ready for Transaction. This bit is set if the user attempts to execute a SPI transaction before the completion of digital initialization. | 0x0 | R/W1C |
| [6:5] | RESERVED | Reserved. | 0x0 | R |
| 4 | CLOCK_COUNT_ERR | Incorrect Number of Clocks Detected in a Transaction. This bit is set if the number of clocks received in a given SPI frame, defined as from $\overline{CS}$ going low to $\overline{CS}$ going high, is not a multiple of 8 bits. | 0x0 | R |
| 3 | CRC_ERR | Invalid/No CRC Received. This bit is set when the controller fails to send a CRC or when the device calculates and checks the CRC and finds the CRC value is incorrect. | 0x0 | R |
| 2 | WR_TO_RD_ONLY_REG_ERR | Attempt to Write Read-Only Register. This bit is set when a write to a register that is read-only is attempted by the SPI controller. | 0x0 | R/W1C |
| 1 | REGISTER_PARTIAL_ACCESS_ERR | Fewer Than Expected Number of Bytes Read/Written. This bit is only valid when strict register access is enabled. | 0x0 | R |
| 0 | ADDRESS_INVALID_ERR | Attempt to Read/Write Non-Existent Register Address. This bit is set when the user tries to access register addresses outside allowed memory map space. | 0x0 | R/W1C |

### Multi-DAC Channel Select Register

This register is used to select which channels are included in the multi-DAC updates

## REGISTERS

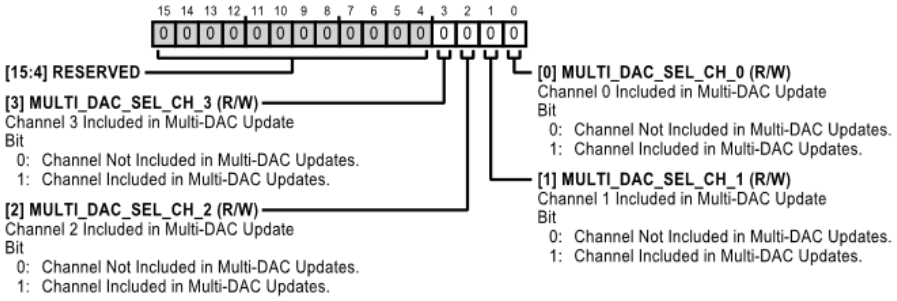



*Figure 94. Address: 0x14, Reset: 0x0000, Name: MULTI_DAC_CH_SEL*

*Table 44. Bit Descriptions for MULTI_DAC_CH_SEL*

| Bits | Bit Name | Description | Reset | Access |
|---|---|---|---|---|
| [15:4] | RESERVED | Reserved. | 0x0 | R |
| 3 | MULTI_DAC_SEL_CH_3 | Channel 3 Included in Multi-DAC Update Bit.<br>0: Channel not included in multi-DAC updates.<br>1: Channel included in multi-DAC updates. | 0x0 | R/W |
| 2 | MULTI_DAC_SEL_CH_2 | Channel 2 Included in Multi-DAC Update Bit.<br>0: Channel not included in multi-DAC updates.<br>1: Channel included in multi-DAC updates. | 0x0 | R/W |
| 1 | MULTI_DAC_SEL_CH_1 | Channel 1 Included in Multi-DAC Update Bit.<br>0: Channel not included in multi-DAC updates.<br>1: Channel included in multi-DAC updates. | 0x0 | R/W |
| 0 | MULTI_DAC_SEL_CH_0 | Channel 0 Included in Multi-DAC Update Bit.<br>0: Channel not included in multi-DAC updates.<br>1: Channel included in multi-DAC updates. | 0x0 | R/W |

### Sync-Async Load Register

This register is used to determine how the DAC output is updated

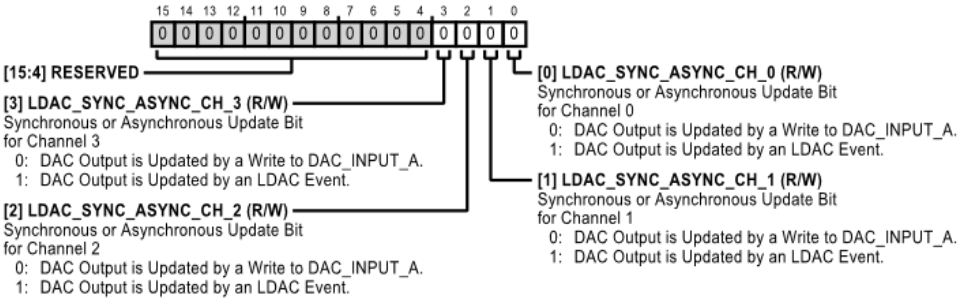



*Figure 95. Address: 0x16, Reset: 0x0000, Name: LDAC_SYNC_ASYNC*

*Table 45. Bit Descriptions for LDAC_SYNC_ASYNC*

| Bits | Bit Name | Description | Reset | Access |
|---|---|---|---|---|
| [15:4] | RESERVED | Reserved. | 0x0 | R |
| 3 | LDAC_SYNC_ASYNC_CH_3 | Synchronous or Asynchronous Update Bit for Channel 3.<br>0: DAC output is updated by a write to DAC_INPUT_A.<br>1: DAC output is updated by an LDAC event. | 0x0 | R/W |
| 2 | LDAC_SYNC_ASYNC_CH_2 | Synchronous or Asynchronous Update Bit for Channel 2. | 0x0 | R/W |

## REGISTERS

*Table 45. Bit Descriptions for LDAC_SYNC_ASYNC (Continued)*

| Bits | Bit Name | Description | Reset | Access |
|---|---|---|---|---|
| | | 0: DAC output is updated by a write to DAC_INPUT_A.<br>1: DAC output is updated by an LDAC event. | | |
| 1 | LDAC_SYNC_ASYNC_CH_1 | Synchronous or Asynchronous Update Bit for Channel 1.<br>0: DAC output is updated by a write to DAC_INPUT_A.<br>1: DAC output is updated by an LDAC event. | 0x0 | R/W |
| 0 | LDAC_SYNC_ASYNC_CH_0 | Synchronous or Asynchronous Update Bit for Channel 0.<br>0: DAC output is updated by a write to DAC_INPUT_A.<br>1: DAC output is updated by an LDAC event. | 0x0 | R/W |

### Hardware-Software Trigger Register

This register is used to determine how the load DAC function is triggered

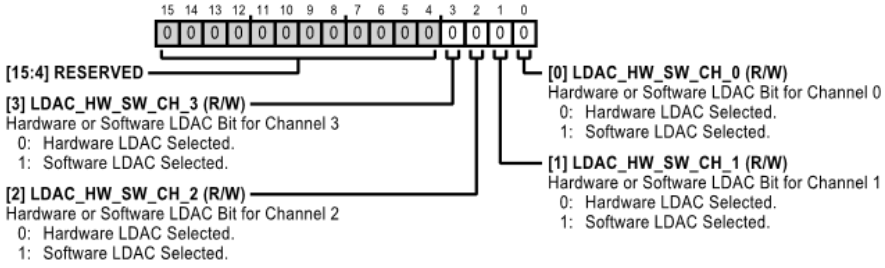



*Figure 96. Address: 0x18, Reset: 0x0000, Name: LDAC_HW_SW*

*Table 46. Bit Descriptions for LDAC_HW_SW*

| Bits | Bit Name | Description | Reset | Access |
|---|---|---|---|---|
| [15:4] | RESERVED | Reserved. | 0x0 | R |
| 3 | LDAC_HW_SW_CH_3 | Hardware or Software LDAC Bit for Channel 3.<br>0: Hardware LDAC selected.<br>1: Software LDAC selected. | 0x0 | R/W |
| 2 | LDAC_HW_SW_CH_2 | Hardware or Software LDAC Bit for Channel 2.<br>0: Hardware LDAC selected.<br>1: Software LDAC selected. | 0x0 | R/W |
| 1 | LDAC_HW_SW_CH_1 | Hardware or Software LDAC Bit for Channel 1.<br>0: Hardware LDAC selected.<br>1: Software LDAC selected. | 0x0 | R/W |
| 0 | LDAC_HW_SW_CH_0 | Hardware or Software LDAC Bit for Channel 0.<br>0: Hardware LDAC selected.<br>1: Software LDAC selected. | 0x0 | R/W |

### LDAC Edge Select Register

This register is used to determine the active edge of the LDAC/TGP/DCK pin

## REGISTERS

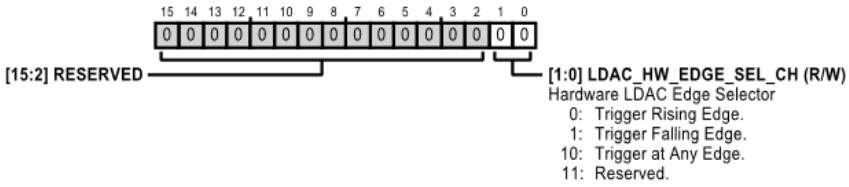



*Figure 97. Address: 0x1A to 0x20 (Increments of 2), Reset: 0x0000, Name: LDAC_EDGE_SEL_CHn*

*Table 47. Bit Descriptions for LDAC_EDGE_SEL_CHn*

| Bits | Bit Name | Description | Reset | Access |
|---|---|---|---|---|
| [15:2] | RESERVED | Reserved. | 0x0 | R |
| [1:0] | LDAC_HW_EDGE_SEL_CH | Hardware LDAC Edge Selector.<br>00: Trigger rising edge.<br>01: Trigger falling edge.<br>10: Trigger at any edge.<br>11: Reserved. | 0x0 | R/W |

### Output Operating Mode Register

This register is used to control the operating mode of the output stage

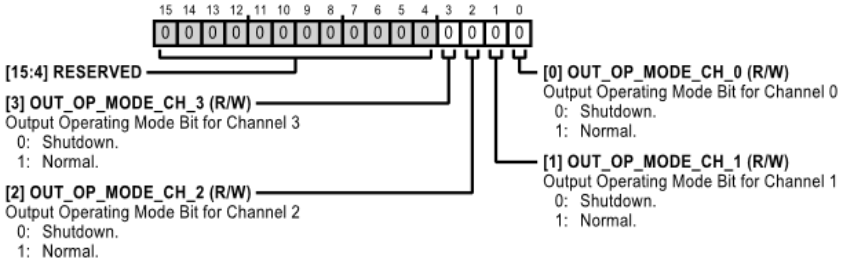



*Figure 98. Address: 0x22, Reset: 0x0000, Name: OUT_OPERATING_MODE*

*Table 48. Bit Descriptions for OUT_OPERATING_MODE*

| Bits | Bit Name | Description | Reset | Access |
|---|---|---|---|---|
| [15:4] | RESERVED | Reserved. | 0x0 | R |
| 3 | OUT_OP_MODE_CH_3 | Output Operating Mode Bit for Channel 3.<br>0: Shutdown.<br>1: Normal. | 0x0 | R/W |
| 2 | OUT_OP_MODE_CH_2 | Output Operating Mode Bit for Channel 2.<br>0: Shutdown.<br>1: Normal. | 0x0 | R/W |
| 1 | OUT_OP_MODE_CH_1 | Output Operating Mode Bit for Channel 1.<br>0: Shutdown.<br>1: Normal. | 0x0 | R/W |
| 0 | OUT_OP_MODE_CH_0 | Output Operating Mode Bit for Channel 0.<br>0: Shutdown.<br>1: Normal. | 0x0 | R/W |

### Output Switch Enable Register

This register is used to control the ground switch of the output stage

## REGISTERS

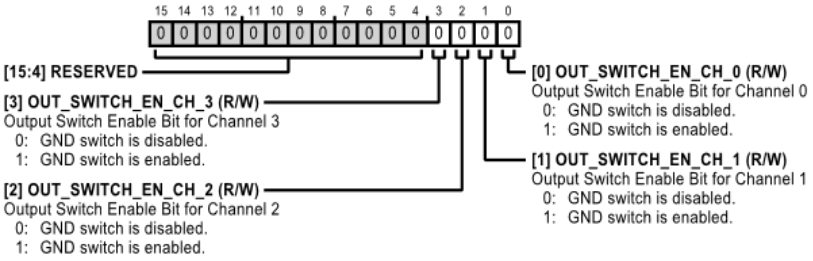



*Figure 99. Address: 0x24, Reset: 0x0000, Name: OUT_SWITCH_EN*

*Table 49. Bit Descriptions for OUT_SWITCH_EN*

| Bits | Bit Name | Description | Reset | Access |
|---|---|---|---|---|
| [15:4] | RESERVED | Reserved. | 0x0 | R |
| 3 | OUT_SWITCH_EN_CH_3 | Output Switch Enable Bit for Channel 3.<br>0: GND switch is disabled.<br>1: GND switch is enabled. | 0x0 | R/W |
| 2 | OUT_SWITCH_EN_CH_2 | Output Switch Enable Bit for Channel 2.<br>0: GND switch is disabled.<br>1: GND switch is enabled. | 0x0 | R/W |
| 1 | OUT_SWITCH_EN_CH_1 | Output Switch Enable Bit for Channel 1.<br>0: GND switch is disabled.<br>1: GND switch is enabled. | 0x0 | R/W |
| 0 | OUT_SWITCH_EN_CH_0 | Output Switch Enable Bit for Channel 0.<br>0: GND switch is disabled.<br>1: GND switch is enabled. | 0x0 | R/W |

### OUT_EN Pin Enable Register

This register is used to enable the shutdown of the output stage with the OUT_EN pin

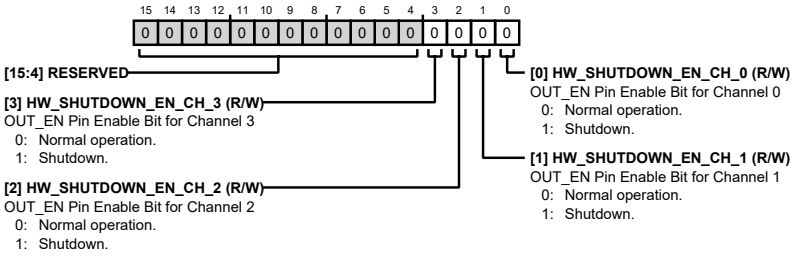



*Figure 100. Address: 0x26, Reset: 0x0000, Name: OUT_EN*

*Table 50. Bit Descriptions for OUT_EN*

| Bits | Bit Name | Description | Reset | Access |
|---|---|---|---|---|
| [15:4] | RESERVED | Reserved. | 0x0 | R |
| 3 | HW_SHUTDOWN_EN_CH_3 | OUT_EN Pin Enable Bit for Channel 3.<br>0: Normal operation.<br>1: IOUT3 output disable. | 0x0 | R/W |
| 2 | HW_SHUTDOWN_EN_CH_2 | OUT_EN Pin Enable Bit for Channel 2.<br>0: Normal operation.<br>1: IOUT2 output disable. | 0x0 | R/W |

## REGISTERS

*Table 50. Bit Descriptions for OUT_EN (Continued)*

| Bits | Bit Name | Description | Reset | Access |
|---|---|---|---|---|
| 1 | HW_SHUTDOWN_EN_CH_1 | OUT_EN Pin Enable Bit for Channel 1.<br>0: Normal operation.<br>1: IOUT1 output disable. | 0x0 | R/W |
| 0 | HW_SHUTDOWN_EN_CH_0 | OUT_EN Pin Enable Bit for Channel 0.<br>0: Normal operation.<br>1: IOUT0 output disable. | 0x0 | R/W |

### Output Range Register

This register is used to control the current range of the output stage

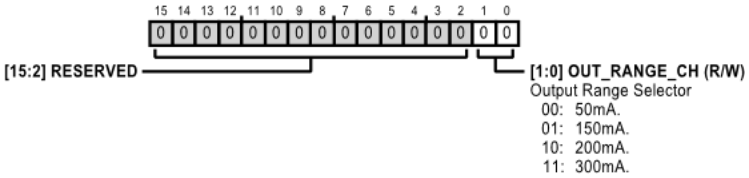



*Figure 101. Address: 0x28 to 0x2E (Increments of 2), Reset: 0x0000, Name: OUT_RANGE_CHn*

*Table 51. Bit Descriptions for OUT_RANGE_CHn*

| Bits | Bit Name | Description | Reset | Access |
|---|---|---|---|---|
| [15:2] | RESERVED | Reserved. | 0x0 | R |
| [1:0] | OUT_RANGE_CH | Output Range Selector.<br>00: 50mA.<br>01: 150mA.<br>10: 200mA.<br>11: 300mA. | 0x0 | R/W |

### Function Enable Register

This register is used to enable the other DAC functions

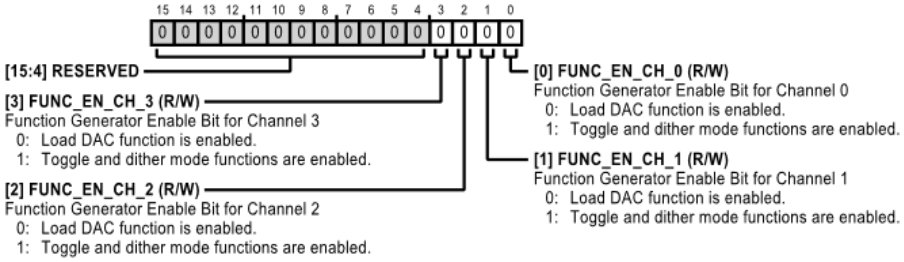



*Figure 102. Address: 0x30, Reset: 0x0000, Name: FUNC_EN*

*Table 52. Bit Descriptions for FUNC_EN*

| Bits | Bit Name | Description | Reset | Access |
|---|---|---|---|---|
| [15:4] | RESERVED | Reserved. | 0x0 | R |
| 3 | FUNC_EN_CH_3 | Function Generator Enable Bit for Channel 3.<br>0: Load DAC function is enabled.<br>1: Toggle and dither mode functions are enabled. | 0x0 | R/W |

## REGISTERS

*Table 52. Bit Descriptions for FUNC_EN (Continued)*

| Bits | Bit Name | Description | Reset | Access |
|---|---|---|---|---|
| 2 | FUNC_EN_CH_2 | Function Generator Enable Bit for Channel 2.<br>0: Load DAC function is enabled.<br>1: Toggle and dither mode functions are enabled. | 0x0 | R/W |
| 1 | FUNC_EN_CH_1 | Function Generator Enable Bit for Channel 1.<br>0: Load DAC function is enabled.<br>1: Toggle and dither mode functions are enabled. | 0x0 | R/W |
| 0 | FUNC_EN_CH_0 | Function Generator Enable Bit for Channel 0.<br>0: Load DAC function is enabled.<br>1: Toggle and dither mode functions are enabled. | 0x0 | R/W |

### Function Mode Selector Register

This register is used to select the active DAC function

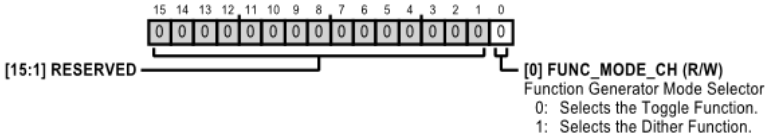



*Figure 103. Address: 0x32 to 0x38 (Increments of 2), Reset: 0x0000, Name: FUNC_MODE_SEL_CHn*

*Table 53. Bit Descriptions for FUNC_MODE_SEL_CHn*

| Bits | Bit Name | Description | Reset | Access |
|---|---|---|---|---|
| [15:1] | RESERVED | Reserved. | 0x0 | R |
| 0 | FUNC_MODE_CH | Function Generator Mode Selector.<br>0: Selects the toggle function.<br>1: Selects the dither function. | 0x0 | R/W |

### DAC Input B Register

This register is used to store the DAC data for the DAC functions

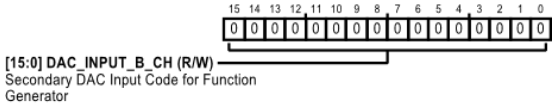



*Figure 104. Address: 0x3A to 0x40 (Increments of 2), Reset: 0x0000, Name: FUNC_DAC_INPUT_B_CHn*

*Table 54. Bit Descriptions for FUNC_DAC_INPUT_B_CHn*

| Bits | Bit Name | Description | Reset | Access |
|---|---|---|---|---|
| [15:0] | DAC_INPUT_B_CH | Secondary DAC Input Code for Function Generator. | 0x0 | R/W |

### Dither Settings Register

This register is used to configure the period for the dither function

## REGISTERS

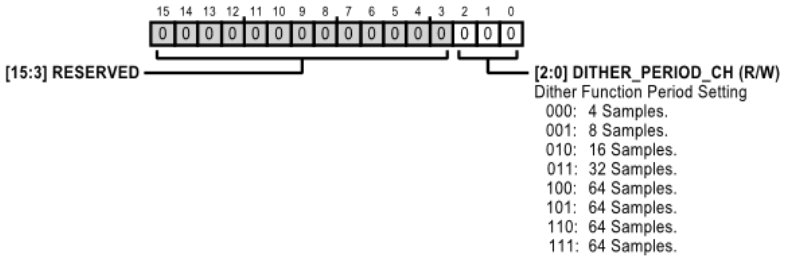



*Figure 105. Address: 0x42 to 0x48 (Increments of 2), Reset: 0x0000, Name: FUNC_DITHER_PERIOD_CHn*

*Table 55. Bit Descriptions for FUNC_DITHER_PERIOD_CHn*

| Bits | Bit Name | Description | Reset | Access |
|---|---|---|---|---|
| [15:3] | RESERVED | Reserved. | 0x0 | R |
| [2:0] | DITHER_PERIOD_CH | Dither Function Period Setting.<br>000: 4 samples.<br>001: 8 samples.<br>010: 16 samples.<br>011: 32 samples.<br>100: 64 samples.<br>101: 64 samples.<br>110: 64 samples.<br>111: 64 samples. | 0x0 | R/W |

### Dither Settings Register

This register is used to configure the phase for the dither function

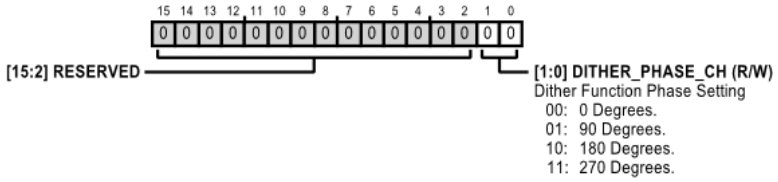



*Figure 106. Address: 0x4A to 0x50 (Increments of 2), Reset: 0x0000, Name: FUNC_DITHER_PHASE_CHn*

*Table 56. Bit Descriptions for FUNC_DITHER_PHASE_CHn*

| Bits | Bit Name | Description | Reset | Access |
|---|---|---|---|---|
| [15:2] | RESERVED | Reserved. | 0x0 | R |
| [1:0] | DITHER_PHASE_CH | Dither Function Phase Setting.<br>00: 0 degrees.<br>01: 90 degrees.<br>10: 180 degrees.<br>11: 270 degrees. | 0x0 | R/W |

### MUX Out Select Register

This register is used to control the MUX_OUT pin

## REGISTERS

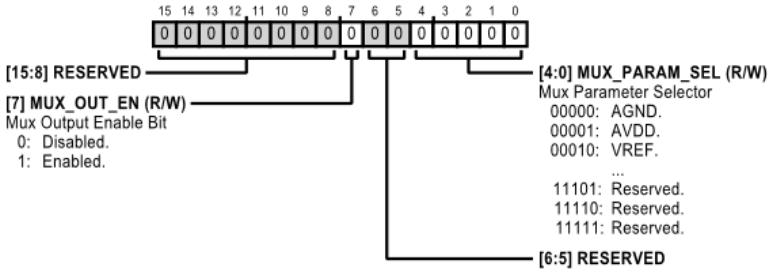



*Figure 107. Address: 0x54, Reset: 0x0000, Name: MUX_OUT_SEL*

*Table 57. Bit Descriptions for MUX_OUT_SEL*

| Bits | Bit Name | Description | Reset | Access |
|---|---|---|---|---|
| [15:8] | RESERVED | Reserved. | 0x0 | R |
| 7 | MUX_OUT_EN | Mux Output Enable Bit.<br>0: Disabled.<br>1: Enabled. | 0x0 | R/W |
| [6:5] | RESERVED | Reserved. | 0x0 | R |
| [4:0] | MUX_PARAM_SEL | Mux Parameter Selector.<br>00000: GND.<br>00001: AVDD.<br>00010: VREF.<br>00011: GND.<br>00100: DAC VOUT CH0.<br>00101: DAC VOUT CH1.<br>00110: DAC VOUT CH2.<br>00111: DAC VOUT CH3.<br>01000: DAC IOUT CH0.<br>01001: DAC IOUT CH1.<br>01010: DAC IOUT CH2.<br>01011: DAC IOUT CH3.<br>01100: DAC PVDD CH0.<br>01101: DAC PVDD CH1.<br>01110: DAC PVDD CH2.<br>01111: DAC PVDD CH3.<br>10000: DAC TDIODE CH0.<br>10001: DAC TDIODE CH1.<br>10010: DAC TDIODE CH2.<br>10011: DAC TDIODE CH3.<br>10100: MUX_IN.<br>10101: MUX_IN.<br>10110: MUX_IN.<br>10111: MUX_IN.<br>11000: Reserved.<br>11001: Reserved.<br>11010: Reserved.<br>11011: Reserved.<br>11100: Reserved.<br>11101: Reserved.<br>11110: Reserved. | 0x0 | R/W |

## REGISTERS

*Table 57. Bit Descriptions for MUX_OUT_SEL (Continued)*

| Bits | Bit Name | Description | Reset | Access |
|---|---|---|---|---|
| | | 11111: Reserved. | | |

### MUX Out Control Register

This register is used to control the buffer on the MUX_OUT pin

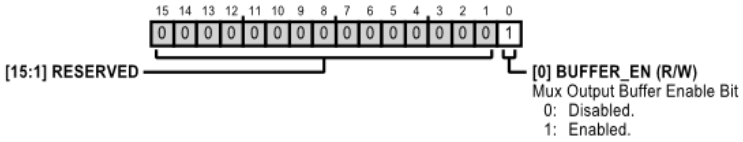



*Figure 108. Address: 0x56, Reset: 0x0001, Name: MUX_OUT_CONTROL*

*Table 58. Bit Descriptions for MUX_OUT_CONTROL*

| Bits | Bit Name | Description | Reset | Access |
|---|---|---|---|---|
| [15:1] | RESERVED | Reserved. | 0x0 | R |
| 0 | BUFFER_EN | Mux Output Buffer Enable Bit.<br>0: Disabled.<br>1: Enabled. | 0x1 | R/W |

### Temperature Warning Interrupt Enabled Register

This register is used to enable the status bits in the TEMP_WARN_STAT register

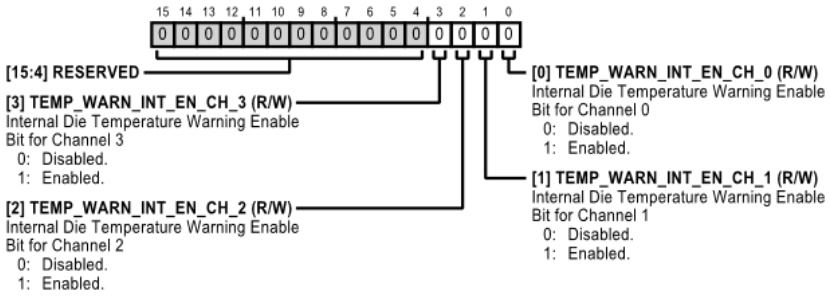



*Figure 109. Address: 0x58, Reset: 0x0000, Name: TEMP_WARN_INT_EN*

*Table 59. Bit Descriptions for TEMP_WARN_INT_EN*

| Bits | Bit Name | Description | Reset | Access |
|---|---|---|---|---|
| [15:4] | RESERVED | Reserved. | 0x0 | R |
| 3 | TEMP_WARN_INT_EN_CH_3 | Internal Die Temperature Warning Enable Bit for Channel 3.<br>0: Disabled.<br>1: Enabled. | 0x0 | R/W |
| 2 | TEMP_WARN_INT_EN_CH_2 | Internal Die Temperature Warning Enable Bit for Channel 2.<br>0: Disabled.<br>1: Enabled. | 0x0 | R/W |
| 1 | TEMP_WARN_INT_EN_CH_1 | Internal Die Temperature Warning Enable Bit for Channel 1.<br>0: Disabled.<br>1: Enabled. | 0x0 | R/W |
| 0 | TEMP_WARN_INT_EN_CH_0 | Internal Die Temperature Warning Enable Bit for Channel 0.<br>0: Disabled. | 0x0 | R/W |

## REGISTERS

*Table 59. Bit Descriptions for TEMP_WARN_INT_EN (Continued)*

| Bits | Bit Name | Description | Reset | Access |
|---|---|---|---|---|
| | | 1: Enabled. | | |

### Software LDAC for Multiple DAC Input Register

This register is used to trigger the software LDAC mode for multiple channels

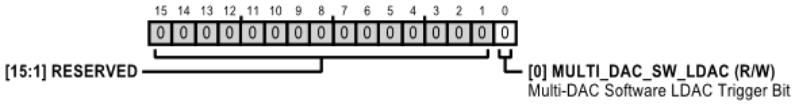



*Figure 110. Address: 0x5A, Reset: 0x0000, Name: MULTI_DAC_SW_LDAC*

*Table 60. Bit Descriptions for MULTI_DAC_SW_LDAC*

| Bits | Bit Name | Description | Reset | Access |
|---|---|---|---|---|
| [15:1] | RESERVED | Reserved. | 0x0 | R |
| 0 | MULTI_DAC_SW_LDAC | Multi-DAC Software LDAC Trigger Bit. | 0x0 | R/W |

### Multiple DAC Input Register

This register contains the data to be used for the multi-DAC function

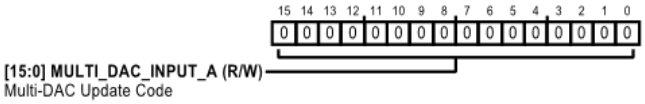



*Figure 111. Address: 0x5C, Reset: 0x0000, Name: MULTI_DAC_INPUT_A*

*Table 61. Bit Descriptions for MULTI_DAC_INPUT_A*

| Bits | Bit Name | Description | Reset | Access |
|---|---|---|---|---|
| [15:0] | MULTI_DAC_INPUT_A | Multi-DAC Update Code. | 0x0 | R/W |

### Software LDAC for DAC Input Register

This register is used to trigger the software LDAC mode

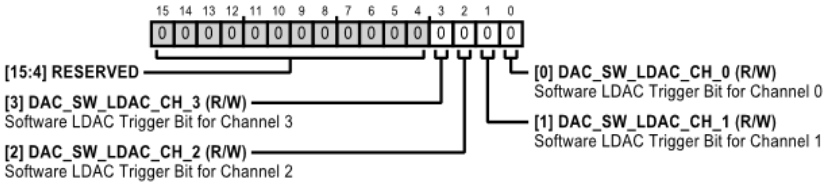



*Figure 112. Address: 0x5E, Reset: 0x0000, Name: DAC_SW_LDAC*

*Table 62. Bit Descriptions for DAC_SW_LDAC*

| Bits | Bit Name | Description | Reset | Access |
|---|---|---|---|---|
| [15:4] | RESERVED | Reserved. | 0x0 | R |
| 3 | DAC_SW_LDAC_CH_3 | Software LDAC Trigger Bit for Channel 3. | 0x0 | R/W |
| 2 | DAC_SW_LDAC_CH_2 | Software LDAC Trigger Bit for Channel 2. | 0x0 | R/W |
| 1 | DAC_SW_LDAC_CH_1 | Software LDAC Trigger Bit for Channel 1. | 0x0 | R/W |
| 0 | DAC_SW_LDAC_CH_0 | Software LDAC Trigger Bit for Channel 0. | 0x0 | R/W |

## REGISTERS

### DAC Input Register

This register is used for storing the DAC data that can be loaded to the DAC_DATA register during load DAC functions

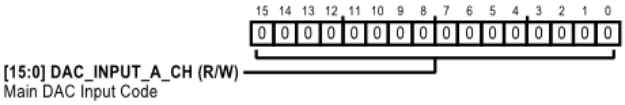



*Figure 113. Address: 0x60 to 0x66 (Increments of 2), Reset: 0x0000, Name: DAC_INPUT_A_CHn*

*Table 63. Bit Descriptions for DAC_INPUT_A_CHn*

| Bits | Bit Name | Description | Reset | Access |
|---|---|---|---|---|
| [15:0] | DAC_INPUT_A_CH | Main DAC Input Code. | 0x0 | R/W |

### DAC Data Readback Register

This register contains the current DAC code in the DAC core

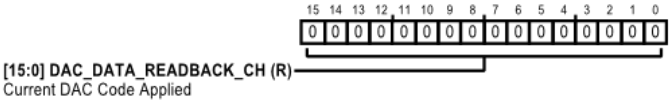



*Figure 114. Address: 0x68 to 0x6E (Increments of 2), Reset: 0x0000, Name: DAC_DATA_READBACK_CHn*

*Table 64. Bit Descriptions for DAC_DATA_READBACK_CHn*

| Bits | Bit Name | Description | Reset | Access |
|---|---|---|---|---|
| [15:0] | DAC_DATA_READBACK_CH | Current DAC Code Applied. | 0x0 | R |

### Temperature Warning Flags Status Register

This register contains the status of the internal die temperature

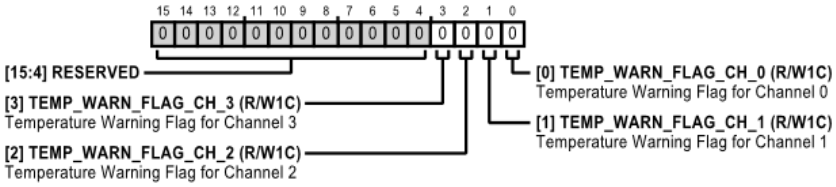



*Figure 115. Address: 0x70, Reset: 0x0000, Name: TEMP_WARN_STAT*

*Table 65. Bit Descriptions for TEMP_WARN_STAT*

| Bits | Bit Name | Description | Reset | Access |
|---|---|---|---|---|
| [15:4] | RESERVED | Reserved. | 0x0 | R |
| 3 | TEMP_WARN_FLAG_CH_3 | Temperature Warning Flag for Channel 3. This bit is set to 1'b1 when the internal die temperature of channel 3 reaches approximately 125°C. This bit can be cleared by writing a 1'b1, as long as the temperature is below 125°C. | 0x0 | R/W1C |
| 2 | TEMP_WARN_FLAG_CH_2 | Temperature Warning Flag for Channel 2. This bit is set to 1'b1 when the internal die temperature of channel 2 reaches approximately 125°C. This bit can be cleared by writing a 1'b1, as long as the temperature is below 125°C. | 0x0 | R/W1C |
| 1 | TEMP_WARN_FLAG_CH_1 | Temperature Warning Flag for Channel 1. This bit is set to 1'b1 when the internal die temperature of channel 1 reaches approximately 125°C. This bit can be cleared by writing a 1'b1, as long as the temperature is below 125°C. | 0x0 | R/W1C |

## REGISTERS

*Table 65. Bit Descriptions for TEMP_WARN_STAT (Continued)*

| Bits | Bit Name | Description | Reset | Access |
|---|---|---|---|---|
| 0 | TEMP_WARN_FLAG_CH_0 | Temperature Warning Flag for Channel 0. This bit is set to 1'b1 when the internal die temperature of channel 0 reaches approximately 125°C. This bit can be cleared by writing a 1'b1, as long as the temperature is below 125°C. | 0x0 | R/W1C |

### Digital Status Register

This register contains the status signals of the digital core

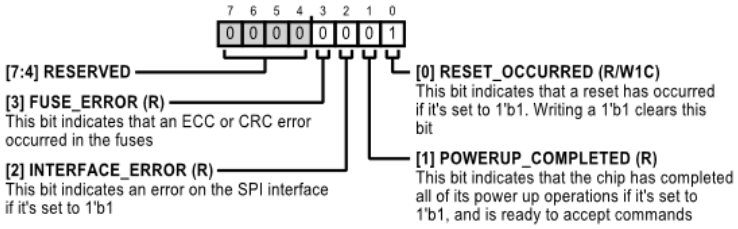



*Figure 116. Address: 0x72, Reset: 0x01, Name: DIGITAL_STATUS*

*Table 66. Bit Descriptions for DIGITAL_STATUS*

| Bits | Bit Name | Description | Reset | Access |
|---|---|---|---|---|
| [7:4] | RESERVED | Reserved. | 0x0 | R |
| 3 | FUSE_ERROR | This bit indicates that an ECC or CRC error occurred in the fuses. | 0x0 | R |
| 2 | INTERFACE_ERROR | This bit indicates an error on the SPI interface if it's set to 1'b1. | 0x0 | R |
| 1 | POWERUP_COMPLETED | This bit indicates that the chip has completed all of its power up operations if it's set to 1'b1 and is ready to accept commands. | 0x0 | R |
| 0 | RESET_OCCURRED | This bit indicates that a reset has occurred if it's set to 1'b1. Writing a 1'b1 clears this bit. | 0x1 | R/W1C |

### Bandgap Control Register

This register contains the control for the internal bandgap

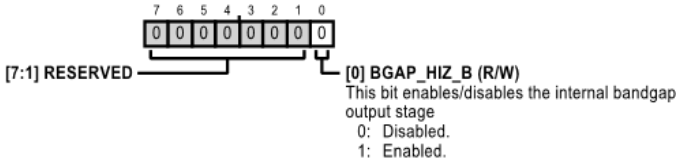



*Figure 117. Address: 0x73, Reset: 0x00, Name: BANDGAP_CONTROL*

*Table 67. Bit Descriptions for BANDGAP_CONTROL*

| Bits | Bit Name | Description | Reset | Access |
|---|---|---|---|---|
| [7:1] | RESERVED | Reserved. | 0x0 | R |
| 0 | BGAP_HIZ_B | This bit enables/disables the internal bandgap output stage.<br>0: Disabled.<br>1: Enabled. | 0x0 | R/W |

### User Spare 0 Register

Spare register

## REGISTERS

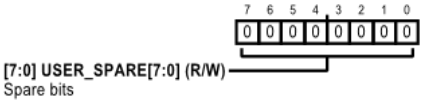



*Figure 118. Address: 0x74, Reset: 0x00, Name: USER_SPARE_0*

*Table 68. Bit Descriptions for USER_SPARE_0*

| Bits | Bit Name | Description | Reset | Access |
|---|---|---|---|---|
| [7:0] | USER_SPARE[7:0] | Spare Bits. | 0x0 | R/W |

### User Spare 1 Register

Spare register

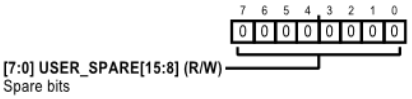



*Figure 119. Address: 0x75, Reset: 0x00, Name: USER_SPARE_1*

*Table 69. Bit Descriptions for USER_SPARE_1*

| Bits | Bit Name | Description | Reset | Access |
|---|---|---|---|---|
| [7:0] | USER_SPARE[15:8] | Spare Bits. | 0x0 | R/W |

### User Spare 2 Register

Spare register

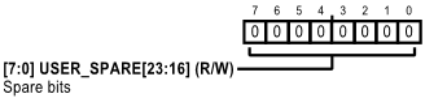



*Figure 120. Address: 0x76, Reset: 0x00, Name: USER_SPARE_2*

*Table 70. Bit Descriptions for USER_SPARE_2*

| Bits | Bit Name | Description | Reset | Access |
|---|---|---|---|---|
| [7:0] | USER_SPARE[23:16] | Spare Bits. | 0x0 | R/W |

### User Spare 3 Register

Spare register

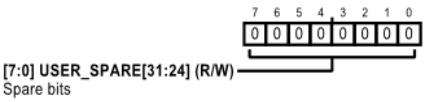



*Figure 121. Address: 0x77, Reset: 0x00, Name: USER_SPARE_3*

*Table 71. Bit Descriptions for USER_SPARE_3*

| Bits | Bit Name | Description | Reset | Access |
|---|---|---|---|---|
| [7:0] | USER_SPARE[31:24] | Spare Bits. | 0x0 | R/W |

## OUTLINE DIMENSIONS

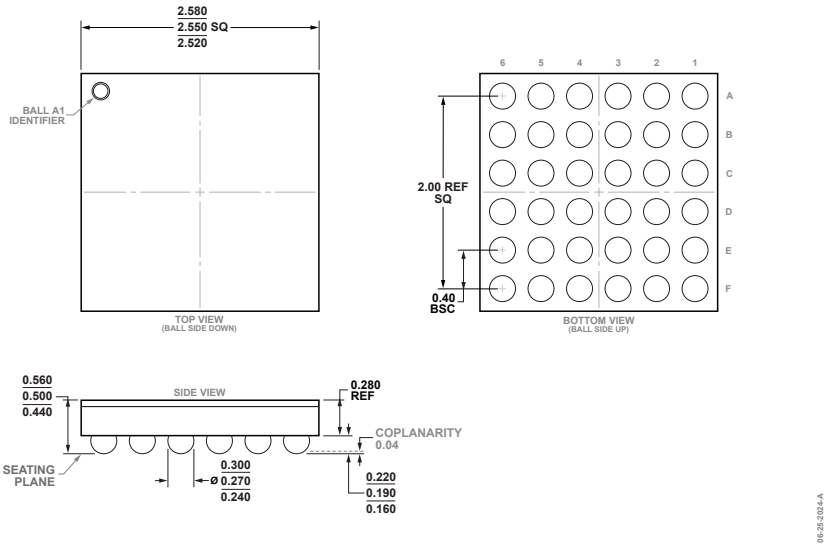
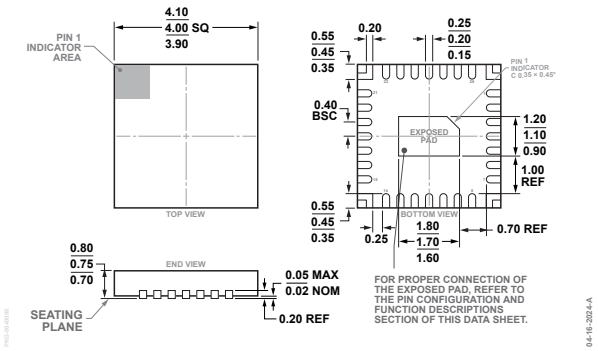



*Figure 122. 36-Ball Wafer Level Chip Scale Package (WLCSP)*
*(CB-36-12)*
*Dimensions shown in millimeters*




RECOMMENDED SOLDER PAD LAYOUT
(TOP VIEW)

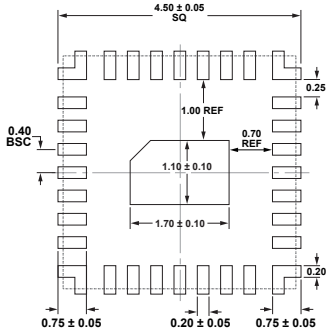



*Figure 123. 28-Lead Lead Frame Chip Scale Package [LFCSP]*
*4mm × 4mm Body and 0.75 mm Package Height*
*(CP-28-17)*
*Dimensions shown in millimeters*

## OUTLINE DIMENSIONS

## ORDERING GUIDE

| Model[1] | Temperature Range | Package Description | Package Option |
|---|---|---|---|
| AD5706RBCBZ-RL7 | −40°C to +125°C | 36-Ball Wafer Level Chip Scale Package [WLCSP] | CB-36-12 |
| AD5706RBCPZ-RL7 | −40°C to +125°C | 28-Lead Lead Frame Chip Scale Package [LFCSP] | CP-28-17 |
| AD5705RBCBZ-RL7 | −40°C to +125°C | 36-Ball Wafer Level Chip Scale Package [WLCSP] | CB-36-12 |
| AD5705RBCPZ-RL7 | −40°C to +125°C | 28-Lead Lead Frame Chip Scale Package [LFCSP] | CP-28-17 |

[1] Z = RoHS Compliant Part.