Data Sheet

# AD5710R

## 8-channel 16-bit Configurable IDAC/VDAC with On-Chip Reference

## FEATURES

- Octal channel, configurable as any combination of VDAC or IDAC
- Guaranteed monotonicity : ±1 $LSB_{16}$ Max DNL
- 50mA output current drive for IDAC and VDAC modes
- 70mV max headroom for VDAC mode
- 0.25V max headroom for IDAC mode
- 2.5V internal voltage reference, 3ppm/°C typical
- Device addressable SPI interface
- Output supply range: +1.71V to +5.5V
- Diagnostic monitoring for output voltage, output current, and die temperature
- Ultra small form factor: 2.1mm × 2.2mm, 25-ball WLCSP
- Operating temperature range: −40 °C to +125 °C

## APPLICATIONS

- Optical networking
- Instrumentation
- Data acquisition
- Automatic test equipment
- Process control and industrial automation

## GENERAL DESCRIPTION

The AD5710R features eight DAC channels that can be independently configured as either a current output digital-to-analog converter (IDAC) or a voltage output digital-to-analog converter (VDAC). The AD5710R includes an integrated 2.5V, 3ppm/°C reference.

When the AD5710R outputs are configured as an IDAC, the output current range is 0mA to 50mA. When configured as a VDAC, the output voltage range is 0V to VREF or 0V to 2 × VREF.

The AD5710R is available in a 25-ball wafer level chip scale packaging (WLCSP) and is specified over the −40°C to +125°C temperature range.

## FUNCTIONAL BLOCK DIAGRAM

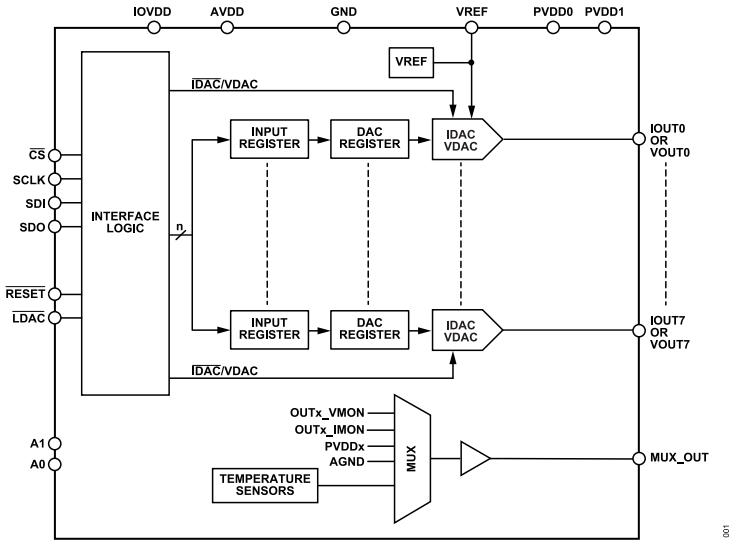


*Figure 1. Functional Block Diagram*

Rev. 0

DOCUMENT FEEDBACK

TECHNICAL SUPPORT

## TABLE OF CONTENTS



## REVISION HISTORY

**10/2025—Revision 0: Initial Version**

## ELECTRICAL CHARACTERISTICS

### DC SPECIFICATIONS

AVDD = 2.7V to 5.5V, PVDDx = 2.7V to AVDD for VDAC channels, PVDDx = 1.71V to AVDD for IDAC channels, IOVDD = 1.08V to 1.98V, VREF = 2.5V (internal or external), all specifications are at $T_J$ = −40°C to +125°C. Typical specifications at $T_A$ = 25°C, unless otherwise noted. $R_L$ = 2kΩ, $C_L$ = 200pF for VDAC channels. $R_L$ = 50Ω for IDAC channels. No capacitor on VREF.

*Table 1. DC Specifications*

| Parameter | Min | Typ | Max | Unit | Test Conditions/Comments |
|---|---|---|---|---|---|
| VDAC STATIC PERFORMANCE | | | | | |
| Resolution | 16 | | | Bits | |
| Integral Nonlinearity (INL)[1] | −10 | | +10 | $LSB_{16}$ | |
| Differential Nonlinearity (DNL)[1] | −1 | | +1 | $LSB_{16}$ | Guaranteed monotonic |
| Zero-Code Error | −4.6 | 1.5 | +4.6 | mV | |
| Offset Error[2] | −5.2 | 1.4 | +5.2 | mV | |
| Gain Error | −0.3 | 0.02 | +0.3 | % of FSR | Range = 0 to VREF |
| | | 0.04 | | % of FSR | Range = 0 to 2 × VREF |
| Total Unadjusted Error (TUE) | −0.34 | ±0.08 | +0.34 | % of FSR | Range = 0 to VREF |
| | −0.16 | ±0.04 | 0.16 | % of FSR | Range = 0 to 2 × VREF |
| Offset Error Drift | | 1.5 | | µV/°C | Range = 0 to VREF |
| | | 2.5 | | µV/°C | Range = 0 to 2 × VREF |
| Gain Error Drift | | 18 | | ppm/°C | Range = 0 to VREF |
| | | 25 | | ppm/°C | Range = 0 to 2 × VREF |
| DC Power Supply Rejection Ratio (PSRR) | | | | | |
| AVDD | | 33 | | µV/V | DAC code = midscale, AVDD = 5V ± 10% |
| PVDD | | 16 | | µV/V | DAC code = midscale, PVDD = 5V ± 10% |
| DC Crosstalk | | | | | |
| VDAC to VDAC | | 4.5 | | µV | Due to single channel, full-scale output change, internal reference, and range = 0 to VREF |
| | | 35 | | µV/mA | Due to single channel, from −10mA to +10mA load current change, internal reference, and range = 0 to VREF |
| | | 14 | | µV | Due to powering down (per channel), internal reference, and range = 0 to VREF |
| IDAC to VDAC | | 168 | | µV | Due to single channel, full-scale output change, internal reference, and range = 0 to VREF |
| | | 298 | | µV | Due to powering down (per channel), internal reference, and range = 0 to VREF |
| VDAC OUTPUT CHARACTERISTICS | | | | | |
| Output Voltage Range | 0 | | VREF | V | Range = 0 to VREF |
| | 0 | | 2 × VREF | V | Range = 0 to 2 × VREF |
| Short Circuit Current | | 50 | | mA | Sourcing |
| | | 10 | | mA | Sinking |
| Capacitive Load Stability | | 2 | | nF | $R_L$ = ∞ |
| | | 10 | | nF | $R_L$ ≤ 2kΩ |
| Load Regulation | | 200 | | µV/mA | PVDD = 5V ± 10%, DAC code = midscale, −30mA ≤ $I_{OUT}$ ≤ +30mA |
| | | 200 | | µV/mA | PVDD = 3V ± 10%, DAC code = midscale, −20mA ≤ $I_{OUT}$ ≤ +20mA |
| Headroom | 70 | 50 | | mV | Source current = 20mA |
| Footroom | 250 | 120 | | mV | Sink current = 10mA |
| Output Impedance | | 10 | | Ω | Output close to VDD or GND |
| Power-Up Time | | 4.5 | | µs | Exiting power-down mode, AVDD = PVDD = 5V |

## ELECTRICAL CHARACTERISTICS

*Table 1. DC Specifications (Continued)*

| Parameter | Min | Typ | Max | Unit | Test Conditions/Comments |
|---|---|---|---|---|---|
| IDAC STATIC PERFORMANCE | | | | | |
| Resolution | 16 | | | Bits | |
| INL[1] | | ±8 | | $LSB_{16}$ | |
| DNL[1] | −1 | | +1 | $LSB_{16}$ | Guaranteed monotonic |
| Offset Error[2] | | 82 | | µA | |
| Gain Error | | 0.1 | | % of FSR | |
| Offset Error Drift | | 0.14 | | µA/°C | |
| Gain Error Drift | | 20 | | ppm/°C | |
| DC PSRR | | | | | |
| AVDD | | 1.2 | | µA/V | DAC code = midscale, AVDD = 5V ± 10% |
| | | 1.5 | | µA/V | DAC code = midscale, AVDD = 3.3V ± 10% |
| PVDD | | 23 | | µA/V | DAC code = midscale, PVDD = 5V ± 10% |
| | | 12.5 | | µA/V | DAC code = midscale, PVDD = 1.8V ± 10% |
| DC Crosstalk | | | | | |
| IDAC to IDAC | | 3 | | µA | Due to single channel, full-scale output change, and internal reference |
| | | 5 | | µA | Due to powering down (per channel), and internal reference |
| VDAC to IDAC | | 0.1 | | µA | Due to single channel, full-scale output change, and internal reference |
| | | 0.5 | | µA | Due to powering down (per channel), internal reference, and VDAC range = 0 to VREF |
| IDAC OUTPUT CHARACTERISTICS | | | | | |
| Output Current Range | 0 | | >50 | mA | |
| Dropout Voltage | | | 250 | mV | Output current = FS |
| Power-Up Time | | 4 | | µs | Exiting power-down mode, AVDD = PVDD = 5V |
| REFERENCE INPUT | | | | | |
| Reference Input Current | | 387 | | µA | VREF = 5.5V |
| | | 682 | | µA | VREF = 5.5V and VDAC range = 0 to 2 × VREF |
| Reference Input Voltage | | 2.5 | | V | |
| Reference Input Impedance | | 14.3 | | kΩ | |
| | | 8.1 | | kΩ | VDAC range = 0 to 2 × VREF |
| REFERENCE OUTPUT | | | | | |
| VREF Output | 2.4925 | 2.5 | 2.5075 | V | |
| Voltage Reference Temperature Coefficient (TC) | | 3 | | ppm/°C | |
| Output Impedance | | 0.04 | | Ω | |
| Load Regulation Sourcing | | 40 | | µV/mA | At ambient temperature |
| Output Current Load Capability | | +7 | | mA | Sourcing, AVDD ≥ 3V |
| | | 400 | | µA | Sinking |
| Line Regulation | | 13.4 | | µV/V | At ambient temperature |
| Thermal Hysteresis | | 2.6 | | ppm | First cycle |
| | | 2.7 | | ppm | Additional cycles |
| INTEGRATED MULTIPLEXER | | | | | |
| Output Voltage Range | 0 | | PVDD | V | |
| Output Current | | ±9 | | mA | |
| Output Impedance | | 0.54 | | Ω | |
| Offset Error | | 50 | | mV | |
| Maximum Capacitive Load | | 10 | | nF | |

## ELECTRICAL CHARACTERISTICS

*Table 1. DC Specifications (Continued)*

| Parameter | Min | Typ | Max | Unit | Test Conditions/Comments |
|---|---|---|---|---|---|
| LOGIC INPUTS | | | | | |
| Input Current | | | 1 | µA | Per pin, leakage current |
| Input Low Voltage ($V_{IL}$) | | | 0.3 × IOVDD | V | |
| Input High Voltage ($V_{IH}$) | 0.7 × IOVDD | | | V | |
| Pin Capacitance | | 1.3 | | pF | |
| LOGIC OUTPUTS SERIAL DATA OUT (SDO) | | | | | |
| Output Low Voltage ($V_{OL}$) | | | 0.4 | V | $I_{SINK}$ = 1µA |
| Output High Voltage ($V_{OH}$) | IOVDD − 0.4 | | | V | $I_{SOURCE}$ = 1µA |
| Floating State Output Capacitance | | 1.2 | | pF | |
| POWER REQUIREMENTS | | | | | |
| IOVDD | 1.08 | | 1.98 | V | |
| $I_{LOGIC}$ | | 16 | | µA | Static Inputs, $V_{IH}$ = IOVDD, $V_{IL}$ = GND |
| AVDD | 2.7 | | 5.5 | V | |
| PVDDx | 1.71 | | AVDD | V | |
| $I_{AVDD}$ | | | | | |
| Normal Mode | | 1.75 | | mA | External reference = 2.5V |
| | | 2.64 | 2.76 | mA | Internal reference |
| Power-Down | | 1.05 | 1.8 | mA | Internal reference |
| | | 690 | | µA | External reference = 2.5V |
| $I_{PVDD0}$, $I_{PVDD1}$ | | 2.3 | | mA | IDAC output at zero scale |
| | | 0.8 | | µA | IDAC output = high-Z |
| | | 0.55 | | mA | VDAC output at zero scale |
| | | 0.35 | | µA | VDAC output = 15kΩ to GND |

[1] Linearity is defined from code 256 to code 65535 for VDAC and IDAC channels.

[2] Offset error is measured at code 6503 for VDAC channels and code 255 for IDAC channels.

## AC SPECIFICATIONS

AVDD = 2.7V to 5.5 V, PVDDx = 2.7V to AVDD for VDAC channels, PVDDx = 1.71V to AVDD for IDAC channels, 1.08V ≤ IOVDD ≤ 1.98V, VREF = 2.5V (internal or external), and all specifications are at $T_J$ = −40°C to +125°C. Typical at $T_A$ = 25°C, unless otherwise noted. $R_L$ = 2kΩ to GND, $C_L$ = 200pF to GND for VDAC channels. $R_L$ = 50Ω to GND for IDAC channels. No capacitor on VREF.

*Table 2. AC Specifications*

| Parameter | Min | Typ | Max | Unit | Test Conditions/Comments |
|---|---|---|---|---|---|
| VDAC SPECIFICATIONS | | | | | |
| Output Voltage Settling Time | | 7 | | µs | ¼ to ¾ scale and vice versa with settling to ±2 $LSB_{16}$ |
| Slew Rate | | 0.7 | | V/µs | 10% FSR to 90% FSR and vice versa |
| Digital-to-Analog Glitch Impulse | | 1 | | nV-sec | 1 $LSB_{16}$ change around major carry (internal reference, range = 0 to VREF) |
| Digital-to-Analog Glitch Peak | | 1.5 | | mV | |
| Digital Feedthrough (Voltage Peak) | | | | | Internal reference and range = 0 to VREF |
| | | 0.04 | | mV | OUT1, OUT2 , OUT3, OUT5, OUT6, and OUT7 as victim channels |
| | | 0.7 | | mV | OUT0 and OUT4 as victim channels |
| VDAC-to-VDAC Crosstalk (Voltage Peak)[1] | | | | | Internal reference and range = 0 to VREF |
| Digital | | 0.05 | | mV | OUT1, OUT2 , OUT3, OUT5, OUT6, and OUT7 as victim channels |
| | | 0.7 | | mV | OUT0 and OUT4 as victim channels |
| Analog | | 1.1 | | mV | Adjacent channels[2] |
| | | 0.2 | | mV | Non adjacent channels[2] |

## ELECTRICAL CHARACTERISTICS

*Table 2. AC Specifications (Continued)*

| Parameter | Min | Typ | Max | Unit | Test Conditions/Comments |
|---|---|---|---|---|---|
| DAC-to-DAC | | 1.1 | | mV | OUT1 and OUT5 as victim channels, with OUT0 and OUT4 as attackers respectively |
| | | 0.3 | | mV | OUT0, OUT2 , OUT3, OUT4, OUT6, and OUT7 as victim channels |
| IDAC-to-VDAC Crosstalk (Voltage Peak)[3] | | | | | Internal reference and range = 0 to VREF |
| Analog | | 5 | | mV | Adjacent channels[2] |
| | | 0.2 | | mV | Non adjacent channels[2] |
| DAC-to-DAC | | 3 | | mV | Adjacent channels[2] |
| | | 0.2 | | mV | Non adjacent channels[2] |
| Mux-to-VDAC Crosstalk[4] | | 0.09 | | mV | OUT1, OUT2 , OUT3, OUT4, OUT5, and OUT7 as victim channels |
| | | 1.3 | | mV | OUT0 and OUT4 as victim channels |
| Output Noise Spectral Density | | | | | DAC code = midscale, 10kHz |
| | | 68 | | nV/√Hz | Range = 0 to VREF and external reference |
| | | 84 | | nV/√Hz | Range = 0 to 2 × VREF and external reference |
| | | 107 | | nV/√Hz | Range = 0 to VREF and internal reference |
| | | 131 | | nV/√Hz | Range = 0 to 2 × VREF and ixternal reference |
| Output Noise | | 15 | | µV p-p | 0.1Hz to 10Hz, range = 0 to VREF |
| | | 25 | | µV p-p | 0.1Hz to 10Hz, range = 0 to 2 × VREF |
| PVDDx AC PSRR | | -93 | | dB | 10Hz, range = 0 to VREF or range = 0 to 2 × VREF |
| | | -79 | | dB | 100Hz, range = 0 to VREF or range = 0 to 2 × VREF |
| | | -59 | | dB | 1kHz, range = 0 to VREF or range = 0 to 2 × VREF |
| IDAC SPECIFICATIONS | | | | | |
| Output Current Settling Time | | 17.5 | | µs | ¼ to ¾ scale with settling to ±2 $LSB_{16}$ |
| Slew Rate | | 18 | | mA/µs | 10% FSR to 90% FSR and vice versa |
| Digital-to-Analog Glitch Impulse | | 15 | | pA-sec | 1 $LSB_{16}$ change around major carry |
| Digital-to-Analog Glitch Peak | | 40 | | µA | |
| Digital Feedthrough (Current Peak) | | | | | Internal reference and range = 0 to VREF |
| | | 2.5 | | µA | OUT1, OUT2 , OUT3, OUT5, OUT6, and OUT7 as victim channels |
| | | 35 | | µA | OUT0 and OUT4 as victim channels |
| IDAC-to-IDAC Crosstalk (Current Peak) | | | | | Internal reference and range = 0 to VREF |
| Digital | | 2.4 | | µA | OUT1, OUT2 , OUT3, OUT5, OUT6, and OUT7 as victim channels |
| | | 35 | | µA | OUT0 and OUT4 as victim channels |
| Analog | | 130 | | µA | OUT1 and OUT5 as victim channels, with OUT0 and OUT4 as attackers respectively |
| | | 12 | | µA | OUT0, OUT2 , OUT3, OUT4, OUT6, and OUT7 as victim channels |
| DAC-to-DAC | | 9.4 | | µA | OUT0, OUT2 , OUT3, OUT4, OUT6, and OUT7 as victim channels |
| | | 65 | | µA | OUT1 and OUT5 as victim channels, with OUT0 and OUT4 as attackers respectively |
| VDAC-to-IDAC Crosstalk (Current Peak) | | | | | Internal reference and range = 0 to VREF |
| Analog | | 6.5 | | µA | OUT0, OUT2 , OUT3, OUT4, OUT6, and OUT7 as victim channels |
| | | 20.5 | | µA | OUT1 and OUT5 as victim channels, with OUT0 and OUT4 as attackers respectively |
| DAC-to-DAC | | 1.3 | | µA | OUT0, OUT2 , OUT3, OUT4, OUT6, and OUT7 as victim channels |
| | | 18.6 | | µA | OUT1 and OUT5 as victim channels, with OUT0 and OUT4 as attackers respectively |
| Output Noise Spectral Density | | | | | DAC code = midscale |
| | | 4.1 | | nA/√Hz | f = 1kHz and external reference |
| | | 3.1 | | nA/√Hz | f = 10kHz and external reference |
| Output Noise | | 1.15 | | µA p-p | 0.1Hz to 10Hz |
| PVDDx AC PSRR | | -97 | | dB | 100Hz |

## ELECTRICAL CHARACTERISTICS

*Table 2. AC Specifications (Continued)*

| Parameter | Min | Typ | Max | Unit | Test Conditions/Comments |
|---|---|---|---|---|---|
| | | -85 | | dB | 1kHz |
| | | -46 | | dB | 100kHz |
| VOLTAGE REFERENCE OUTPUT | | | | | |
| Output Voltage Noise | | 22 | | µV p-p | 0.1Hz to 10Hz |
| Output Voltage Noise Density | | 93 | | nV/√Hz | f = 10kHz |

[1] Crosstalk measurements are computed as the average of all attacker channels affecting a specific victim, from zero code to full scale transition, or vice versa, on the attacker channel.

[2] Adjacent channels are defined within the same PVDD supply domain. Channels CH0 to CH3 share PVDD0 and are considered adjacent only to each other, and channels CH4 to CH7 share PVDD1 and are adjacent only to each other.

[3] Impact caused by a code change in an IDAC channel on the output of a VDAC channel.

[4] The peak glitch seen on the VDAC channels when any channel is monitored through MUX_OUT_SELECT Register.

## DIGITAL INTERFACE TIMING

All input signals are specified for a standalone device 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. AVDD = 2.7V to 5.5V, 1.08V ≤ IOVDD ≤ 1.98V. VREF = 2.5V. All specifications are at $T_J$ = −40°C to +125°C, unless otherwise noted.

*Table 3. Digital Interface Timing Specifications*

| Parameter | Test Conditions / Comments | Min | Typ | Max | Units |
|---|---|---|---|---|---|
| $t_1$ (Write) | SCLK cycle period write | 20[1] | | | ns |
| $t_1$ (Read) | SCLK cycle period read | 40[2] | | | ns |
| $t_2$ | SCLK high time | | $t_1$ × 0.5 | | ns |
| $t_3$ | SCLK low time | | $t_1$ × 0.5 | | ns |
| $t_4$ | SCLK rising edge to $\overline{CS}$ falling edge | 10 | | | ns |
| $t_5$ | $\overline{CS}$ falling edge to SCLK rising edge setup time | 7 | | | ns |
| $t_6$ | SCLK rising edge to $\overline{CS}$ rising edge, $\overline{LDAC}$ idle high mode | 4 | | | ns |
| $t_7$ | $\overline{CS}$ rising edge to SCLK rising edge | 6 | | | ns |
| $t_8$ | Data hold time | 2 | | | ns |
| $t_9$ | Data setup time | 5 | | | ns |
| $t_{10}$ | $\overline{CS}$ high time (single, combined, or all channel update) | 10 | | | ns |
| $t_{11}$ | SCLK falling edge to SDO data available | | | 9 | ns |
| $t_{12}$ | SCLK falling edge to SDO data remains valid | | | 10 | ns |
| $t_{13}$ | $\overline{CS}$ rising edge to SDO disabled | | | 9 | ns |
| $t_{14}$ | SCLK falling edge to SDO enabled | | | 10 | ns |
| $t_{OUT\_SYNC}$ | Last SCLK rising edge to VOUT transition start (VDAC mode) | | 1.06 | | µs |
| | Last SCLK rising edge to IOUT transition start (IDAC mode) | | 2.41 | | µs |
| $t_{RESET}$ | $\overline{RESET}$ low pulse width | 160 | | | ns |
| $t_{OUT\_RESET}$ | $\overline{RESET}$ falling edge to OUT transition start | | 3.6 | | µs |
| $t_{SPI\_RDY}$ | $\overline{RESET}$ rising edge to serial peripheral interface (SPI) transaction begin | | | 167 | µs |

[1] Equivalent to 50MHz for write operation only.

[2] Equivalent to 25MHz for read operation only.

## ELECTRICAL CHARACTERISTICS

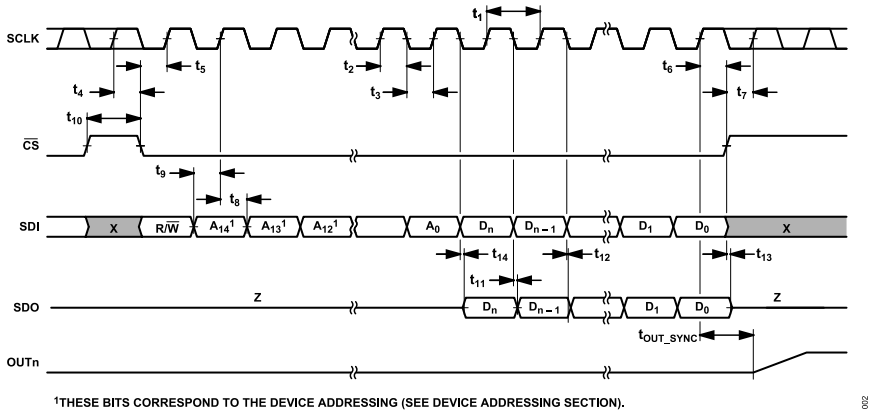


*Figure 2. Serial Read and Write Operation*

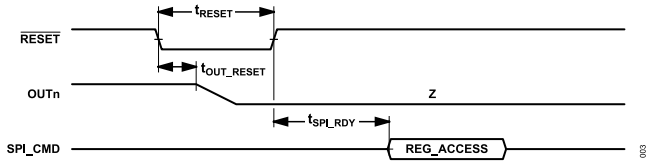


*Figure 3. Reset Timing*

*Table 4. DAC Update Timing Specifications*

| Parameter | Test Conditions / Comments | Min | Typ | Max | Units |
|---|---|---|---|---|---|
| $t_{L1}$ | $\overline{LDAC}$ low pulse width | 120 | | | ns |
| $t_{L2}$ | $\overline{LDAC}$ falling edge to SPI DAC update | 640 | | | ns |
| $t_{L3}$ | SPI DAC update to $\overline{LDAC}$ falling edge | 640 | | | ns |
| $t_{L4}$ | $\overline{LDAC}$ falling edge to VOUT transition (VDAC mode) | | 0.6 | | µs |
| | $\overline{LDAC}$ falling edge to IOUT transition (IDAC mode) | | 2.6 | | µs |

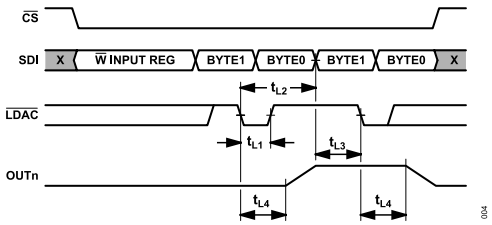


*Figure 4. $\overline{LDAC}$ Operation Timing*

## ABSOLUTE MAXIMUM RATINGS

*Table 5. Absolute Maximum Ratings*

| Parameter | Rating |
|---|---|
| AVDD to GND | −0.3V to +6.5V |
| PVDDx to GND | −0.3V to +6.5V or AVDD + 0.3V, whichever is less |
| IOVDD to GND | −0.3V to +2.1V |
| OUTn to GND | −0.3V to +6.5V or PVDD + 0.3V, whichever is less. |
| VREF[1] to GND | −0.3V to +6.5V or AVDD + 0.3V, whichever is less |
| Digital Input/Output Voltage to GND | −0.3V to +2.1V or IOVDD + 0.3V, whichever is less |
| Temperature | |
| Operating Junction Temperature Range | −40°C to +125°C |
| Storage Temperature Range | −65°C to +150°C |
| Absolute Maximum Junction Temperature | 150°C |
| Reflow Soldering Peak Temperature, Pb-Free (J-STD-020) | 260°C |

[1] Configured as the reference input pin.

Stresses at or above those listed under Absolute Maximum Ratings may cause permanent damage to the product. This is a stress rating only; functional operation of the product at these or any other conditions above those indicated in the operational section of this specification is not implied. Operation beyond the maximum operating conditions for extended periods may affect product reliability.

## THERMAL RESISTANCE

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

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

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

*Table 6. Thermal Resistance*

| Package Type | $\theta_{JA}$ | $\theta_{JB}$ | $\theta_{JC}$ | $\Psi_{JT}$ | $\Psi_{JB}$ | Unit |
|---|---|---|---|---|---|---|
| 25-ball WLCSP[1] | 51.8 | 16 | 2.3 | 3.7 | 16 | °C/W |

[1] Simulation values on JEDEC 2S2P board with 4 thermal vias, still air (0m/sec airflow).

## 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

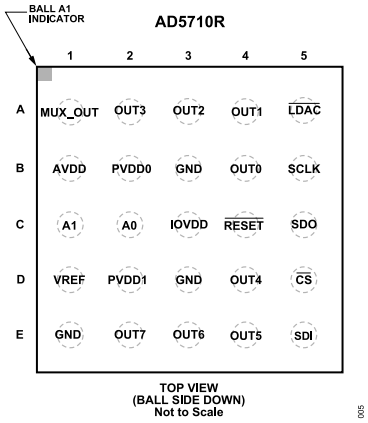


*Figure 5. WLCSP Pin Configuration*

*Table 7. Pin Function Descriptions*

| WLCSP | Mnemonic | Type | Description |
|---|---|---|---|
| D1 | VREF | AI/O | 2.5V Voltage Reference Input/Output. By default, the VREF pin is configured as an input. When the internal reference is enabled, this pin provides a 2.5V reference output.<br>By default, no decoupling capacitor is needed from this pin to GND for specified performance in internal reference mode. |
| A1 | MUX_OUT | AO | Analog Multiplexer Output. This pin is used to monitor internal die temperature, output voltages, and output current of a selected channel. |
| A2 | OUT3 | AO | Analog Voltage Output from VDAC 3/Current Output of IDAC 3. |
| A3 | OUT2 | AO | Analog Voltage Output from VDAC 2/Current Output of IDAC 2. |
| A4 | OUT1 | AO | Analog Voltage Output from VDAC 1/Current Output of IDAC 1. |
| B4 | OUT0 | AO | Analog Voltage Output from VDAC 0/Current Output of IDAC 0. |
| B1 | AVDD | S | Power Supply Input. Bypass to GND with a 0.1µF capacitor. |
| B2 | PVDD0 | S | PVDD Supply Voltage Input for the IDAC/VDAC Channels (OUT0 to OUT3). |
| D2 | PVDD1 | S | PVDD Supply Voltage Input for the IDAC/VDAC Channels (OUT4 to OUT7). |
| D5 | $\overline{CS}$ | DI | Active Low Control Input. This is the frame synchronization signal for the input data. |
| C3 | IOVDD | DI | Logic Power Supply. IOVDD must be between 1.08V and 1.98V. This pin supplies power to the serial interface circuit blocks on the device. Bypass to GND with a 0.1µF capacitor. |
| C5 | SDO | DO | Serial Data Output. A readback operation provides data on this output pin as a serial data stream. Data is clocked out on the falling edge of SCLK and is valid on the rising edge of SCLK. |
| E5 | 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 SCLK. |
| B5 | SCLK | DI | Serial Clock Input. Data transfers at rates of up to 50MHz for write operation and 25MHz for read operation. |
| E1, B3, and D3 | GND | S | Ground Reference Point for All Circuitry on the Device. |
| D4 | OUT4 | AO | Analog Voltage Output from VDAC 4/Current Output of IDAC 4. |
| E4 | OUT5 | AO | Analog Voltage Output from VDAC 5/Current Output of IDAC 5. |
| E3 | OUT6 | AO | Analog Voltage Output from VDAC 6/Current Output of IDAC 6. |
| E2 | OUT7 | AO | Analog Voltage Output from VDAC 7/Current Output of IDAC 7. |
| C4 | $\overline{RESET}$ | DI | Asynchronous Reset. Active low logic input, low level triggered. |
| A5 | $\overline{LDAC}$ | DI | Asynchronous Load DAC. Active low logic input, falling edge sensitive. |
| C2 | A0 | DI | Logic Input: Programmable Address Bit 0. |
| C1 | A1 | DI | Logic Input: Programmable Address Bit 1. |

## TYPICAL PERFORMANCE CHARACTERISTICS

All typical performance characteristics are under the following conditions unless otherwise specified in the figures: AVDD = 5V, PVDD = 5V, VLOGIC = 1.8V, VREF = 2.5V internal, and $T_A$ = 25°C. For VDAC channels $R_L$ = 2kΩ to GND, $C_L$ = 200pF to GND. For IDAC channels $R_L$ = 50Ω to GND.

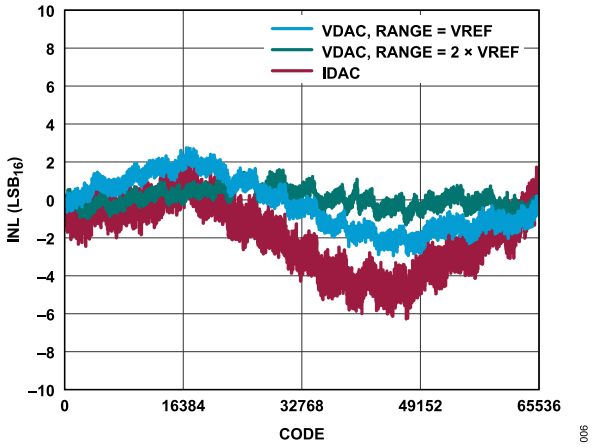


Figure 6. INL Error vs. Code

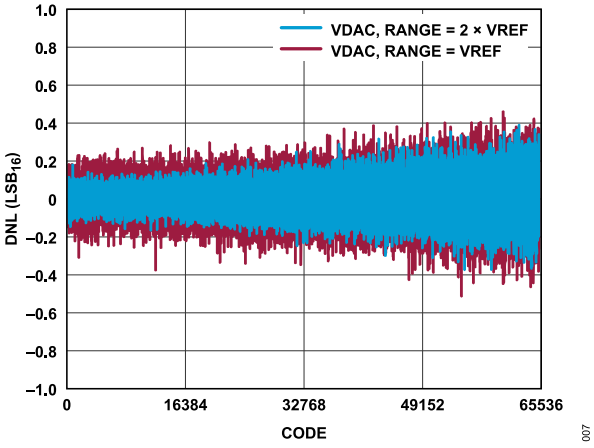


Figure 7. DNL Error vs. Code

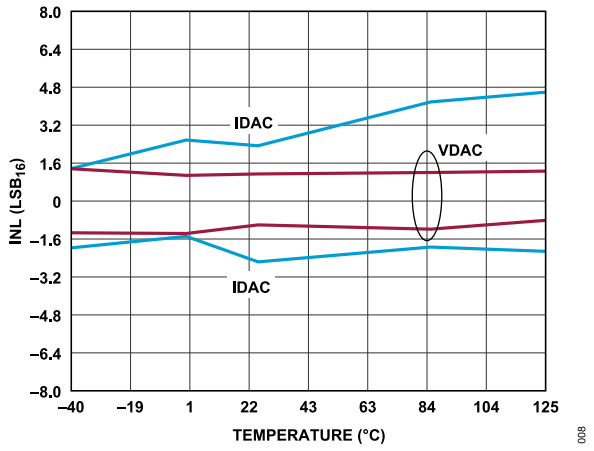


Figure 8. INL Error vs. Temperature

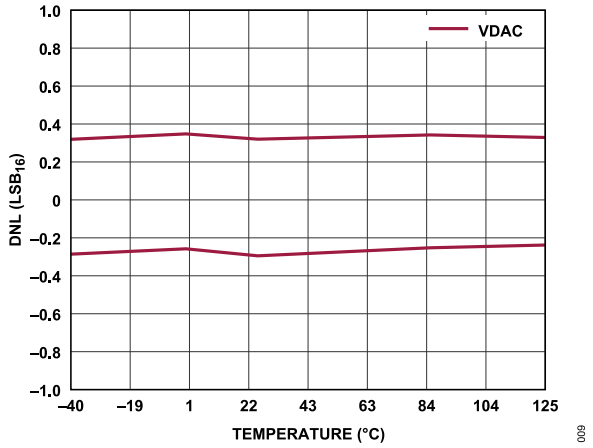


Figure 9. DNL Error vs. Temperature

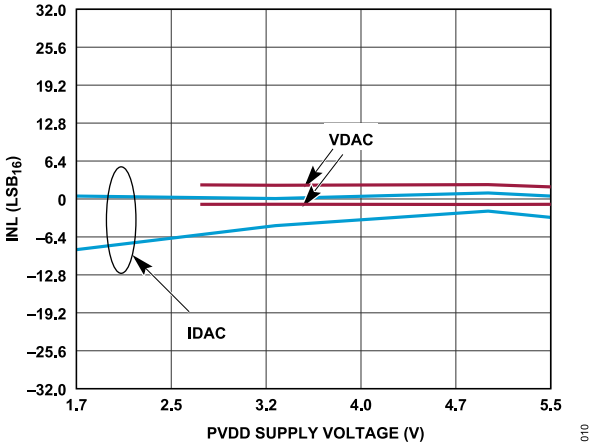


Figure 10. INL Error vs. Supply Voltage

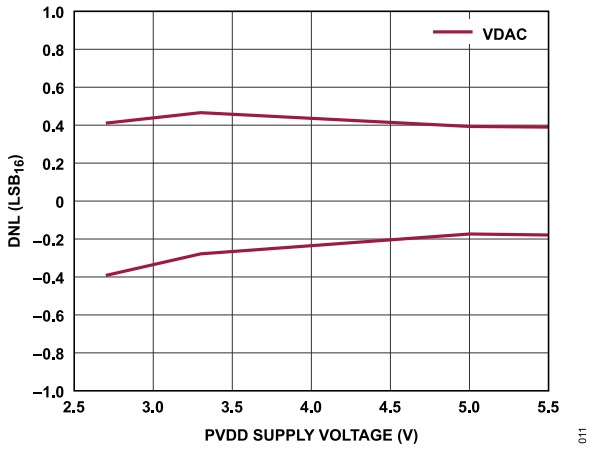


Figure 11. DNL Error vs. Supply Voltage

## TYPICAL PERFORMANCE CHARACTERISTICS

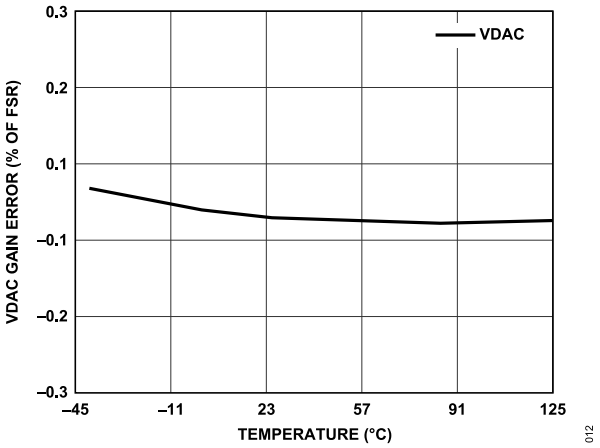


*Figure 12. Gain Error vs. Temperature*

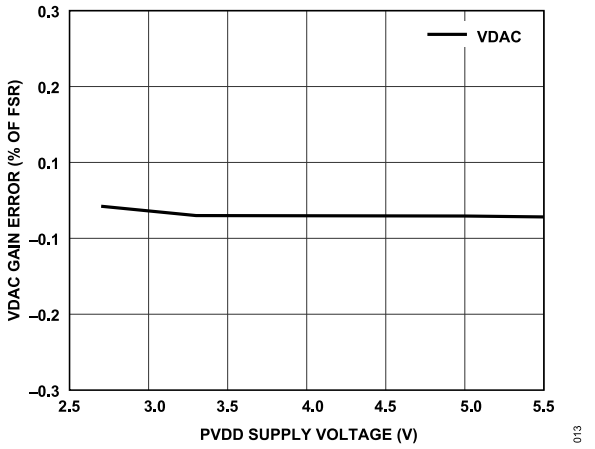


*Figure 13. Gain Error vs. Supply Voltage*

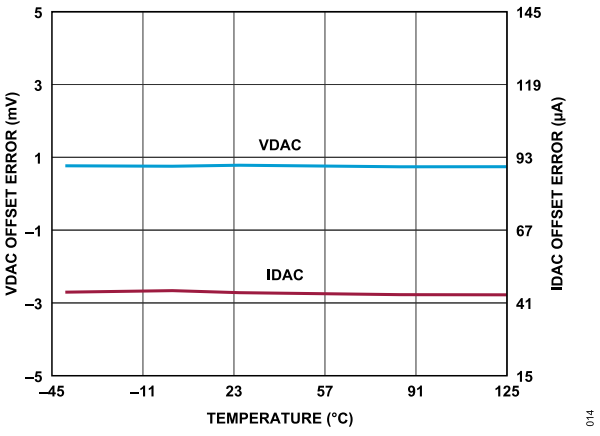


*Figure 14. Offset Error vs. Temperature*

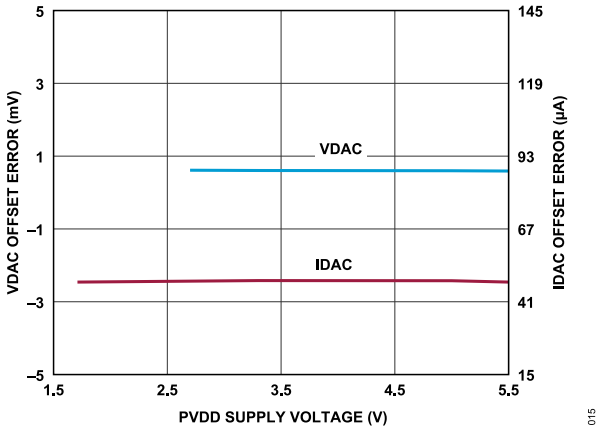


*Figure 15. Offset Error vs. Supply Voltage*

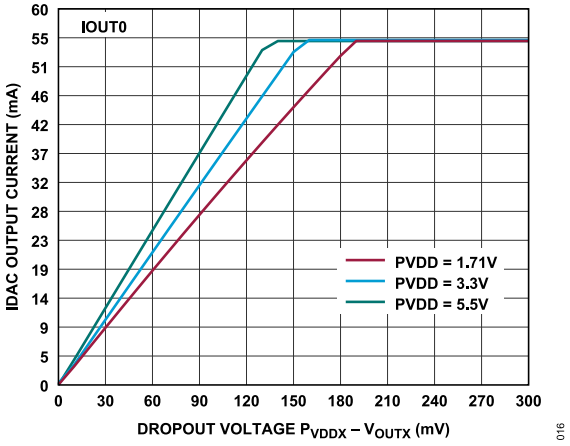


*Figure 16. Dropout Across Different PVDD Supply Ranges*

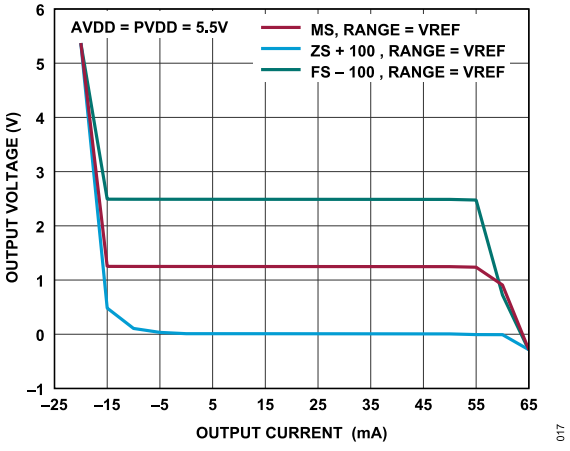


*Figure 17. VDAC Source and Sink Capability at RANGE = VREF*

## TYPICAL PERFORMANCE CHARACTERISTICS

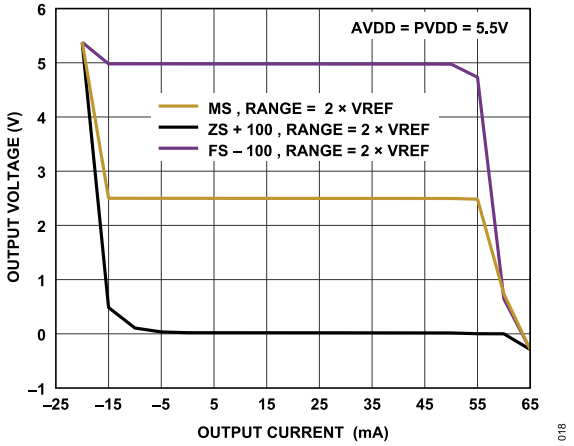


*Figure 18. VDAC Source and Sink Capability at RANGE = 2 × VREF*

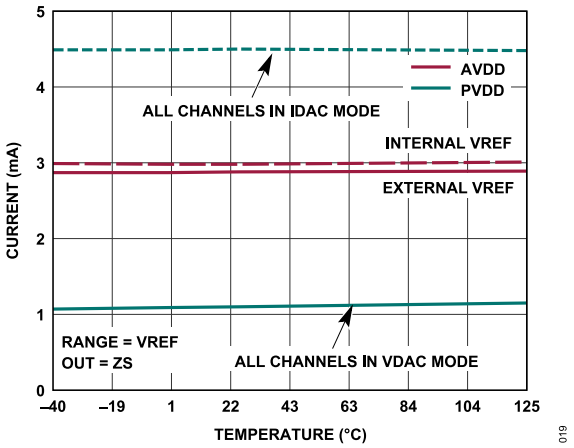


*Figure 19. $I_{DD}$ vs. Temperature*

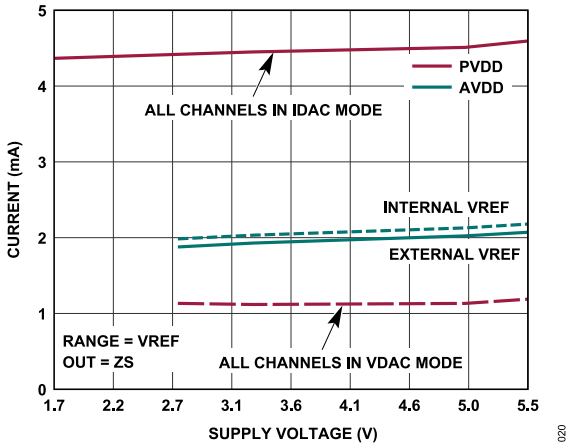


*Figure 20. $I_{DD}$ vs. Supply Voltage*

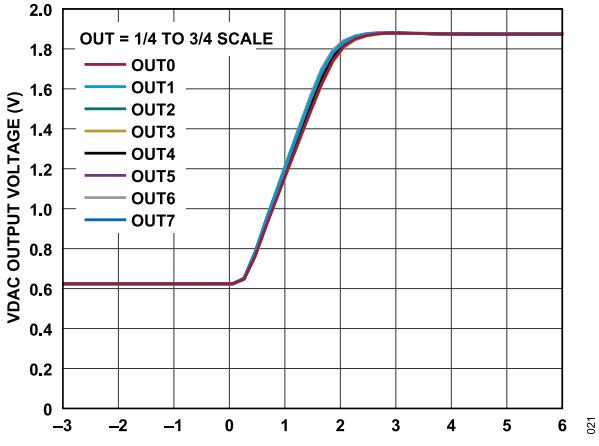


*Figure 21. VDAC Output Settling Time*

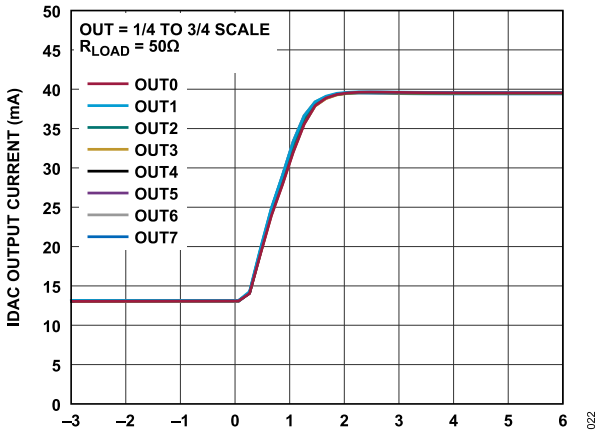


*Figure 22. IDAC Output Settling Time*

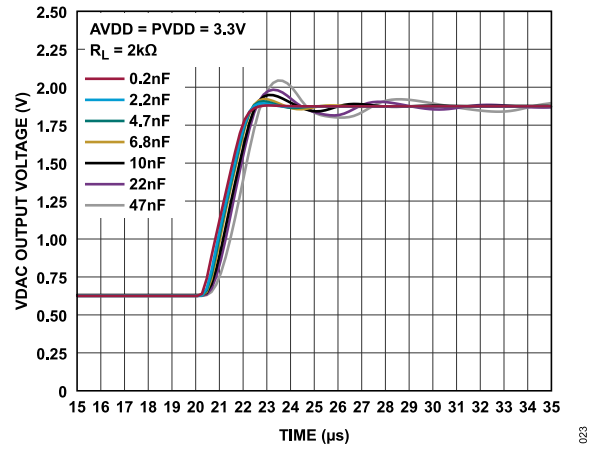


*Figure 23. VDAC Output Settling Time at Various Capacitive Loads*

## TYPICAL PERFORMANCE CHARACTERISTICS

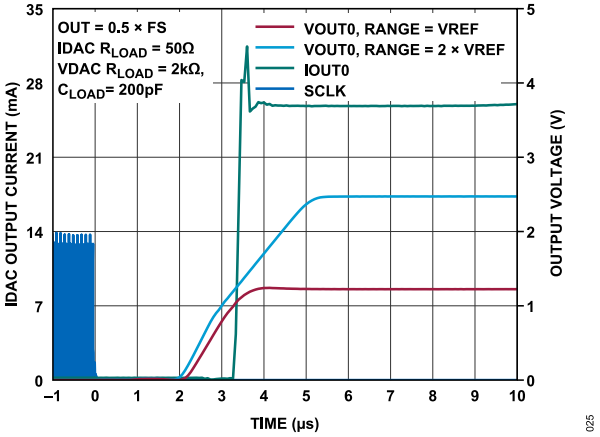


*Figure 24. IDAC and VDAC Channel Exiting Power-Down to Midscale*

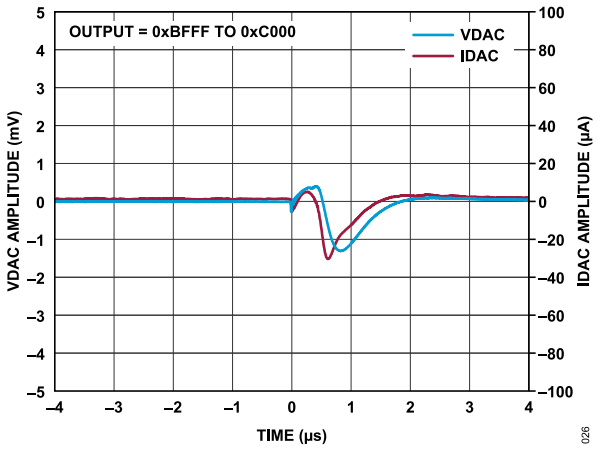


*Figure 25. Digital-to-Analog Glitch Impulse Ascending*

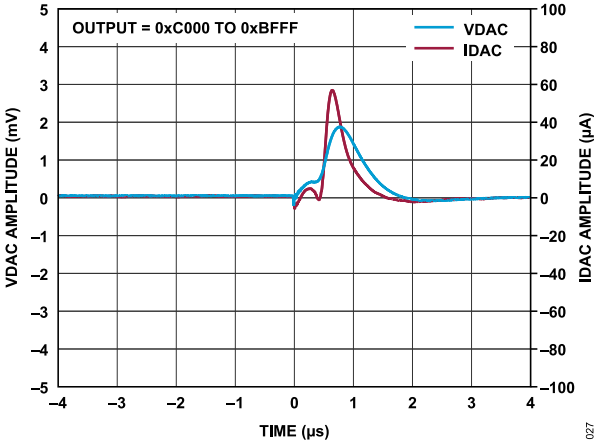


*Figure 26. Digital-to-Analog Glitch Impulse Descending*

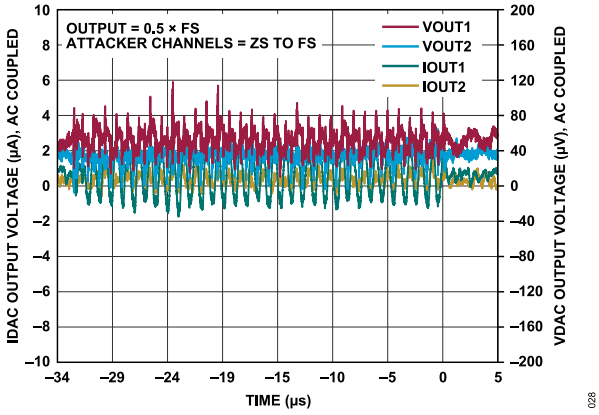


*Figure 27. IDAC and VDAC Digital Feedthrough, OUT1 and OUT2*

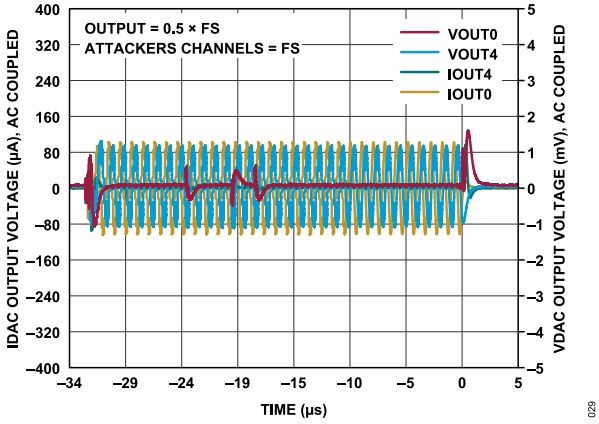


*Figure 28. IDAC and VDAC Digital Feedthrough, OUT0 and OUT4*

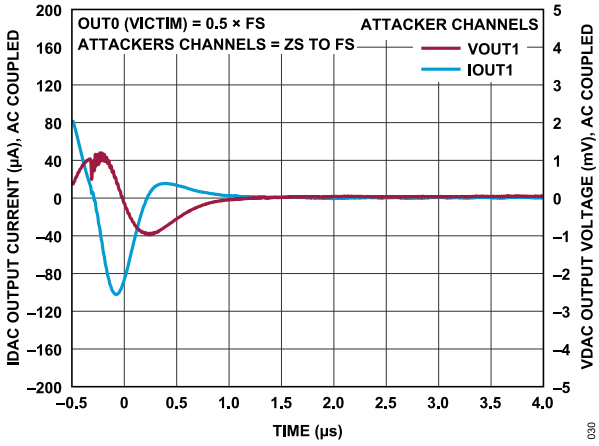


*Figure 29. IDAC and VDAC Digital Crosstalk, OUT0*

## TYPICAL PERFORMANCE CHARACTERISTICS

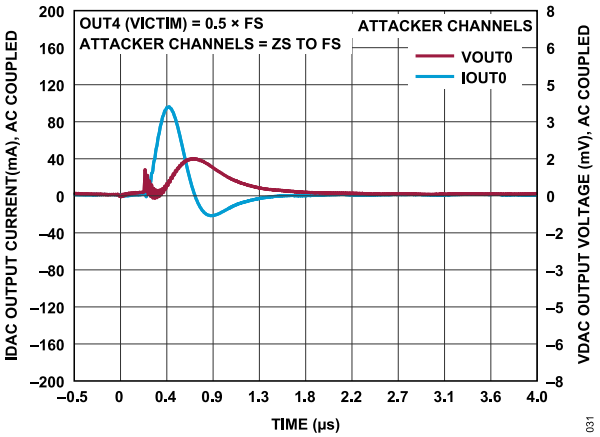


Figure 30. IDAC and VDAC Digital Crosstalk, OUT4

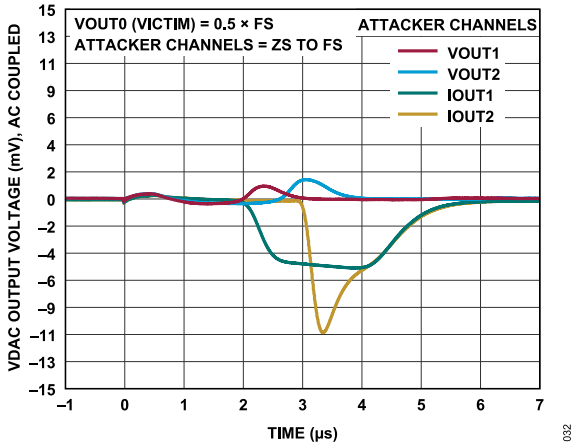


Figure 31. VDAC Analog Crosstalk, OUT0

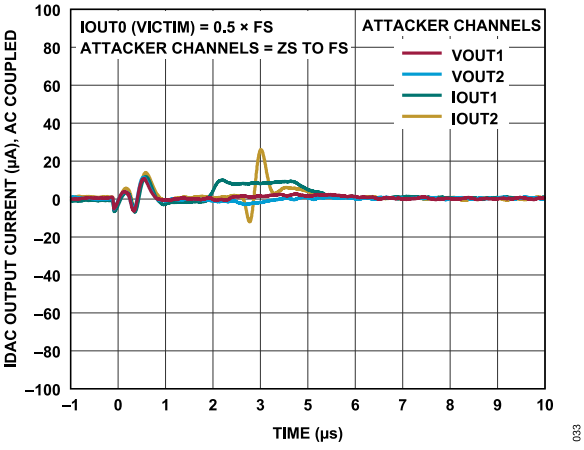


Figure 32. IDAC Analog Crosstalk, OUT0

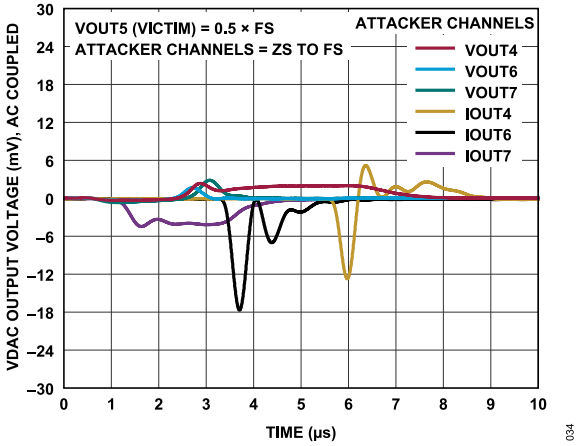


Figure 33. VDAC Analog Crosstalk, OUT5

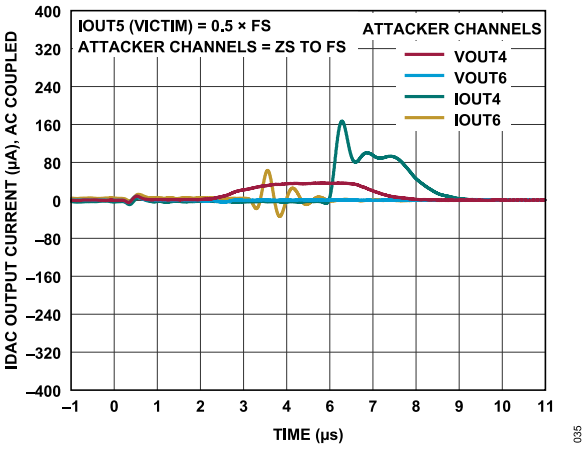


Figure 34. IDAC Analog Crosstalk, OUT5

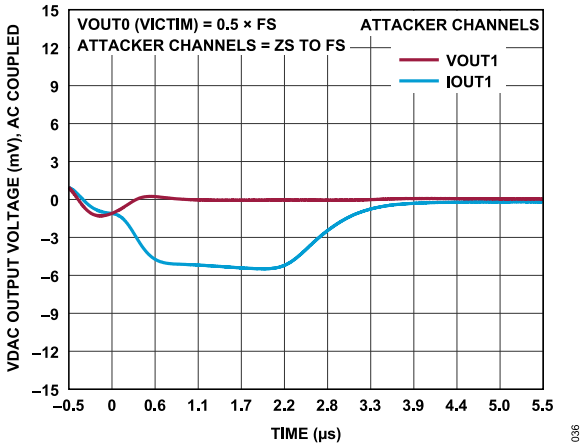


Figure 35. VDAC DAC-to-DAC Crosstalk, OUT0

## TYPICAL PERFORMANCE CHARACTERISTICS

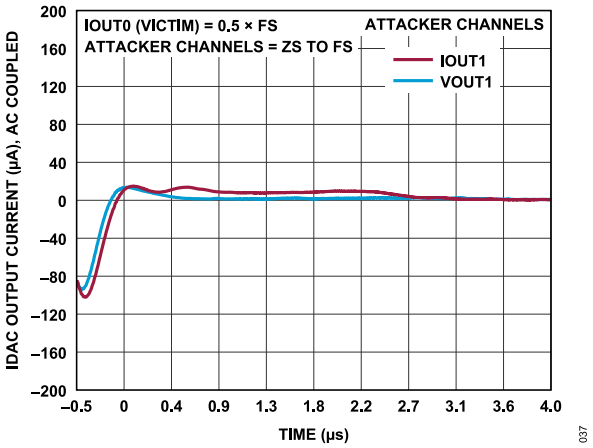


Figure 36. IDAC DAC-to-DAC Crosstalk, OUT0

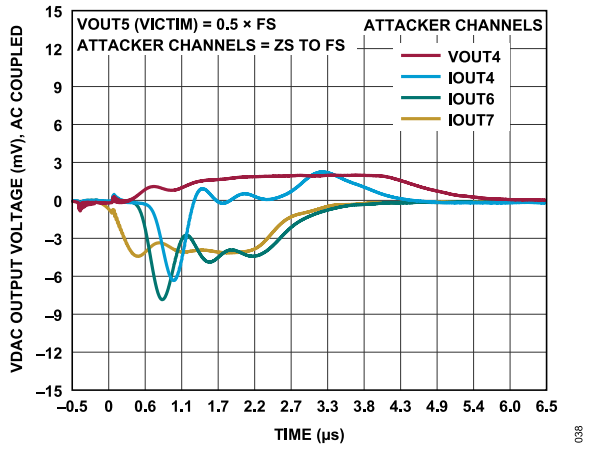


Figure 37. VDAC DAC-to-DAC Crosstalk, OUT5

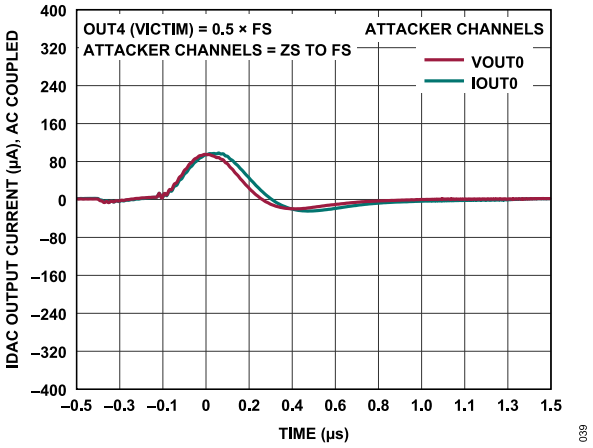


Figure 38. IDAC DAC-to-DAC Crosstalk, OUT4

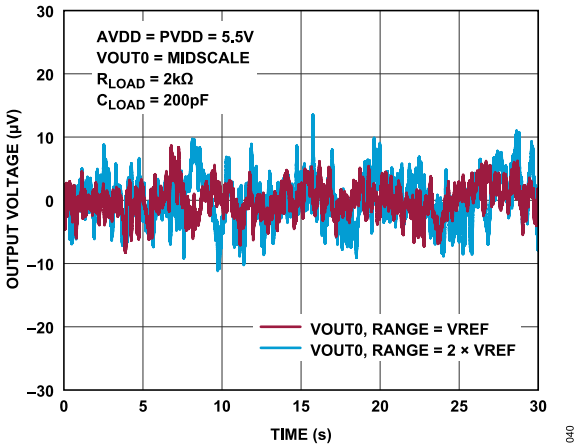


Figure 39. VDAC 0.1Hz to 10Hz (1/f) Noise

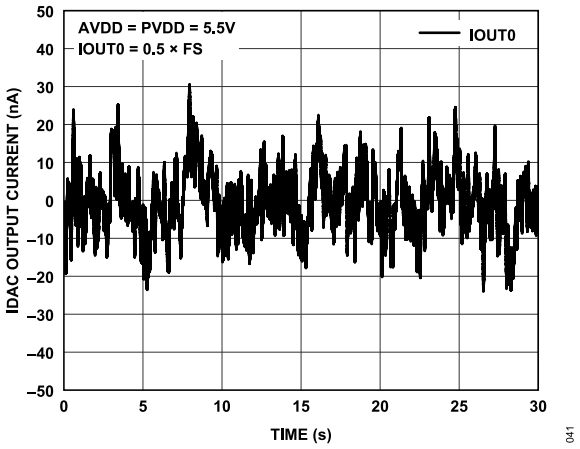


Figure 40. IDAC 0.1Hz to 10Hz (1/f) Noise

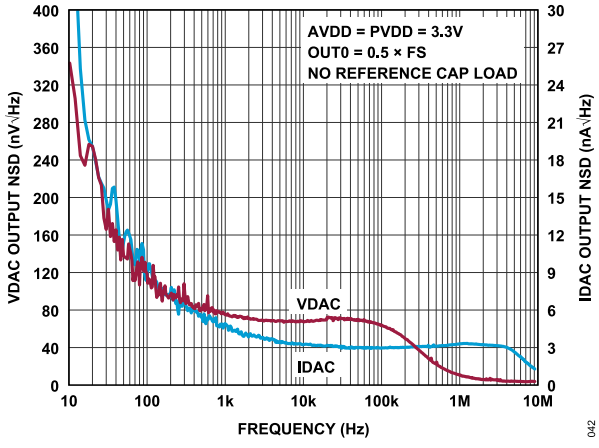


Figure 41. VDAC and IDAC Noise Spectral Density (NSD)

## TYPICAL PERFORMANCE CHARACTERISTICS

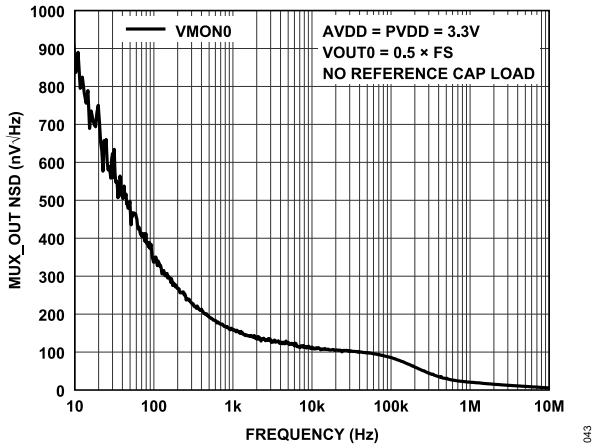


Figure 42. MUX_OUT NSD

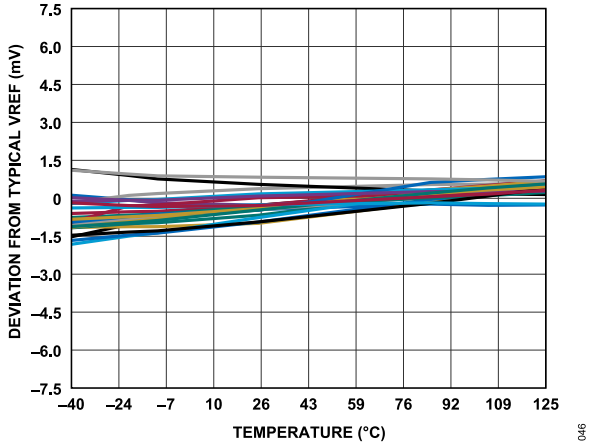


Figure 45. VREF vs. Temperature for 23 Devices

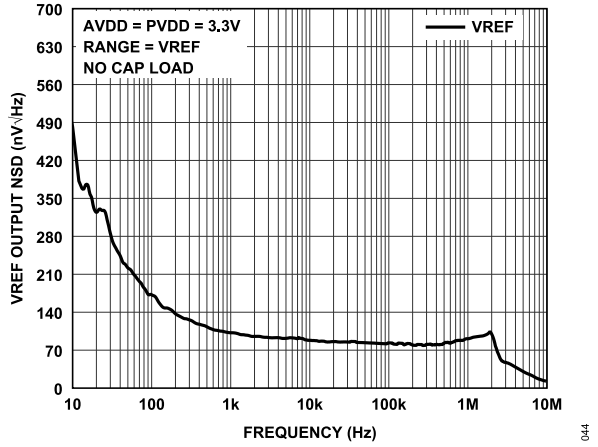


Figure 43. Internal Reference NSD

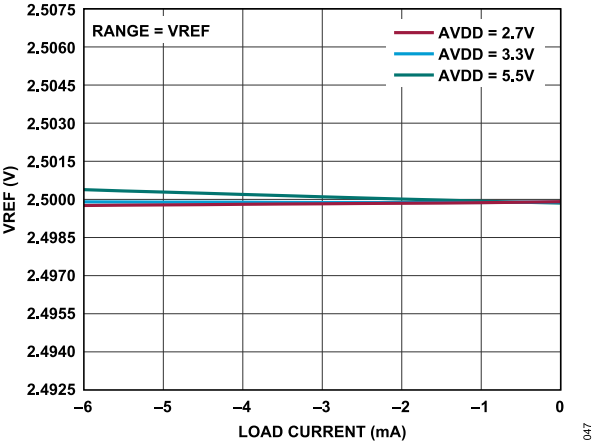


Figure 46. VREF vs. Load Current

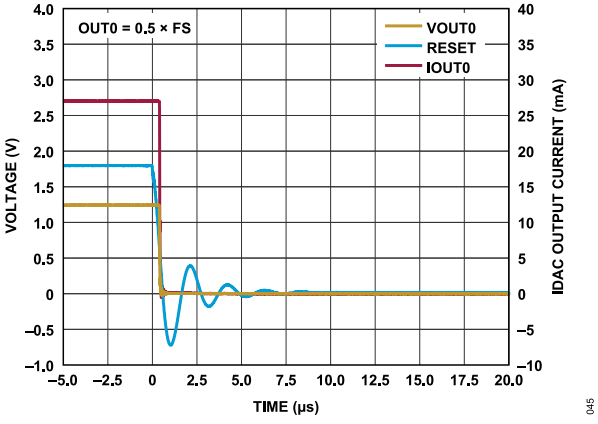


Figure 44. Hardware Reset

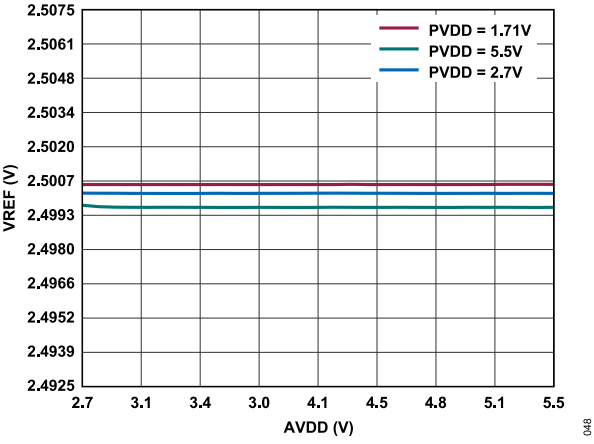


Figure 47. VREF vs. AVDD

## TYPICAL PERFORMANCE CHARACTERISTICS

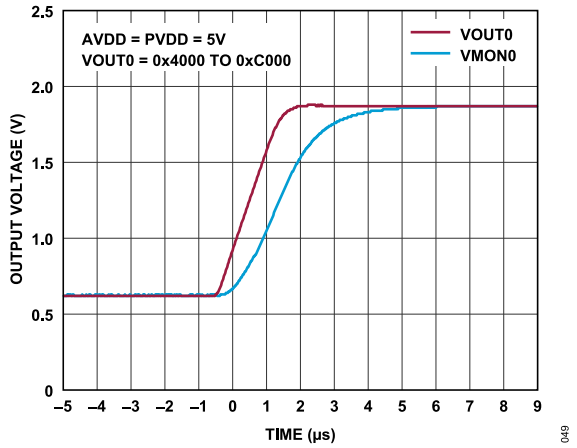


Figure 48. MUX_OUT vs. Output Voltage Transient, Rising

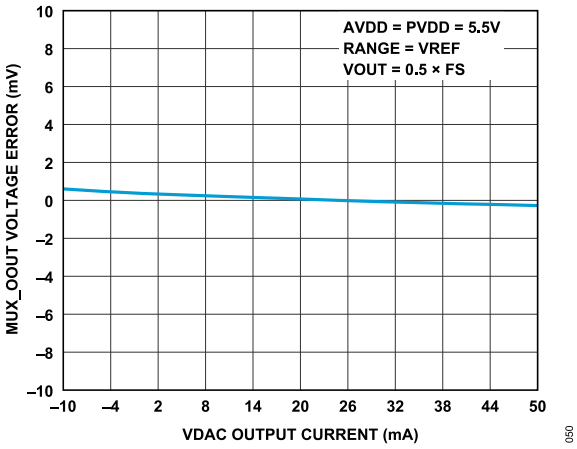


Figure 49. MUX_OUT Error vs. Output Current

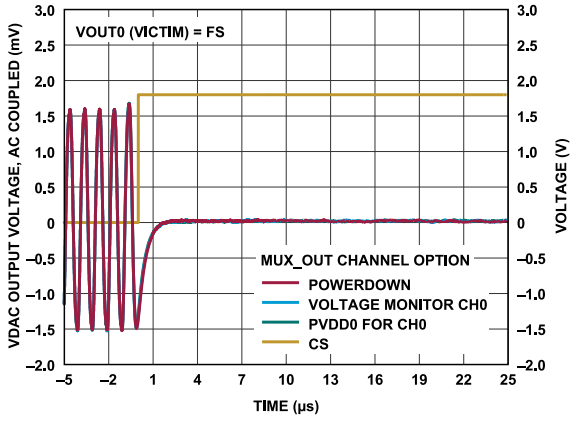


Figure 50. MUX_OUT to VOUTx Glitch

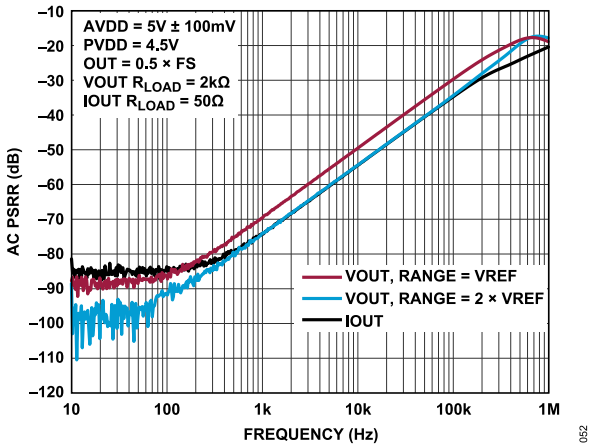


Figure 51. AVDD AC PSRR vs. Frequency

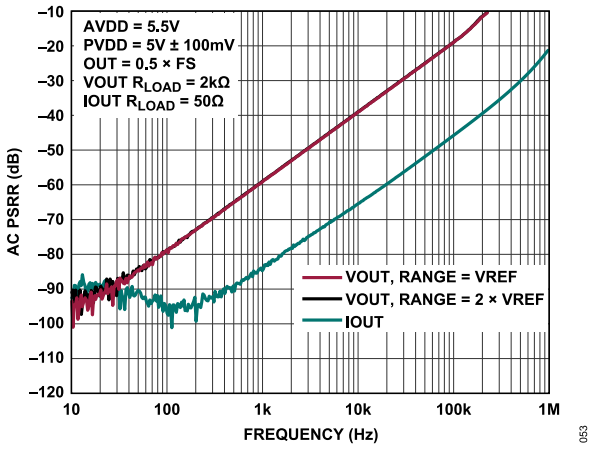


Figure 52. PVDD AC PSRR vs. Frequency

## TERMINOLOGY

### Relative Accuracy or Integral Nonlinearity (INL)

For the DAC, relative accuracy or integral nonlinearity 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)

Differential nonlinearity is the difference between the measured change and the ideal 1 LSB change between any two adjacent codes.

### Offset Error

Offset error is a measure of the difference between the actual output $V_{OUT}$ or $I_{OUT}$ and the ideal output $V_{OUT}$ or $I_{OUT}$ expressed in mV or µA in the linear region of the transfer function. It can be negative or positive.

### Offset Error Drift

Offset error drift is a measurement of the change in offset error with a change in temperature. For a IDAC output, the offset error drift is expressed in µA/°C and for VDAC output is expressed as µV/°C.

### 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 % of FSR.

### Gain Error Drift

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

### Zero-Code Error

Zero-code error measures the deviation from the ideal value at zero scale, at 25°C. The error is expressed in mV for the VOUT channel.

### DC PSRR and AC PSRR

DC PSRR quantifies how variations in the power supply affect the DAC output. It is defined as the ratio of the change in output voltage ($V_{OUT}$) or output current ($I_{OUT}$) to the change in supply voltage, measured at the DAC's midscale output. For $V_{OUTX}$, DC PSRR is expressed in mV/V and for $I_{OUTX}$ it is expressed in µA/V, with PVDDx varied independently by ±10%. The AC PSRR is measured in dB by injecting a ±100mV peak-to-peak AC sweep signal onto PVDDx.

### Output Settling Time

Output voltage settling time is the amount of time it takes for the output of a DAC to settle to a specified level for a given step change.

### Digital-to-Analog Glitch Impulse

Digital-to-analog glitch impulse is the impulse injected into the analog output when the input code in the DAC register changes state. For VDAC, the glitch is normally specified as the area in nV·s and is measured when the digital input code changes by 1 LSB. For IDAC, it is specified as the area in nA·s under the same condition.

### Output Noise Spectral Density

Noise spectral density is a measurement of the internally generated random noise. Noise is measured at the DAC output when it is loaded with the midscale code. For VDAC is measured in nV/$\sqrt{Hz}$ and for IDAC is measured in nA/$\sqrt{Hz}$.

### Total Unadjusted Error (TUE)

Total unadjusted error (TUE) is a measure of the total output error versus the ideal . The Ideal output is based on an exact 5.000V reference. TUE consists of all the various error sources, namely INL error, offset error, and gain error and it is guaranteed over the temperature and supply range.

### Voltage Reference Temperature Coefficient (TC)

Voltage reference TC is a measure of the change in the reference output voltage with a change in temperature. The reference TC is calculated using the box method, which defines the TC as the maximum change in the reference output over a given temperature range expressed in ppm/°C, as shown in the following equation:

$$TC = \left(\frac{V_{REF\_MAX} - V_{REF\_MIN}}{V_{REF\_NOM} \times TEMP\_RANGE}\right) \times 10^6 \quad (1)$$

- where:
- $V_{REF\_MAX}$ is the maximum reference output measured over the total temperature range.
- $V_{REF\_MIN}$ is the minimum reference output measured over the total temperature range.
- $V_{REF\_NOM}$ is the nominal reference output voltage, 2.5V.
- TEMP_RANGE is the specified temperature range, −40°C to +125°C.

### 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. It is measured with a full-scale output change on one DAC while monitoring another DAC kept at midscale. It is expressed in µV for VDAC and µA for IDAC.

### Digital Feedthrough

Digital feedthrough is a measure of the impulse injected into the analog output of the DAC from the digital inputs of the DAC, but it is measured when the DAC output is not updated. For the IDAC, digital feedthrough is specified as the peak output glitch current in µA, measured during a full-scale code change on the data bus, which means from all 0s to all 1s and from all 1s to all 0s. For the VDAC, digital feedthrough is specified as the peak output glitch voltage in µV.

## TERMINOLOGY

### 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 (from 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 current peak current of the output glitch in μA. For a voltage-output DAC, it is specified as the peak voltage of the output glitch in μV. Both are measured on one DAC channel at a time.

### DAC-TO-DAC CROSSTALK

DAC-to-DAC crosstalk refers to the energy of the glitch impulse that is transferred to the output of one DAC when a digital code change is loaded into the DAC register of another channel, resulting in a corresponding analog output change. For a current-output DAC, this crosstalk is measured as the peak of the current output glitch in μA. For a voltage-output DAC, it is measured as the peak of the voltage output glitch in μV. Both measurements are taken during a full-scale change on one DAC output, by writing directly to the DAC register while monitoring the other DAC output, which is held at midscale.

### ANALOG CROSSTALK

Analog crosstalk refers to the glitch impulse transferred to the output of one DAC when a digital code change is loaded into the input register of another channel and its output is updated via an LDAC transition. It is measured by applying a full-scale code change (from all 0s to all 1s and vice versa) to the input register of the active channel, performing an LDAC update, and monitoring the output of the channel whose code remains unchanged. It is expressed as the peak current in μA for a current-output DAC and as the peak voltage in μV for a voltage-output DAC.

## THEORY OF OPERATION

### DIGITAL-TO-ANALOG CONVERTER

The AD5710R is a low power, 8-channel, 16-bit DAC that operates on analog supply voltages of 2.7V to 5.5V for AVDD, 1.71V to AVDD for PVDD0/1 and interface logic supply IOVDD of 1.08V to 1.98V. Each channel can be configured as VDAC or IDAC. A group of four DAC channels, OUT[3:0], is powered by PVDD0 and another group of four DAC channels, OUT[7:4], is powered by PVDD1.

The device has a 3ppm/°C, 2.5V on-chip reference and incorporates various functions such as load DAC, device reset, output monitoring, and die temperature monitoring. A simplified block diagram of a DAC channel is shown in Figure 53.

The AD5710R offers a versatile 4-wire serial interface compatible with classic SPI. See the Digital Interface section for more details.

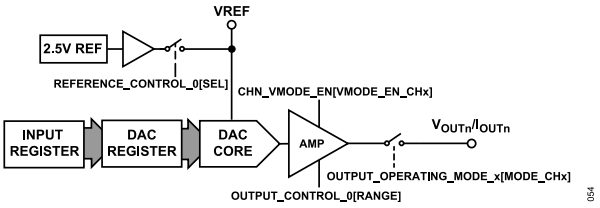


*Figure 53. DAC Channel Block Diagram*

### IDAC Channels

The device integrates eight 16-bit current-output DAC (IDAC) channels. Each channel is configured as an IDAC by default and is capable of sourcing currents up to 50mA.

The input coding to the DAC is straight binary. The conversion of the digital input code to the ideal output voltage is given by the following equation:

$$I_{OUTn} = \left(\frac{D}{2^N}\right) \times 50 \quad mA \tag{2}$$

where:

$I_{OUTn}$ is the output current sourced through the load at the selected DAC channel n.

D is the decimal equivalent of the straight binary code that is loaded into the DAC register.

N is the DAC resolution in bits.

By default, the output of any channel is configured as IDAC in high-Z mode and the output stage of each channel is powered down. Alternatively, each channel can be configured as IDAC output by setting the corresponding CHx_VMODE_EN bit to 0. The output of each channel can be enabled by writing 2'b00 to MODE_CH_x bits corresponding to each channel.

### VDAC Channels

The device contains eight buffered VDAC channels capable of sourcing 50mA and sinking 10mA current independently. VDAC channels have a shared gain bit OUTPUT_CONTROL_0, that sets the output range to 0V to VREF or 0V to 2 × VREF. Consequently, it is not possible to set different output ranges on a per channel basis.

The input coding to the DAC is straight binary. The conversion of the digital input code to the ideal output voltage is given by the following equation:

$$V_{OUTn} = V_{REF} \times \left(\frac{D}{2^N}\right) \times G \tag{3}$$

where:

$V_{OUTn}$ is the output voltage seen at the selected DAC channel n.

$V_{REF}$ is the voltage present on the VREF pin. For internal reference, $V_{REF}$= 2.5V.

D is the decimal equivalent of the straight binary code that is loaded into the DAC register.

N is the DAC resolution in bits.

G is the gain of the output amplifier. G = 1 if OUTPUT_CONTROL_0, Bit 2 (range) = 0 (default), and G = 2 if OUTPUT_CONTROL_0, Bit 2 (range) = 1.

By default, the output of any channel is configured as IDAC in high-Z mode. Each channel can be configured as VDAC output by setting the corresponding CHx_VMODE_EN bit to 1. The DAC output has an effective output resistance of 15kΩ to GND during the shutdown state. The output of each channel can be enabled by writing 2'b00 to MODE_CH_x bits corresponding to each channel.

### Output Voltage Limitation Due to PVDD

When PVDD is configured below the selected VDAC output range, the output voltage of the channel is limited and clamps just below PVDD, regardless of the applied full-scale digital code. In such conditions, the device cannot achieve the full-scale output voltage of 2.5V for an output range of VREF or 5V for an output range of 2 × VREF.

### DAC Operating Modes

Upon power-up or following a power-on reset, all channel outputs default to a high-impedance (high-Z) state while operating in IDAC mode. In this state, the output stage of each channel is powered down. To enable output, the corresponding channel must be configured by writing 2'b00 to the MODE_CH_x bits.

By default, each channel is configured as an IDAC. To switch a channel to VDAC mode, set the VMODE_EN_CHn bit for the respective channel. Prior to transitioning between IDAC and VDAC modes, the DAC register for the corresponding channel must be cleared to 0x0000.

Note that in power-down mode, VDAC channels exhibit an effective internal output resistance of approximately 15 kΩ to GND.

## THEORY OF OPERATION

*Table 8. AD5710R Operating Modes*

| Output State | MODE_CH_n[1:0] |
|---|---|
| Normal operation | 2'b00 |
| Power-down state<br>▶ High-Z state in IDAC mode<br>▶ 15kΩ to GND in VDAC mode | Others |

Entering into the power-down state of any DAC channel does not affect other register settings or the read and write capability of those registers of the respective DAC channel. The input or DAC registers can still be updated but do not reflect on the DAC output pins, except during switching between DAC output modes.

### Recommended Switching Flow: IDAC Mode and VDAC Mode

Figure 54 shows the recommended switching procedures between VDAC and IDAC modes, including their enabling and disabling operations.

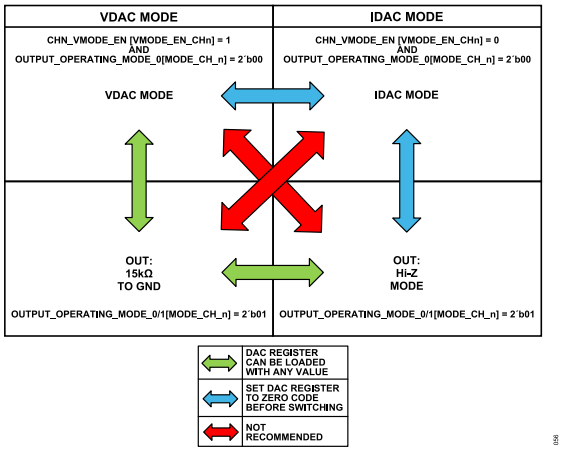


*Figure 54. Recommended Switching Between Operating Modes and VDAC/ IDAC Modes*

## VOLTAGE REFERENCE

The device has an on-chip, buffered, 2.5V, 3ppm/°C reference available at the VREF pin that is capable of sourcing external loads up to +5mA.

By default, upon power-up or after a power-on reset, the VREF pin is configured as an input pin, and an external reference voltage must be provided. The internal reference can be enabled by setting Bit 0 (SEL) in the REFERENCE_CONTROL_0 register to 1. It is recommended to place the external reference IC on the same supply as the AD5710R such that VREF is always less than or equal to AVDD during power-up.

When operating with the internal reference, the device meets specified performance without the need for an external capacitor on the VREF pin.

## INTEGRATED MULTIPLEXER

The device contains a 27:1 analog multiplexer that can output a voltage on the MUX_OUT pin that is a representative of either the output voltage or the output current of a chosen channel, or the internal die temperature of the device. A voltage output represents the full scale range of the monitor channel that can be selected by configuring the SEL bits on the MUX_OUT_SELECT register. An invalid write to the SEL bits is ignored, and the SEL value does not change.

The following signals can be routed to the MUX_OUT output:

- OUTx_VMON output voltages
- OUTx_IMON output through a sense resistor
- Supply rail voltages
- Internal die temperature

By default, the MUX_OUT is powered down. The available channel selections and their corresponding register field values are listed in Table 9.

*Table 9. Multiplexer Channels Options*

| SEL Field | MUX_OUT |
|---|---|
| 0 | Power down (default) |
| 1 | Voltage monitor, CH0 |
| 2 | Current monitor through RSENSE, CH0 |
| 3 | PVDD0 for CH0 |
| 4 | Voltage monitor, CH1 |
| 5 | Current monitor through RSENSE, CH1 |
| 6 | PVDD0 for CH1 |
| 7 | Voltage monitor, CH2 |
| 8 | Current monitor through RSENSE, CH2 |
| 9 | PVDD0 for CH2 |
| 10 | Voltage monitor, CH3 |
| 11 | Current monitor through RSENSE, CH3 |
| 12 | PVDD0 for CH3 |
| 13 | Voltage monitor, CH4 |
| 14 | Current monitor through RSENSE, CH4 |
| 15 | PVDD1 for CH4 |
| 16 | Voltage monitor, CH5 |
| 17 | Current monitor through RSENSE, CH5 |
| 18 | PVDD1 for CH5 |
| 19 | Voltage monitor, CH6 |
| 20 | Current monitor through RSENSE, CH6 |
| 21 | PVDD1 for CH6 |
| 22 | Voltage monitor, CH7 |
| 23 | Current monitor through RSENSE, CH7 |
| 24 | PVDD1 for CH7 |
| 25 | Die Temperature |

## THEORY OF OPERATION

*Table 9. Multiplexer Channels Options (Continued)*

| SEL Field | MUX_OUT |
|---|---|
| 26 | AGND |

The corresponding full-scale output voltages for each monitored channel are provided in Table 10.

*Table 10. Full Scale Output Voltages*

| MUX_OUT Channel | Full Scale Output (V) |
|---|---|
| Current Monitor through RSENSE, CHx | PVDD |
| Voltage Monitor, CHx | VREF or 2 × VREF |
| Supply Voltage, PVDDx for CHx | PVDD |
| Die Temperature at 125°C | 0.61 |

### Voltage Monitor Channels

The transfer function of the integrated multiplexer, when voltage output monitor channel is selected, is given by the following equations.

For VDAC mode with OUTPUT_CONTROL_0, Bit 2 (range) = 0

$$V_{MEAS} = MUX\_OUT \quad (4)$$

For VDAC mode with OUTPUT_CONTROL_0, Bit 2 (range) = 1

$$V_{MEAS} = 2 \times MUX\_OUT \quad (5)$$

where:
VMEAS is the measured voltage output of the selected channel.
MUX_OUT is the voltage output on the MUX_OUT pin in volts.

A voltage output of VREF represents the full scale range of the DAC channel being monitored regardless of the OUTPUT_CONTROL_0, Bit 2 (range) value.

### Current Monitor Channels

When the MUX_OUT pin is configured to monitor the current of the OUTx channel, the transfer function depends on the value of the sense resistor ($R_{SENSE}$). The relationship is typically expressed as the following equation:

$$I_{OUTX} = \frac{P_{VDDx} - MUX\_OUT(V)}{R_{SENSE}} \quad (6)$$

where:
MUX_OUT is the voltage at the MUX_OUT pin in volts.
PVDDx is the supply voltage.
$R_{SENSE}$ is the internal sense resistor, with value of 1.84Ω ±10% nominal tolerance.

$PVDD_x$ can be measured directly at the supply pin or using the MUX_OUT function. Using the known value of the internal sense resistor $R_{SENSE}$ and the measured MUX_OUT voltage, the output current $I_{OUT_x}$ can be calculated using Equation 6.

### Die Temperature Monitor

The internal die temperature can also be monitored through the MUX_OUT pin by setting the SEL bits to 0x19. The transfer function used to derive the measured temperature is given by the following equation:

$$T_{MEAS} = \frac{MUX\_OUT(V) - 0.434V}{1.774mV/°C} + 25°C \quad (7)$$

where:
$T_{MEAS}$ is the measured internal die temperature in °C.
MUX_OUT is the voltage at the MUX_OUT pin in volts.

### Supply Voltages Monitor

The supplies PVDD0 and PVDD1 can be monitored through the MUX_OUT pin by setting the corresponding SEL bits. The transfer function is given by the following equation:

$$V_{PVDDX} = MUX\_OUT \quad (8)$$

where MUX_OUT is the voltage output on the MUX_OUT pin in volts.

## DAC CORE FUNCTIONS

Each DAC channel has an individual input register and a DAC register, as shown in Figure 1. Both registers are accessible through the serial interface. The DAC register stores digital code equivalent to the DAC output voltage while the input register acts as a temporary staging register before being passed on to the DAC register. With the LDAC function, one or more DAC registers can be updated in parallel with the data held in the input registers.

The DAC registers can be written to directly, in which case the corresponding output updates immediately without the need for a hardware or software LDAC. Directly writing to the DAC register does not affect the data stored in the input register.

Writing to the MULTI_INPUT_CH register allows one or more input registers to be updated in a single write operation. The MULTI_INPUT_SEL_0 register determines which channel's input register can be updated with the data written to the MULTI_INPUT_CH register. See the DAC Update section for additional information.

Similarly, writing to the MULTI_DAC_CH register allows one or more DAC registers to be updated in a single write operation. MULTI_DAC_INPUT_SEL_0 determines which DAC register can be updated with the data written to the MULTI_DAC_CH register. See the DAC Update section for more information.

To ensure that the DAC update is successful, DAC register updates should only occur once every 640ns. Refer to $t_{L2}$ and $t_{L3}$ from Table 3. An error flag is asserted when a DAC update write is unsuccessful, which can be check by reading the UPDATE_ERR bit on the STATUS_CONTROL_0 register.

### LDAC FUNCTION

The LDAC function is used to initiate the transfer of the contents of input registers of the selected channels to the corresponding DAC registers, thereby updating one or more OUT pins at the same time. The LDAC function can be executed by hardware through the $\overline{\text{LDAC}}$ pin or by software through SW_LDAC_TRIG_A or SW_LDAC_TRIG_B registers. Both hardware and software LDAC perform the same function.

#### Hardware LDAC

The AD5710R has an active low $\overline{\text{LDAC}}$ pin that is falling edge sensitive. If the $\overline{\text{LDAC}}$ signal is brought low, the selected input register contents are transferred to corresponding DAC register. If the $\overline{\text{LDAC}}$ is held low when writing to the device, the input registers appear transparent, and when an input register is written to, the DAC register is updated with the contents of the input register at the same time. When $\overline{\text{LDAC}}$ is held high, DAC codes can be written to any input registers without affecting the DAC output.

The $\overline{\text{LDAC}}$ is used to determine the DAC channels to be updated from the input registers of the corresponding DAC channels when $\overline{\text{LDAC}}$ is active low or asserted. By default, all DAC channels are selected and the HLD_EN_CH_n bitfields contain a 1b'1. A 1b'0 set on a HLD_EN_CH_n bitfield disables the hardware LDAC feature for the corresponding DAC channel.

#### Software LDAC

The software LDAC function is synonymous to an $\overline{\text{LDAC}}$ falling edge. It provides a way to initiate a transfer of content between input registers of the selected DAC channels to corresponding DAC registers through the serial interface via writing 1b'1 to SLD_TRIG_A bit on the SW_LDAC_TRIG_A register or to the SLD_TRIG_B bit on the SW_LDAC_TRIG_B register.

The SW_LDAC_EN_0 register is used to determine the DAC channels to be updated from the corresponding input registers when a software LDAC is performed. By default, all DAC channels are selected and the SLD_EN_CH_n bitfields contain a 1b'1. A 1b'0 set in a SLD_EN_CH_n bitfield disables the software LDAC feature for the corresponding DAC channel.

### DAC UPDATE

There are multiple methods for updating the DAC_CHn registers that directly affect the VDAC and IDAC outputs. Figure 55 applies when unique data is written to one or more channels, with variations depending on whether the update is immediate or controlled using the hardware or software LDAC. Figure 56 applies when the same data is written to multiple channels, allowing for immediate or simultaneous updates or controlled updates using either the hardware or software LDAC. The appropriate method depends on whether a single channel or multiple channels are being updated, whether the data is unique or the same, and the configuration of the LDAC signal.

## DAC CORE FUNCTIONS

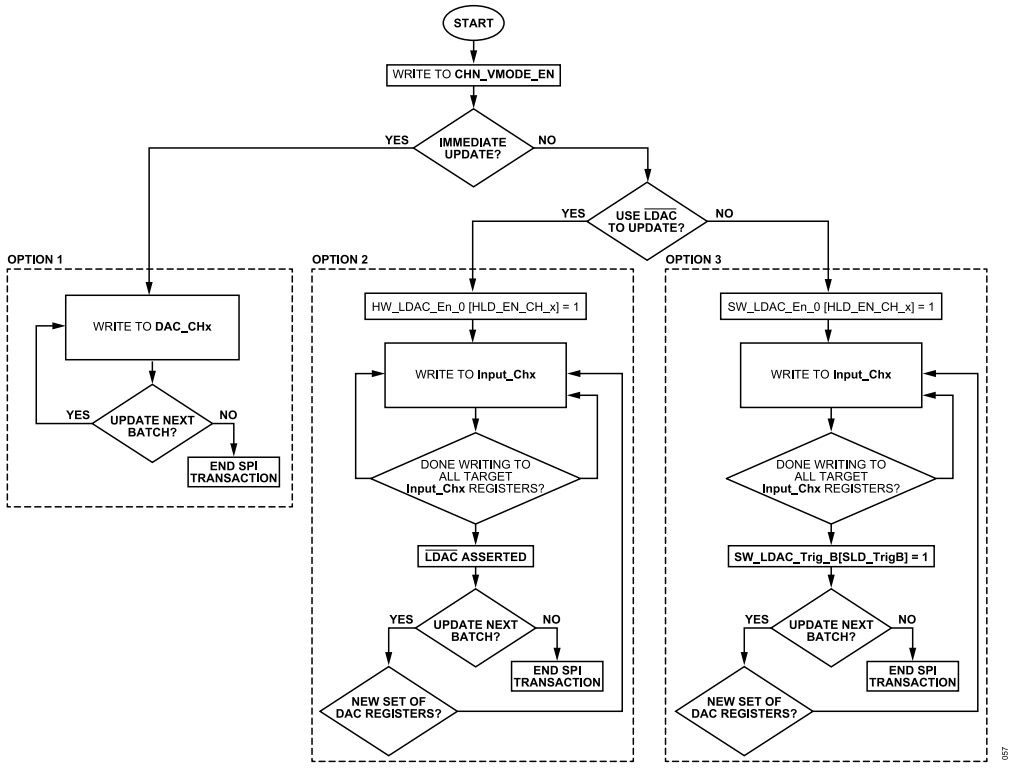


*Figure 55. DAC Update Flowchart for Option1, Option 2, and Option 3*

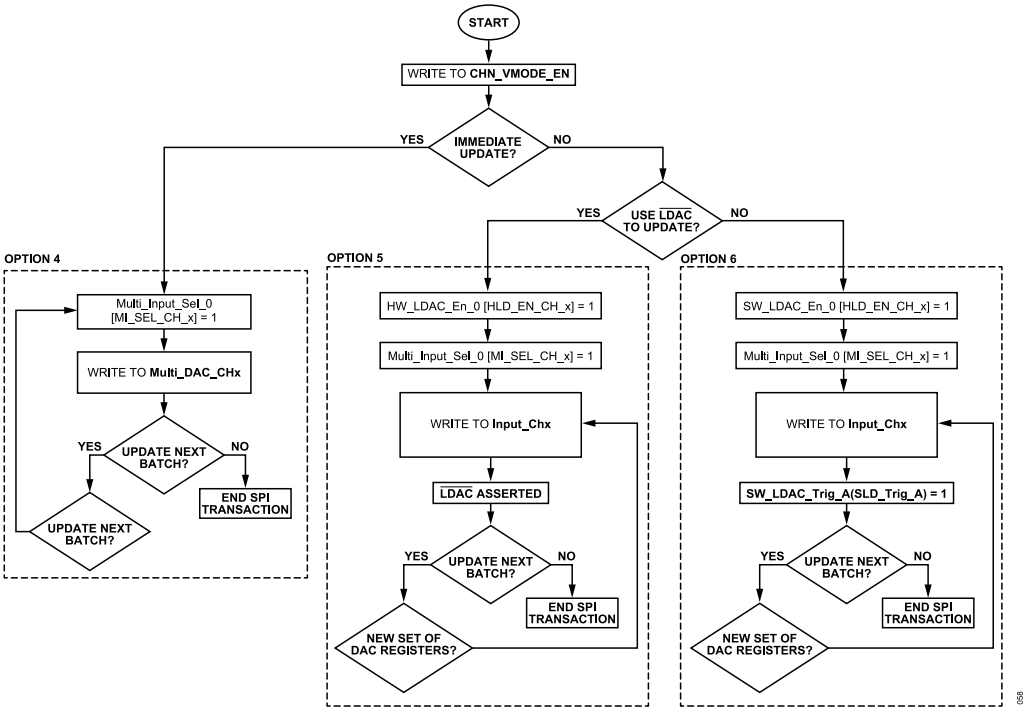


*Figure 56. DAC Update Flowchart for Option 4, Option 5, and Option 6*

## DAC CORE FUNCTIONS

### Option 1 (Immediate Update, Unique Data, No LDAC, and Single and Multiple Channels)

Option 1 allows the immediate update of DAC_CHn registers after writing the whole 16-bit data. An LDAC is not required, and it is applicable to both single channel and multiple channel updates in single instruction or stream modes.

A single instruction is sent with the descending mode selected. The higher address of the multibyte register is called on the instruction phase (0xD3 : DAC_CH0 register, followed by two 8-bit data) that updates the output immediately after the last SCLK.

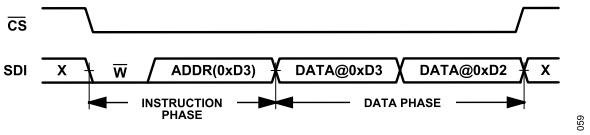


*Figure 57. Option 1 Example: Write to the DAC_Ch0 Register, Single Instruction Mode, and Address Descending*

### Option 2 (Controlled Update, Unique Data, Hardware LDAC, and Single and Multiple Channels)

Option 2 allows controlled update timing of the DAC_CHn registers from the INPUT_CHn registers through a hardware LDAC.

This option is applicable to both single channel and multiple channel updates in single instruction or stream modes.

A single instruction is sent to write to the HW_LDAC_EN_0 register, enabling a hardware LDAC for the selected channels. Then a stream mode is initiated, in default descending mode, writing to the INPUT_CH7 first up to INPUT_CH0. The LDAC is asserted at the end of the stream updating the DAC registers and the DAC output (if the correct LDAC timing is observed).

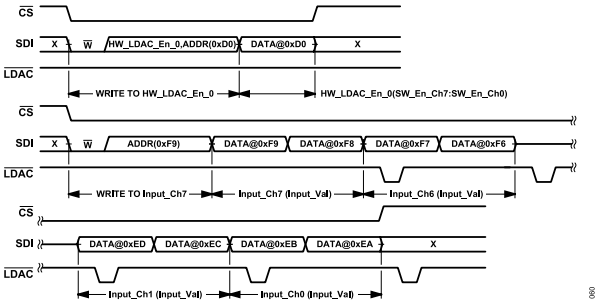


*Figure 58. Option 2 Example: Write to the INPUT_CH7 to INPUT_CH0 Registers, Hardware LDAC Enabled, Stream Mode, and Address Descending*

### Option 3 (Controlled Update, Unique Data, Software LDAC, and Single and Multiple Channels)

Option 3 allows controlled update timing of the DAC_CHn registers from the INPUT_CHn registers through the software LDAC.

This option is applicable to both single channel and multiple channel updates in single instruction and stream modes.

This option is similar to Option 2, except a software LDAC function is used instead of a hardware LDAC. The SW_LDAC_EN_0 register determines which channel gets affected by a software LDAC command. The DAC registers and DAC outputs are updated with the input data written after the last SCLK of the SW LDAC command.

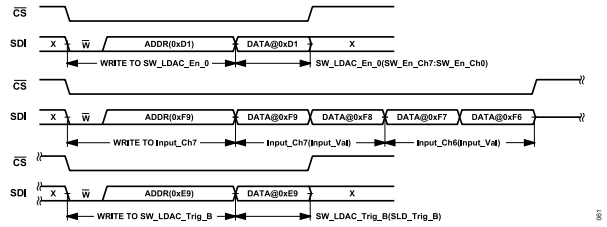


*Figure 59. Option 3 Example: Write to the INPUT_CH7 and INPUT_CH6 Registers with a Software LDAC, Stream Mode, and Address Descending*

### Option 4 (Immediate Update, Same Data, No LDAC, and Multiple Channels)

Option 4 allows the immediate and simultaneous update of multiple DAC_CHn registers identified by MULTI_DAC_SEL_0 (MD_SEL_CH_n) bitfields with the same data. Data is contained in MULTI_DAC_CH register and the update is initiated after writing the whole 16-bit data. An LDAC is not required.

Option 4 is ideal for multiple channel updates in both single instruction and stream modes.

A write instruction is sent to the MULTI_DAC_SEL_0 register, enabling multiple DAC functions for the selected channels. With stream mode descending enabled, the next commands are the data for the adjacent multibyte register 0xE3 (and 0xE2) MULTI_DAC_CH. Assuming that STREAM_MODE(LOOP_COUNT) is set to 0x3, the succeeding data stream loops back to the start address, 0xE4, and repeats the process, until CS is deasserted.

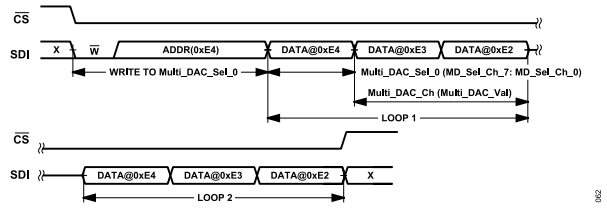


*Figure 60. Option 4 Example: Write to the MULTI_DAC_CHn Register, Stream Mode (Loop), and Address Descending*

### Option 5 (Controlled Update, Same Data, Hardware LDAC, and Multiple Channels)

Option 5 allows for the controlled update timing of multiple DAC_CHn registers identified by the MULTI_INPUT_SEL_0(MI_SEL_CH_n) and HW_LDAC_EN_0(HLD_EN_CH_n) bitfields with the same data.

## DAC CORE FUNCTIONS

Data is contained in the MULTI_INPUT_CHn register, and the update is initiated by providing a valid LDAC pulse.

Option 5 is ideal for multiple channel updates in both single instruction and stream modes. This option is similar to Option 2, except Option 5 makes use of the MULTI_INPUT_SEL_0 and MULTI_INPUT_CHn registers to select and update the input registers of multiple DAC channels. The LDAC is asserted after each loop ends, updating the DAC registers and the DAC outputs (when correct LDAC is observed).

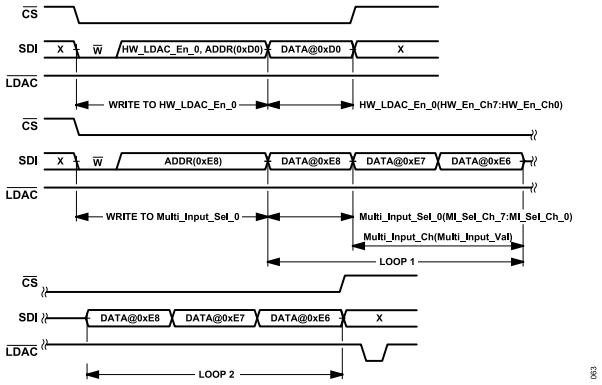


*Figure 61. Option 5 Example: Write to MULTI_INPUT_CHn Register with a Hardware LDAC, Stream Mode (Loop), and Address Ascending*

### Option 6 (Controlled Update, Same Data, Software LDAC, and Multiple Channels)

Option 6 allows immediate and simultaneous updating of multiple DAC_CHn registers identified by the MULTI_INPUT_SEL_0(MI_SEL_CH_n) and SW_LDAC_EN_0(SLD_EN_CH_n) bitfields with the same data. Data is contained in MULTI_INPUT_CHn register and is initiated through the software LDAC.

Option 6 is ideal for multiple channel updates in both single instruction and stream modes. It is similar to Option 5, except a software LDAC function is used instead of a hardware LDAC. The SW_LDAC_EN_0 register determines which channel is affected by a software LDAC command.

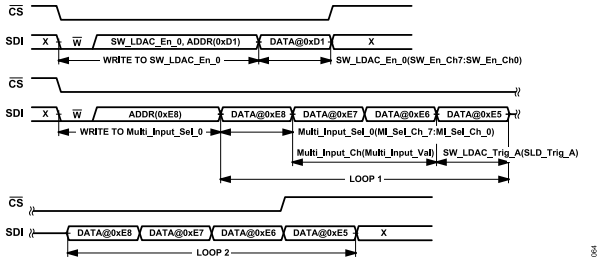


*Figure 62. Option 6 Example: Write to MULTI_INPUT_CHn Register with a Software LDAC, Stream Mode (Loop), and Address Ascending*

## DIGITAL INTERFACE

The device uses a 4-wire serial interface ($\overline{CS}$, SCLK, SDI, and SDO) that is compatible with classic SPI, QSPI, and MICROWIRE interface standards, as well as most digital signal processors (DSPs).

See Figure 2 for a timing diagram of a typical read-write sequence. Register read and write transactions are framed by $\overline{CS}$. While $\overline{CS}$ is high, SCLK edges are ignored, and SDO is high-Z. A falling edge on $\overline{CS}$ initiates a SPI frame. For all SPI transactions, data is shifted MSB first.

Data is sampled by the device on the rising edge of the SCLK while the data is shifted out on SDO on the falling edges of SCLK. This corresponds to SPI Mode 0 (CPOL = 0, CPHA = 0).

Communication with the device is separated into two distinct phases of operation. The first phase is the instruction phase and is used to initiate some action of the device. In the instruction phase, the register address is 15-bit wide (SHORT_INSTRUCTION bit = 0 in the INTERFACE_CONFIG_B register) by default. The second phase is the data phase where the data is either passed to the device to operate on or received from the device in response to the instruction phase.

Figure 63 shows a SPI write transaction using a 15-bit address in the instruction phase and eight bits for a single byte register for the data phase while Figure 64 shows a SPI read transaction using a 7-bit address in the instruction phase and eight bits for the data phase.

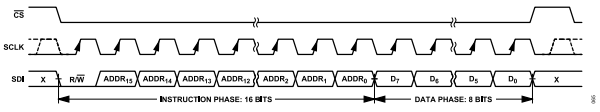


*Figure 63. Classic SPI Write*

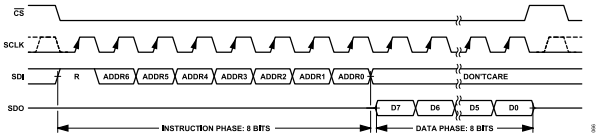


*Figure 64. Classic SPI Read*

## SPI FRAME SYNCHRONIZATION

The $\overline{CS}$ pin frames data during a SPI transaction. A falling edge on the $\overline{CS}$ enables the digital interface and initiates a SPI transaction. Each SPI transaction consists of at least one instruction phase and one data phase. For all SPI transactions, data is aligned MSB first. Deasserting the $\overline{CS}$ during a SPI transaction terminates part or all of the data transfer and disables the digital interface. If the $\overline{CS}$ is deasserted (returned high) after one or more registers are written, completed registers are written or read, but any partially written register is aborted. Figure 63 and Figure 64 show detailed timing diagrams for performing register reads and writes via the SPI interface.

## INSTRUCTION PHASE

Every SPI frame starts with the instruction phase. The instruction phase immediately follows the falling edge of the $\overline{CS}$ that initiates the SPI transaction. The instruction phase consists of a read/write bit (R/$\overline{W}$) followed by a register address word. Setting R/$\overline{W}$ low initiates a write instruction, whereas setting R/$\overline{W}$ high initiates a read instruction. The register address word specifies the address of the register to be accessed. The register address word is 15 bits in length by default. If required, enable 7-bit addressing with the SHORT_INSTRUCTION bit on the INTERFACE_CONFIG_B register. See the Interface Configuration B Register section for additional information.

## DATA PHASE

The data phase immediately follows the instruction phase (as shown in Figure 63 and Figure 64). The data phase can include the data for a single-byte register, a multibyte register, or multiple registers.

If the data phase of a SPI write transaction does not include the entire byte of data for the register being updated, the contents of the register are not updated, and the INTERFACE_STATUS_A register, Bit 4 (CLOCK_COUNT_ERR) is set.

## MULTIBYTE REGISTERS

Besides the one byte registers, the AD5710R also contains registers with two bytes of data stored in adjacent addresses that are referred to as multibyte registers. When writing to a multibyte register, all bytes must be accessed in a single SPI transaction. For this reason, the INTERFACE_CONFIG_C, Bit 5 (STRICT_REGISTER_ACCESS) is read only and set to 1. A write transaction to a multibyte register takes effect after the 16th SCLK edge of the data phase.

The address of a multibyte register always depends on the INTERFACE_CONFIG_A, Bit 5 (ADDR_ASCENSION). With addresses descending, the first byte accessed in the data phase must be the most significant byte of the multibyte register, and each subsequent byte corresponds to the data in the next lowest address. With addresses ascending, the first byte accessed in the data phase must be the least significant byte of the multibyte register, and each subsequent byte corresponds to the data in the next highest address.

For example, the DAC_CH0 register is two bytes long, and the addresses of its least significant byte and most significant byte are 0xD2 and 0xD3, respectively. Figure 65 and Figure 66 show read transactions of this register for address ascending and descending mode, respectively.

## DIGITAL INTERFACE

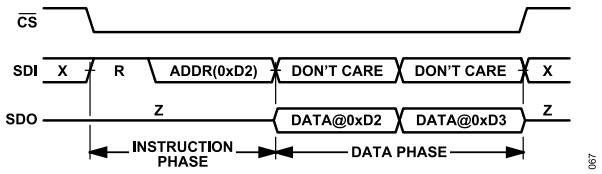


*Figure 65. Multibyte Read in Ascending Mode*

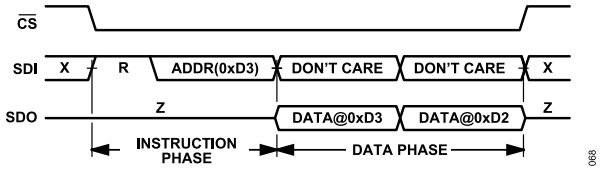


*Figure 66. Multibyte Read in Descending Mode*

Address direction is selected with the INTERFACE_CONFIG_A, Bit 5 (ADDR_ASCENSION). If this bit is set to 0, the address decrements after each byte is accessed. If this bit is set to 1, the address increments after each byte is accessed. If a SPI write transaction to a multibyte register is attempted on a per byte basis, the register contents are not updated on the device, and the INTERFACE_STATUS_A, Bit 1 (REGISTER_PARTIAL_ACCESS_ERR) is set.

This device contains the following multibyte registers: DAC_CHn, INPUT_CHn, MULTI_DAC_CH, and MULTI_INPUT_CH.

### SINGLE INSTRUCTION MODE

When the INTERFACE_CONFIG_B, Bit 7 (SINGLE_INST) 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 nonadjacent addresses in a single SPI frame (see Figure 67), whereas streaming mode only allows either reading or writing to contiguous registers without pulsing $\overline{CS}$ high to initiate a new instruction phase.

When accessing multibyte registers in single instruction mode, data phase should include all two bytes or 16 SCLK cycles, and the register address order is dependent on the INTERFACE_CONFIG_A, Bit 5 (ADDR_ASCENSION).

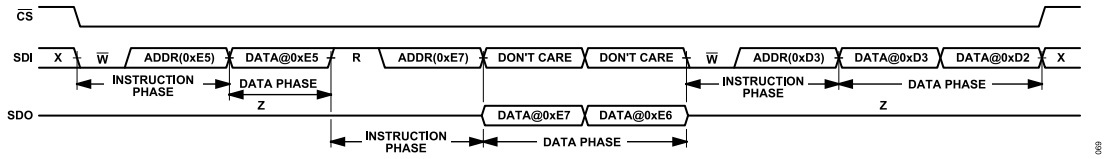


*Figure 67. Single Instruction Mode*

## DIGITAL INTERFACE

### STREAMING MODE

When the INTERFACE_CONFIG_B, Bit 7 (SINGLE_INST) bit 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 phase and data phase, allowing efficient access of contiguous regions of memory (for example, during initial device configuration). The streaming mode is selected by default.

When in streaming mode, each SPI frame consists of a single instruction phase, and the following data phase contains data for multiple registers with adjacent addresses. A starting register address is specified by the digital host in the instruction phase, and this address is automatically incremented or decremented (based on the address direction setting) after each byte of data is accessed. Therefore, the data phase can be multiple bytes long, and each consecutive byte of read or write data corresponds to the next highest or lowest register address (for ascending and descending address direction, respectively).

When writing to a multibyte register in streaming mode with address ascending, the least significant byte of the register must be addressed in the instruction phase and data must be provided starting from the least significant byte in the data phase. When writing to a multibyte register in streaming mode with the address descending, the user must start 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.

When reading from a multibyte register in streaming mode with address descending, read back the 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.

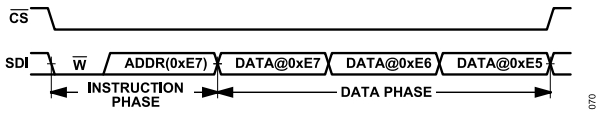


*Figure 68. Streaming Mode SPI Transfer*

The stream mode register can be used to specify a set of consecutive registers to loop through in the data phase (see the Stream Mode Register section). Looping allows the digital host to repeatedly read from or write to a set of registers as efficiently as possible.

If the address direction is set to 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 (0xFF).

If the address direction is set to ascending, the address increments until it reaches the highest valued byte address available (0xFF). On the subsequent byte access, the address is reset to 0x00.

If STREAM_MODE is set to a value other than 0, looping is enabled, and the value in STREAM_MODE 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 value of the LOOP_COUNT bits can be kept or returned to the default value of 0 when the frame transaction is completed, that is, the $\overline{CS}$ is brought high. The STREAM_MODE behavior is controlled by the KEEP_STREAM_LENGTH_VAL bit.

When using STREAM_MODE, be aware of the DAC update timings mentioned in the DAC Core Functions section.

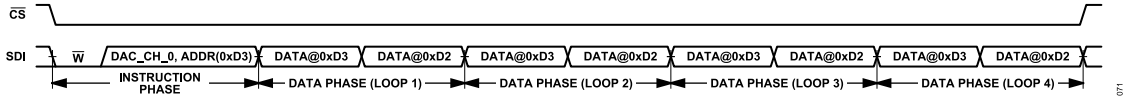


*Figure 69. Looping Enabled with LOOP_COUNT = 3*

## DIGITAL INTERFACE

### DEVICE ADDRESSING

The AD5710R devices 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 70. Care should be taken to ensure that every device on the same SPI bus has unique device address on the address pins.

To communicate with any AD5710R, the instruction phase in the SPI frame has three MSB Bits A[14:12] which correspond to the device addresses as mentioned in Figure 2 and Table 11. It is possible to select and write to a specific device using the corresponding address bits in the instruction phase. A SPI read transaction must address a specific device to read out data on the SDO line. There are user configurable registers with Address 0x00 to Address 0x11 that can be read or written to while ignoring the values set for the address bits in the instruction phase. For other registers, the AD5710R ignores the SPI read and write transactions where the address pins are not matched with the address bits in the instruction phase.

*Table 11. Device Addressing Truth Table*

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

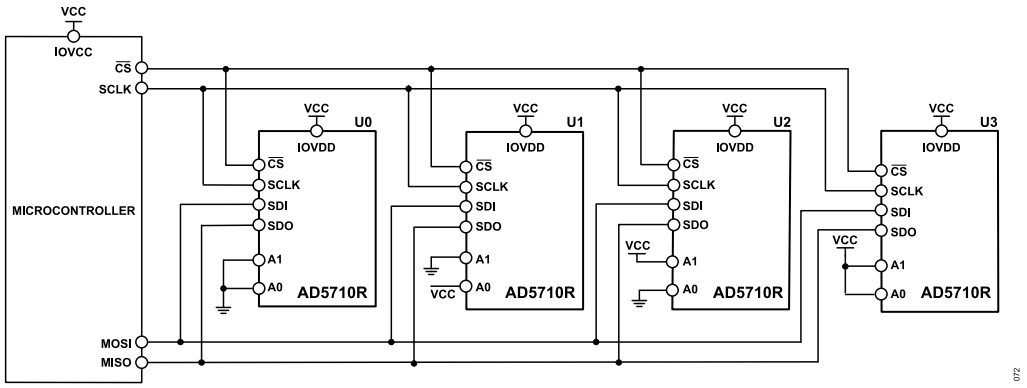


*Figure 70. Communication with Multiple AD5710R Devices on a Single SPI Bus*

## DIGITAL INTERFACE

When multiple AD5710R 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 SCLK frequency to allow more time for the SDO to settle after a transition.

### DEVICE STATUS ERRORS

The AD5710R has a INTERFACE_STATUS_A register that contains all the status bits related to the digital interface. These status bits are set based on the actions related to the SPI transactions, until explicitly clearly by writing a 1 to the bit location to clear the status bit.

#### Partial Register Access Error

The REGISTER_ PARTIAL_ACCESS_ERR bit in the INTERFACE_STATUS_A register is set when a multibyte register is accessed for read or write partially, which means that the transaction ends before all the bytes of a multibyte register have been accessed. To clear this error, write 1 to the REGISTER_ PARTIAL_ACCESS_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, this 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 SCLK cycles is not in accordance with the amount required to shift a multiple of eight bits.

#### Interface Not Ready Status

The NOT_READY_ERR bit in the INTERFACE_STATUS_A register is not an error. It is a status bit that can be polled to know 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.

### CYCLIC REDUNDANCY CHECK (CRC) ERROR DETECTION

The AD5710R DAC features an optional cyclic redundancy check (CRC) to provide error detection for SPI transactions between the digital host and the DAC (target). The CRC error detection is disabled by default. CRC error detection allows the SPI host and targets to detect bit transfer errors with significant reliability. The CRC algorithm involves using a seed value and polynomial division to generate a CRC code. The controller and target both calculate the CRC code independently to determine the validity of transferred data.

This DAC uses the CRC-8 standard with the following polynomial:

$$x^8 + x^2 + x + 1 \quad (9)$$

CRC error detection is enabled with the CRC_EN and CRC_EN_B bits in the INTERFACE_CONFIG_C register. The value of CRC_EN is only updated if CRC_EN_B is set to the CRC_EN inverted value in the same register write instruction. Therefore, to enable the CRC, the CRC_EN must be set to 0b01 while CRC_EN_B is set to 0b10 in the same write transaction.

To disable the CRC, the CRC_EN must be set to 0b00 and the CRC_EN_B is set to 0b11 in the same write transaction.

Writing inverted values to two separate fields reduces the chances of CRC being enabled in error. The $\overline{CS}$ 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 the CRC must still include the CRC code on SDI, but the following transaction does not require the CRC code.

Figure 71 and Figure 72 show how a CRC code is appended to the write or read, respectively, for the digital host or DAC to validate the data. For register writes, the digital host must generate the CRC using the calculation described in Equation 9. For register reads, the host must also send the correct CRC byte that is checked by the DAC. The first byte of data sent contributes to the CRC calculation. Therefore, a value of 0x00 is recommended. In the same read transaction, the DAC provides the CRC code for the digital host to verify.

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

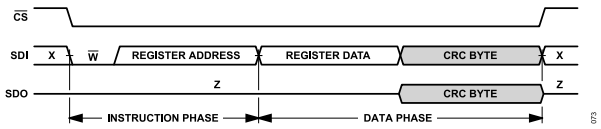


*Figure 71. SPI Write with CRC Enabled*

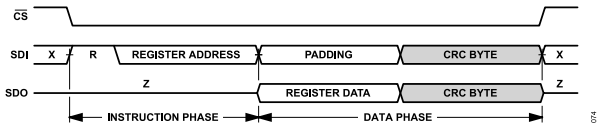


*Figure 72. SPI Read with CRC Enabled*

When the CRC error detection is enabled, the DAC does not update its register contents in response to a register write transaction unless it receives a valid CRC code at the end of the register data on the SDI. If the CRC code is invalid, or the digital host fails to transmit the CRC code, the AD5710R does not update its register

## DIGITAL INTERFACE

contents, and the CRC_ERR flag in the INTERFACE_STATUS_A register is set. The CRC_ERR flag is write-1-to-clear (W1C) and the correct CRC is required for the write to clear to take effect.

Table 12 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 the streaming mode, the first CRC code in a SPI frame also uses 0xA5 as the seed value, but subsequent CRC codes in the same frame are calculated using the LSB of the register address being accessed in the SPI transaction as the seed value.

*Table 12. 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 |

### DEVICE RESET

The device requires a minimum of 167µs between any reset event and a register read/write transaction as shown in Figure 3, and represented by tSPI_RDY in Table 3. If a 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. See Device Status Errors for more information about NOT_READY_ERR bit.

The AD5710R offers three reset mechanisms: Power-On Reset (POR), Hardware Reset, and Software Reset via the serial interface.

### Power-On Reset (POR)

A POR signal is generated when the supply voltage first crosses the nominal threshold of 2.5V on AVDD. The POR resets the state of the user programmable registers if the supply voltage on AVDD drops below this threshold. On power-up, all registers are reset to their default values. Meanwhile, the POR circuit ensures that the output stages of all DAC channels are powered down (see the DAC Operating Modes section) until the output operating mode for the channel is changed.

### Hardware Reset

$\overline{\text{RESET}}$ is an active low signal that is low level triggered. Asserting $\overline{\text{RESET}}$ sets the device into the POR state. While $\overline{\text{RESET}}$ is asserted, all SPI transactions and $\overline{\text{LDAC}}$ pulses are ignored, and the SDO output is in a high-Z state. When the $\overline{\text{RESET}}$ is deasserted, the digital core initialization is performed, and all registers are reset to their default values.

### Sofware Reset

The INTERFACE_CONFIG_A register includes two control bits, SW_RESET and $\overline{\text{RESET}}$_SW used to initiate a software reset through the serial interface. Both bits must be set simultaneously within the same data phase to successfully trigger the reset. Once the software reset transaction is completed, the device performs a POR sequence followed by digital core initialization. All registers are reset to their default values, except for the INTERFACE_CONFIG_A register, which retains its current configuration.

## APPLICATIONS INFORMATION

### POWER SUPPLY RECOMMENDATIONS

The AD5710R does not have any restrictions for power supply sequencing. The outputs are maintained at POR state with a known pull-down resistance until proper register configurations are set.

The application module is very limited in terms of space, therefore it is necessary to reduce both the number of external components and their size without compromising functional or specification requirements. Typically, capacitor values are considered with a ±5% tolerance unless otherwise specified.

*Table 13. External Passive Components*

| Pin Name | Passive Components Required |
|---|---|
| AVDD | 0.1µF \|\| 10µF capacitor per pin to AGND |
| PVDDx | 0.1µF \|\| 10µF capacitor per pin to AGND |
| IOVDD | 0.1µF capacitor to AGND |

### MICROPROCESSOR INTERFACING

Microprocessor interfacing to the AD5710R 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 (MISO), a data output signal (MOSI), and a synchronization signal (SS).

The SPI interface of the AD5710R is designed for easy connection to industry-standard DSPs and microcontrollers. Figure 73 shows the AD5710R connected to the ADuCM320. The ADuCM320 has an integrated SPI port that can be connected directly to the SPI pins of the AD5710R.

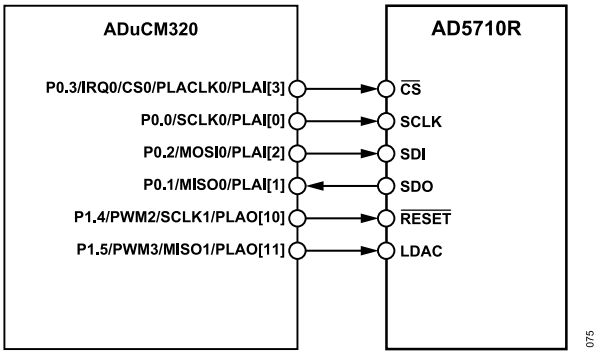


*Figure 73. ADuCM320 SPI Interface*

### LAYOUT GUIDELINES AND ASSEMBLY GUIDELINES

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 and a maximum operating junction temperature ($T_{J\_MAX}$) of 125°C (see the Absolute Maximum Ratings section). To meet the specified performance, the AD5710R must be operated at a junction temperature no greater than 125°C. The junction temperature is directly affected by the power dissipated across the AD5710R and the ambient temperature.

Most of the power dissipation of the AD5710R comes from the IOUT channels. The DC Specifications section 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 25-ball, 2.143mm × 2.193mm × 0.500mm WLCSP. The thermal resistance values are specified in Table 6. Table 14 provides examples of the maximum allowed power dissipation, the maximum allowed ambient temperature, and the maximum board temperature under certain specified conditions.

## APPLICATIONS INFORMATION

*Table 14. Thermal Considerations for AD5710R 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>$P_{DISS\_MAX} = \frac{T_{J\_MAX} - T_A}{\theta_{JA}} = \frac{125\ ^\circ C\ - 85\ ^\circ C}{51.8\ ^\circ C\ /W} = 772.20mW$ (10)<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>$P_{DISS\_MAX} = \frac{T_{J\_MAX} - T_B}{\Psi_{JB}} = \frac{125\ ^\circ C\ - 95\ ^\circ C}{16\ ^\circ C\ /W} = 1.87W$ (11)<br>Using the preceding equation, calculate the AD5710R $P_{DISS}$ as follows, and confirm $P_{DISS}$ is lower than the allowed $P_{DISS\_MAX}$.<br>Supplies conditions chosen as follows:<br>$R_{LOAD}$ = 25Ω and PVDDx = 1.71V, per channel.<br>AD5710R quiescent power dissipation = 0.8µA × 1.71V = 1.368µW<br>IOUT0 current: 50mA, power dissipation = 23mW<br>$P_{DISS}$ = 1.368µW (quiescent) + 23mW (active) = 23.001mW (Total) |
| Maximum Ambient Temperature | Maximum allowed ambient temperature, when dissipating 23.001W across the AD5710R is<br>$T_{A\_MAX} = T_{J\_MAX} - \left(P_{DISS} \times \theta_{JA}\right) = 125°C - \left(23.001mW \times 51.8°C/W\right) = 123.80°C$ (12) |
| Maximum Board Temperature | Maximum allowed board temperature, when dissipating 23.001W across the AD5710R is<br>$T_{B\_MAX} = T_{J\_MAX} - \left(P_{DISS} \times \Psi_{JB}\right) = 125°C - \left(23.001mW \times 16°C/W\right) = 124.63°C$ (13) |

## REGISTERS

The AD5710R has programmable user configuration registers that are used to configure the device. Table 15 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 a SPI write transaction, whereas read/write bits can. The error flags in the INTERFACE_STATUS_A and STATUS_CONTROL_0 registers are also read or write 1 to clear (R/W1C) and are only reset when a SPI write transaction writes a 1 to their location.

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

### REGISTER SUMMARY

*Table 15. AD5710R Register Summary*

| Reg | Name | Bits | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Reset | Access |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 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_MODES | | 0x00 | R |
| 0x03 | CHIP_TYPE | [7:0] | RESERVED | | | | CHIP_TYPE | | | | 0x08 | R |
| 0x04 | PRODUCT_ID_L | [7:0] | PRODUCT_ID[7:0] | | | | | | | | 0x08 | R |
| 0x05 | PRODUCT_ID_H | [7:0] | PRODUCT_ID[15:8] | | | | | | | | 0x00 | R |
| 0x06 | CHIP_GRADE | [7:0] | GRADE | | | | DEVICE_REVISION | | | | 0x01 | R |
| 0x0A | SCRATCH_PAD | [7:0] | SCRATCH_VALUE | | | | | | | | 0x00 | R/W |
| 0x0B | SPI_REVISION | [7:0] | SPI_TYPE | | VERSION | | | | | | 0x84 | 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 | RESERVED | ACTIVE_INTERFACE_MODE | | CRC_ENABLEB | | 0x23 | R/W |
| 0x11 | INTERFACE_STATUS_A | [7:0] | NOT_READY_ERR | RESERVED | | CLOCK_COUNT_ERR | CRC_ERR | RESERVED | REGISTER_PARTIAL_ACCESS_ERR | RESERVED | 0x00 | R/W |
| 0x20 | OUTPUT_OPERATING_MODE_0 | [7:0] | MODE_CH_3 | | MODE_CH_2 | | MODE_CH_1 | | MODE_CH_0 | | 0xFF | R/W |
| 0x21 | OUTPUT_OPERATING_MODE_1 | [7:0] | MODE_CH_7 | | MODE_CH_6 | | MODE_CH_5 | | MODE_CH_4 | | 0xFF | R/W |
| 0x2A | OUTPUT_CONTROL_0 | [7:0] | RESERVED | | | | | RANGE | RESERVED | | 0x00 | R/W |
| 0x3C | REFERENCE_CONTROL_0 | [7:0] | RESERVED | | | | | | | SEL | 0x00 | R/W |
| 0x93 | MUX_OUT_SELECT | [7:0] | RESERVED | | | SEL | | | | | 0x00 | R/W |
| 0xC2 | STATUS_CONTROL_0 | [7:0] | RESERVED | | | | UPDATE_ERR | RESET_WARNING | INTERFACE_ERR | RESERVED | 0x04 | R/W |
| 0xD0 | HW_LDAC_En_0 | [7:0] | HLD_EN_CH_7 | HLD_EN_CH_6 | HLD_EN_CH_5 | HLD_EN_CH_4 | HLD_EN_CH_3 | HLD_EN_CH_2 | HLD_EN_CH_1 | HLD_EN_CH_0 | 0xFF | R/W |

## REGISTERS

*Table 15. AD5710R Register Summary (Continued)*

| Reg | Name | Bits | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Reset | Access |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0xD1 | SW_LDAC_En_0 | [7:0] | SLD_EN_CH_7 | SLD_EN_CH_6 | SLD_EN_CH_5 | SLD_EN_CH_4 | SLD_EN_CH_3 | SLD_EN_CH_2 | SLD_EN_CH_1 | SLD_EN_CH_0 | 0xFF | R/W |
| 0xD2 to 0xE0 by 2 | DAC_CHn | [15:8] | DAC_VAL[15:8] | | | | | | | | 0x0000 | R/W |
| | | [7:0] | DAC_VAL[7:0] | | | | | | | | | |
| 0xE2 | MULTI_DAC_CH | [15:8] | MULTI_DAC_VAL[15:8] | | | | | | | | 0x0000 | R/W |
| | | [7:0] | MULTI_DAC_VAL[7:0] | | | | | | | | | |
| 0xE4 | MULTI_DAC_SEL_0 | [7:0] | MD_SEL_CH_7 | MD_SEL_CH_6 | MD_SEL_CH_5 | MD_SEL_CH_4 | MD_SEL_CH_3 | MD_SEL_CH_2 | MD_SEL_CH_1 | MD_SEL_CH_0 | 0xFF | R/W |
| 0xE5 | SW_LDAC_TRIG_A | [7:0] | SLD_TRIG_A | RESERVED | | | | | | | 0x00 | W |
| 0xE6 | MULTI_INPUT_CH | [15:8] | MULTI_INPUT_VAL[15:8] | | | | | | | | 0x0000 | R/W |
| | | [7:0] | MULTI_INPUT_VAL[7:0] | | | | | | | | | |
| 0xE8 | MULTI_INPUT_SEL_0 | [7:0] | MI_SEL_CH_7 | MI_SEL_CH_6 | MI_SEL_CH_5 | MI_SEL_CH_4 | MI_SEL_CH_3 | MI_SEL_CH_2 | MI_SEL_CH_1 | MI_SEL_CH_0 | 0xFF | R/W |
| 0xE9 | SW_LDAC_TRIG_B | [7:0] | SLD_TRIG_B | RESERVED | | | | | | | 0x00 | W |
| 0xEA to 0xF8 by 2 | INPUT_CHn | [15:8] | INPUT_VAL[15:8] | | | | | | | | 0x0000 | R/W |
| | | [7:0] | INPUT_VAL[7:0] | | | | | | | | | |
| 0xFF | CHn_VMODE_EN | [7:0] | VMODE_EN_CH7 | VMODE_EN_CH6 | VMODE_EN_CH5 | VMODE_EN_CH4 | VMODE_EN_CH3 | VMODE_EN_CH2 | VMODE_EN_CH1 | VMODE_EN_CH0 | 0x00 | R/W |

# REGISTERS

## REGISTER DETAILS

### Interface Configuration A Register

Interface configuration settings

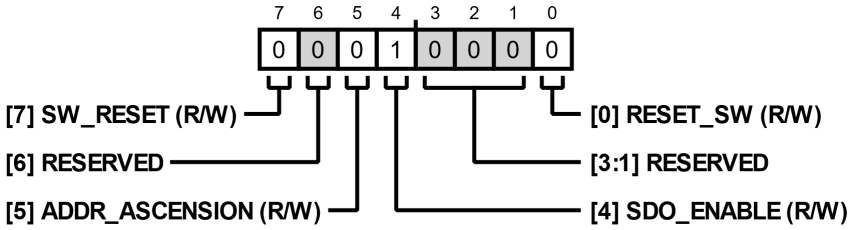


*Figure 74. Address: 0x00, Reset: 0x10, Name: INTERFACE_CONFIG_A*

*Table 16. Bit Descriptions for INTERFACE_CONFIG_A*

| Bits | Bit Name | Description | Reset | Access |
|---|---|---|---|---|
| 7 | SW_RESET | First of Two 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.<br>1'b0: SDO pin disabled.<br>1'b1: SDO pin enabled (default, 4-wire). | 0x1 | R/W |
| [3:1] | RESERVED | Reserved. | 0x0 | R |
| 0 | RESET_SW | Second of Two 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

Additional interface configuration settings

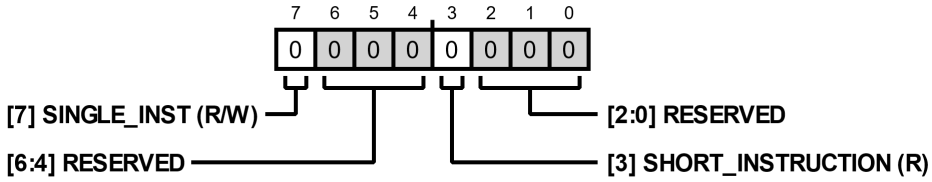


*Figure 75. Address: 0x01, Reset: 0x00, Name: INTERFACE_CONFIG_B*

*Table 17. 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. | 0x0 | R/W |

## REGISTERS

*Table 17. Bit Descriptions for INTERFACE_CONFIG_B (Continued)*

| Bits | Bit Name | Description | Reset | Access |
|---|---|---|---|---|
| | | 1: Single Instruction Mode is Enabled. | | |
| [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 |
| [2:0] | RESERVED | Reserved. | 0x0 | R |

### Device Configuration Register

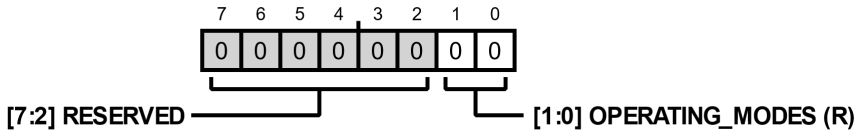


*Figure 76. Address: 0x02, Reset: 0x00, Name: DEVICE_CONFIG*

*Table 18. Bit Descriptions for DEVICE_CONFIG*

| Bits | Bit Name | Description | Reset | Access |
|---|---|---|---|---|
| [7:2] | RESERVED | Reserved. | 0x0 | R |
| [1:0] | OPERATING_MODES | Operating Modes. Read only. | 0x0 | R |

### Chip Type Register

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

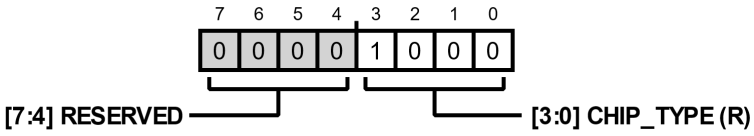


*Figure 77. Address: 0x03, Reset: 0x08, Name: CHIP_TYPE*

*Table 19. 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

## REGISTERS

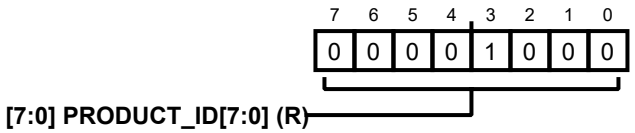


*Figure 78. Address: 0x04, Reset: 0x08, Name: PRODUCT_ID_L*

*Table 20. 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 the Chip Type to identify a product.<br>0x8: AD5710R. | 0x8 | R |

### Product ID High Register

High byte of the Product ID

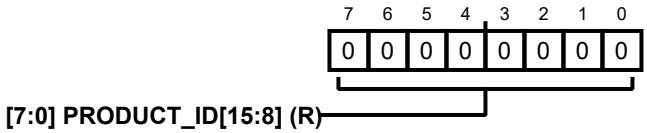


*Figure 79. Address: 0x05, Reset: 0x00, Name: PRODUCT_ID_H*

*Table 21. 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 the Chip Type to identify a product.<br>0x8: AD5710R. | 0x0 | R |

### Chip Grade Register

Identifies product variations and device revisions

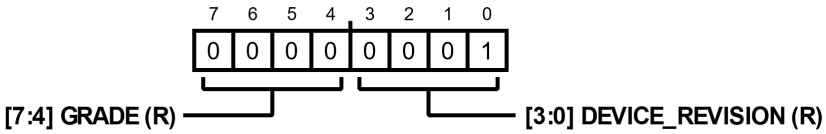


*Figure 80. Address: 0x06, Reset: 0x01, Name: CHIP_GRADE*

*Table 22. 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. | 0x1 | R |

### Scratch Pad Register

This may be used to test writes and reads

## REGISTERS

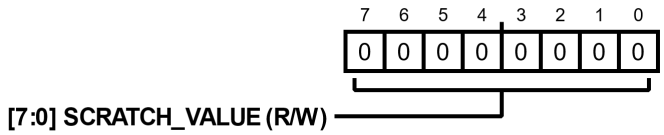


*Figure 81. Address: 0x0A, Reset: 0x00, Name: SCRATCH_PAD*

*Table 23. 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 |

### SPI Revision Register

Indicates the SPI interface revision

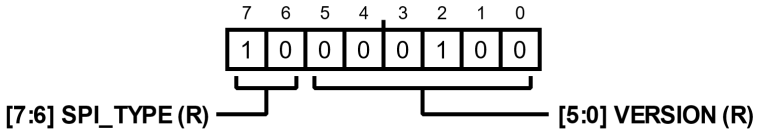


*Figure 82. Address: 0x0B, Reset: 0x84, Name: SPI_REVISION*

*Table 24. Bit Descriptions for SPI_REVISION*

| Bits | Bit Name | Description | Reset | Access |
|---|---|---|---|---|
| [7:6] | SPI_TYPE | Always Reads as 0x2. | 0x2 | R |
| [5:0] | VERSION | SPI Version. | 0x4 | R |

### Vendor ID Low Register

Low byte of the Vendor ID

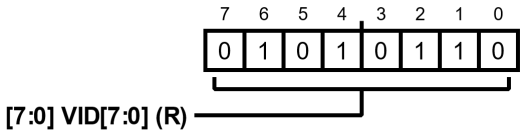


*Figure 83. Address: 0x0C, Reset: 0x56, Name: VENDOR_L*

*Table 25. 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

## REGISTERS

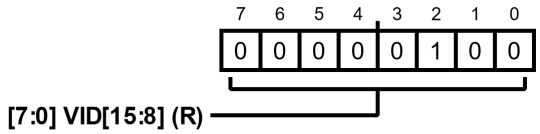


*Figure 84. Address: 0x0D, Reset: 0x04, Name: VENDOR_H*

*Table 26. 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

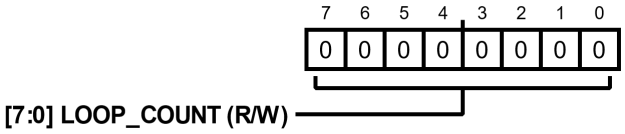


*Figure 85. Address: 0x0E, Reset: 0x00, Name: STREAM_MODE*

*Table 27. Bit Descriptions for STREAM_MODE*

| Bits | Bit Name | Description | Reset | Access |
|---|---|---|---|---|
| [7:0] | LOOP_COUNT | Sets the Data Byte Count Before Looping to Start Address. When streaming data, a nonzero 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 the movement of data between master and slave registers

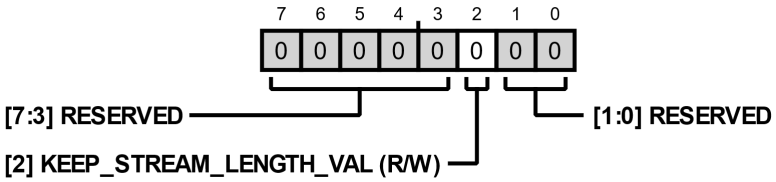


*Figure 86. Address: 0x0F, Reset: 0x00, Name: TRANSFER_CONFIG*

*Table 28. 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 CSB Rising Edge. | 0x0 | R/W |
| [1:0] | RESERVED | Reserved. | 0x0 | R |

## REGISTERS

### Interface Configuration C Register

Additional interface configuration settings

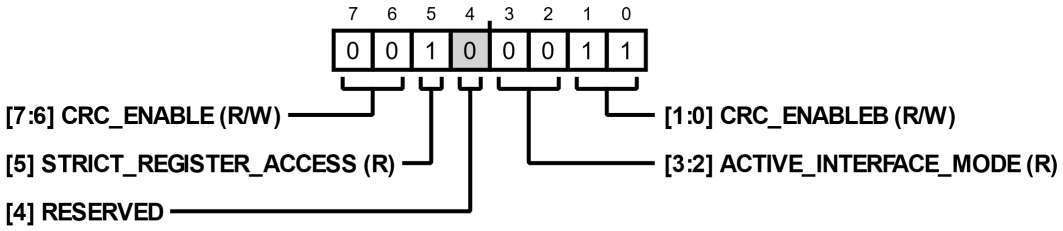


*Figure 87. Address: 0x10, Reset: 0x23, Name: INTERFACE_CONFIG_C*

*Table 29. 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.<br>1: CRC Enabled. | 0x0 | R/W |
| 5 | STRICT_REGISTER_ACCESS | Multibyte Registers Must Be Read/Written in Full. When this mode is enabled, all bytes of a multibyte 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 | RESERVED | Reserved. | 0x0 | R |
| [3:2] | ACTIVE_INTERFACE_MODE | Active Interface Mode. This is the active mode the SPI interface is operating in.<br>0: | 0x0 | 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 and can be cleared by writing a '1' to the corresponding bit location

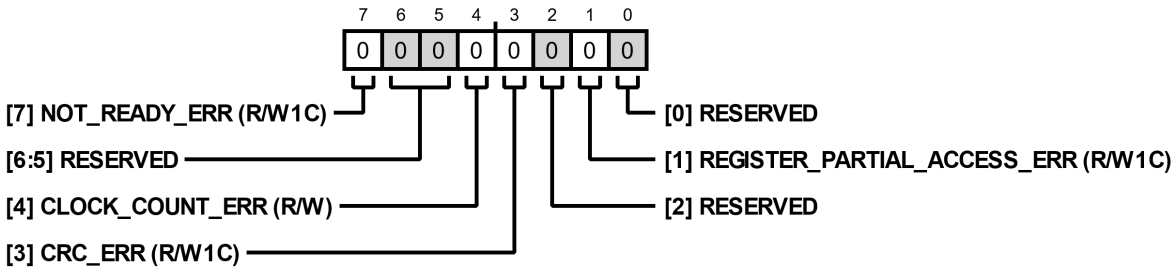


*Figure 88. Address: 0x11, Reset: 0x00, Name: INTERFACE_STATUS_A*

*Table 30. Bit Descriptions for INTERFACE_STATUS_A*

| Bits | Bit Name | Description | Reset | Access |
|---|---|---|---|---|
| 7 | NOT_READY_ERR | Device Not Ready for Transaction. This error 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 |

## REGISTERS

*Table 30. Bit Descriptions for INTERFACE_STATUS_A (Continued)*

| Bits | Bit Name | Description | Reset | Access |
|---|---|---|---|---|
| 4 | CLOCK_COUNT_ERR | Incorrect Number of Clocks Detected in a Transaction. | 0x0 | R/W |
| 3 | CRC_ERR | Invalid/No CRC Received. This is set when the master fails to send a CRC or when the device calculates and checks the CRC and finds the CRC value is incorrect. | 0x0 | R/W1C |
| 2 | RESERVED | Reserved. | 0x0 | R |
| 1 | REGISTER_PARTIAL_ACCESS_ERR | Set When Fewer Than Expected Number of Bytes Read/Written. This bit is only valid when strict register access is enabled. | 0x0 | R/W1C |
| 0 | RESERVED | Reserved. | 0x0 | R |

### Output Operating Mode 0 Register

Configures the operating modes for Channel 0 to Channel 3

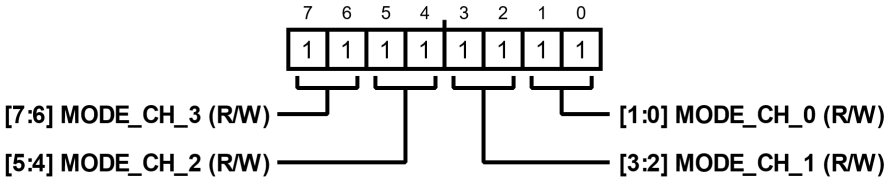


*Figure 89. Address: 0x20, Reset: 0xFF, Name: OUTPUT_OPERATING_MODE_0*

*Table 31. Bit Descriptions for OUTPUT_OPERATING_MODE_0*

| Bits | Bit Name | Description | Reset | Access |
|---|---|---|---|---|
| [7:6] | MODE_CH_3 | Mode Channel 3. Output operating mode for Channel 3.<br>00: Channel Enabled.<br>01: Channel Disabled: VMODE: 15kΩ output impedance; IMODE: high-Z.<br>10: Channel Disabled: VMODE: 15kΩ output impedance; IMODE: high-Z.<br>11: Channel Disabled: VMODE: 15kΩ output impedance; IMODE: high-Z. | 0x3 | R/W |
| [5:4] | MODE_CH_2 | Mode Channel 2. Output operating mode for Channel 2.<br>00: Channel Enabled.<br>01: Channel Disabled: VMODE: 15kΩ output impedance; IMODE: high-Z.<br>10: Channel Disabled: VMODE: 15kΩ output impedance; IMODE: high-Z.<br>11: Channel Disabled: VMODE: 15kΩ output impedance; IMODE: high-Z. | 0x3 | R/W |
| [3:2] | MODE_CH_1 | Mode Channel 1. Output operating mode for Channel 1.<br>00: Channel Enabled.<br>01: Channel Disabled: : VMODE: 15kΩ output impedance; IMODE: high-Z.<br>10: Channel Disabled: : VMODE: 15kΩ output impedance; IMODE: high-Z.<br>11: Channel Disabled: : VMODE: 15kΩ output impedance; IMODE: high-Z. | 0x3 | R/W |
| [1:0] | MODE_CH_0 | Mode Channel 0. Output operating mode for Channel 0.<br>00: Channel Enabled.<br>01: Channel Disabled: : VMODE: 15kΩ output impedance; IMODE: high-Z.<br>10: Channel Disabled: : VMODE: 15kΩ output impedance; IMODE: high-Z.<br>11: Channel Disabled: : VMODE: 15kΩ output impedance; IMODE: high-Z. | 0x3 | R/W |

### Output Operating Mode 1 Register

Configures the operating modes for Channel 4 to Channel 7

## REGISTERS

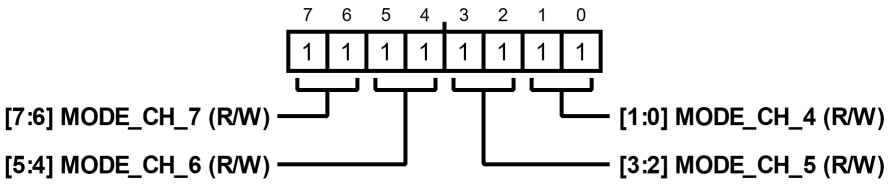


*Figure 90. Address: 0x21, Reset: 0xFF, Name: OUTPUT_OPERATING_MODE_1*

*Table 32. Bit Descriptions for OUTPUT_OPERATING_MODE_1*

| Bits | Bit Name | Description | Reset | Access |
|---|---|---|---|---|
| [7:6] | MODE_CH_7 | Mode Channel 7. Output operating mode for Channel 7.<br>00: Channel Enabled.<br>01: Channel Disabled: VMODE: 15kΩ output impedance; IMODE: high-Z.<br>10: Channel Disabled: VMODE: 15kΩ output impedance; IMODE: high-Z.<br>11: Channel Disabled: VMODE: 15kΩ output impedance; IMODE: high-Z. | 0x3 | R/W |
| [5:4] | MODE_CH_6 | Mode Channel 6. Output operating mode for Channel 6.<br>00: Channel Enabled.<br>01: Channel Disabled: VMODE: 15kΩ output impedance; IMODE: high-Z.<br>10: Channel Disabled: VMODE: 15kΩ output impedance; IMODE: high-Z.<br>11: Channel Disabled: VMODE: 15kΩ output impedance; IMODE: high-Z. | 0x3 | R/W |
| [3:2] | MODE_CH_5 | Mode Channel 5. Output operating mode for Channel 5.<br>00: Channel Enabled.<br>01: Channel Disabled: VMODE: 15kΩ output impedance; IMODE: high-Z.<br>10: Channel Disabled: VMODE: 15kΩ output impedance; IMODE: high-Z.<br>11: Channel Disabled: VMODE: 15kΩ output impedance; IMODE: high-Z. | 0x3 | R/W |
| [1:0] | MODE_CH_4 | Mode Channel 4. Output operating mode for Channel 4.<br>00: Channel Enabled.<br>01: Channel Disabled: VMODE: 15kΩ output impedance; IMODE: high-Z.<br>10: Channel Disabled: VMODE: 15kΩ output impedance; IMODE: high-Z.<br>11: Channel Disabled: VMODE: 15kΩ output impedance; IMODE: high-Z. | 0x3 | R/W |

### Output Control 0 Register

Configures the output range for all channels

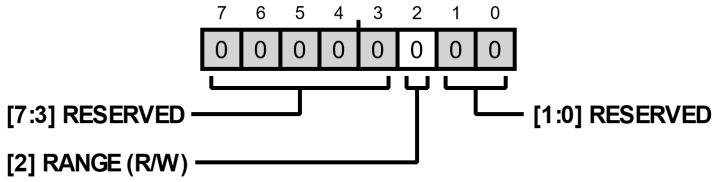


*Figure 91. Address: 0x2A, Reset: 0x00, Name: OUTPUT_CONTROL_0*

*Table 33. Bit Descriptions for OUTPUT_CONTROL_0*

| Bits | Bit Name | Description | Reset | Access |
|---|---|---|---|---|
| [7:3] | RESERVED | Reserved. | 0x0 | R |
| 2 | RANGE | Output Range. Bitfield used to configure output range for all channels.<br>1: Range 1. VMODE: Output range from 0V to 2 × VREF; IMODE: Output range from 0A to 50mA. | 0x0 | R/W |

## REGISTERS

*Table 33. Bit Descriptions for OUTPUT_CONTROL_0 (Continued)*

| Bits | Bit Name | Description | Reset | Access |
|---|---|---|---|---|
| | | 0: Range 0. VMODE: Output range from 0V to VREF; IMODE: Output range from 0A to 50mA. | | |
| [1:0] | RESERVED | Reserved. | 0x0 | R |

### Reference Control 0 Register

Configures the reference source for all channels

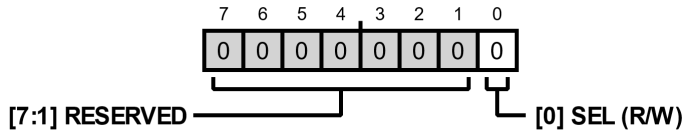


*Figure 92. Address: 0x3C, Reset: 0x00, Name: REFERENCE_CONTROL_0*

*Table 34. Bit Descriptions for REFERENCE_CONTROL_0*

| Bits | Bit Name | Description | Reset | Access |
|---|---|---|---|---|
| [7:1] | RESERVED | Reserved. | 0x0 | R |
| 0 | SEL | Reference Select. Selects the voltage reference source for all channels<br>0: Select 0. VREF pin is an input pin, an external reference should be provided through this pin.<br>1: Select 1. VREF pin is an output pin, an internal reference is used by the part and is also available on VREF pin. | 0x0 | R/W |

### Multiplexer Input Select Register

Selects which of the multiplexer's input signals is monitored on the MUX_OUT pin

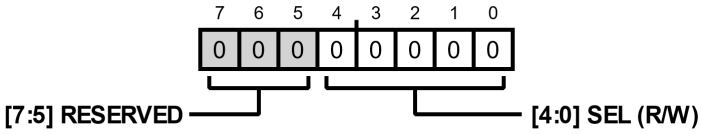


*Figure 93. Address: 0x93, Reset: 0x00, Name: MUX_OUT_SELECT*

*Table 35. Bit Descriptions for MUX_OUT_SELECT*

| Bits | Bit Name | Description | Reset | Access |
|---|---|---|---|---|
| [7:5] | RESERVED | Reserved. | 0x0 | R |
| [4:0] | SEL | Multiplexer Input Select. Selects which of the multiplexer's input signals is monitored on the MUX_OUT pin. Invalid selection does not change MUX SEL value.<br>0x0: Power down. MUX_OUT pin is at unmonitored state or powered down.<br>0x1: Sense VOUT0.<br>0x2: Sense IOUT0 through a sense resistor.<br>0x3: Sense PVDD0 for DAC0.<br>0x4: Sense VOUT1.<br>0x5: Sense IOUT1 through a sense resistor.<br>0x6: Sense PVDD0 for DAC1.<br>0x7: Sense VOUT2.<br>0x8: Sense IOUT2 through a sense resistor.<br>0x9: Sense PVDD0 for DAC2. | 0x0 | R/W |

## REGISTERS

*Table 35. Bit Descriptions for MUX_OUT_SELECT (Continued)*

| Bits | Bit Name | Description | Reset | Access |
|---|---|---|---|---|
| | | 0xA: Sense VOUT3.<br>0xB: Sense IOUT3 through a sense resistor.<br>0xC: Sense PVDD0 for DAC3.<br>0xD: Sense VOUT4.<br>0xE: Sense IOUT4 through a sense resistor.<br>0xF: Sense PVDD1 for DAC4.<br>0x10: Sense VOUT5.<br>0x11: Sense IOUT5 through a sense resistor.<br>0x12: Sense PVDD1 for DAC5.<br>0x13: Sense VOUT6.<br>0x14: Sense IOUT6 through a sense resistor.<br>0x15: Sense PVDD1 for DAC6.<br>0x16: Sense VOUT7.<br>0x17: Sense IOUT7 through a sense resistor.<br>0x18: Sense PVDD1 for DAC7.<br>0x19: Die temperature. Sense voltage to measure internal die temperature.<br>0x1A: MUX_OUT pin internally tied to AGND. | | |

### Status Control 0 Register

Event flags due to start up sequence, interface, reset, and update can be read; write 1 to clear

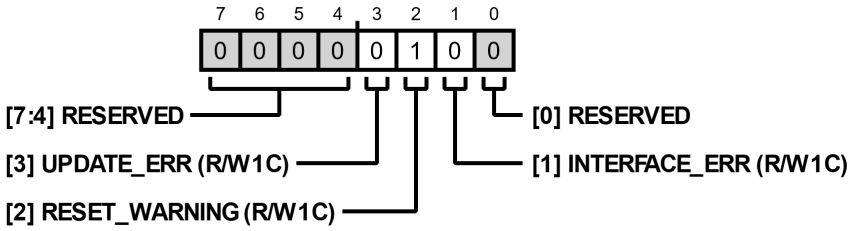


*Figure 94. Address: 0xC2, Reset: 0x04, Name: STATUS_CONTROL_0*

*Table 36. Bit Descriptions for STATUS_CONTROL_0*

| Bits | Bit Name | Description | Reset | Access |
|---|---|---|---|---|
| [7:4] | RESERVED | Reserved. | 0x0 | R |
| 3 | UPDATE_ERR | Update Error. Status to indicate that there was an attempt to update a DAC_CHn within 640ns since the last update.<br>0: Error 0. All updates successful.<br>1: Error 1. Overlapping updates attempted. | 0x0 | R/W1C |
| 2 | RESET_WARNING | Reset Warning. Status to indicate if the device went through a reset event.<br>0: Warning 0. Reset warning flag cleared.<br>1: Warning 1. Reset event occurred. | 0x1 | R/W1C |
| 1 | INTERFACE_ERR | Interface Error. Status to indicate an error flag is asserted in INTERFACE_STATUS_A.<br>0: Error 0. No interface error.<br>1: Error 1. Interface error. | 0x0 | R/W1C |
| 0 | RESERVED | Reserved. | 0x0 | R |

## REGISTERS

### Hardware LDAC Enable 0 Register

Enables hardware LDAC functionality for Channel 0 to Channel 7

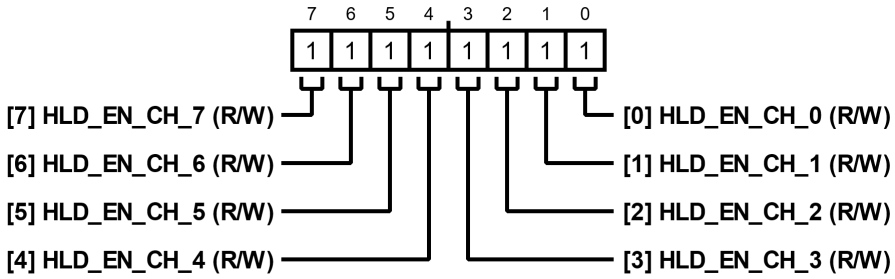


*Figure 95. Address: 0xD0, Reset: 0xFF, Name: HW_LDAC_EN_0*

*Table 37. Bit Descriptions for HW_LDAC_EN_0*

| Bits | Bit Name | Description | Reset | Access |
|---|---|---|---|---|
| 7 | HLD_EN_CH_7 | Hardware LDAC Enable Channel 7. Enable/disable hardware LDAC functionality on Channel 7.<br>0: Disable hardware LDAC on Channel 7.<br>1: Enable hardware LDAC on Channel 7. | 0x1 | R/W |
| 6 | HLD_EN_CH_6 | Hardware LDAC Enable Channel 6. Enable/disable hardware LDAC functionality on Channel 6.<br>0: Disable hardware LDAC on Channel 6.<br>1: Enable hardware LDAC on Channel 6. | 0x1 | R/W |
| 5 | HLD_EN_CH_5 | Hardware LDAC Enable Channel 5. Enable/disable hardware LDAC functionality on Channel 5.<br>0: Disable hardware LDAC on Channel 5.<br>1: Enable hardware LDAC on Channel 5. | 0x1 | R/W |
| 4 | HLD_EN_CH_4 | Hardware LDAC Enable Channel 4. Enable/disable hardware LDAC functionality on Channel 4.<br>0: Disable hardware LDAC on Channel 4.<br>1: Enable hardware LDAC on Channel 4. | 0x1 | R/W |
| 3 | HLD_EN_CH_3 | Hardware LDAC Enable Channel 3. Enable/disable hardware LDAC functionality on Channel 3.<br>0: Disable hardware LDAC on Channel 3.<br>1: Enable hardware LDAC on Channel 3. | 0x1 | R/W |
| 2 | HLD_EN_CH_2 | Hardware LDAC Enable Channel 2. Enable/disable hardware LDAC functionality on Channel 2.<br>0: Disable hardware LDAC on Channel 2.<br>1: Enable hardware LDAC on Channel 2. | 0x1 | R/W |
| 1 | HLD_EN_CH_1 | Hardware LDAC Enable Channel 1. Enable/disable hardware LDAC functionality on Channel 1.<br>0: Disable hardware LDAC on Channel 1.<br>1: Enable hardware LDAC on Channel 1. | 0x1 | R/W |
| 0 | HLD_EN_CH_0 | Hardware LDAC Enable Channel 0. Enable/disable hardware LDAC functionality on Channel 0.<br>0: Disable hardware LDAC on Channel 0.<br>1: Enable hardware LDAC on Channel 0. | 0x1 | R/W |

### Software LDAC Enable 0 Register

Enables software LDAC functionality for Channel 0 to Channel 7

## REGISTERS

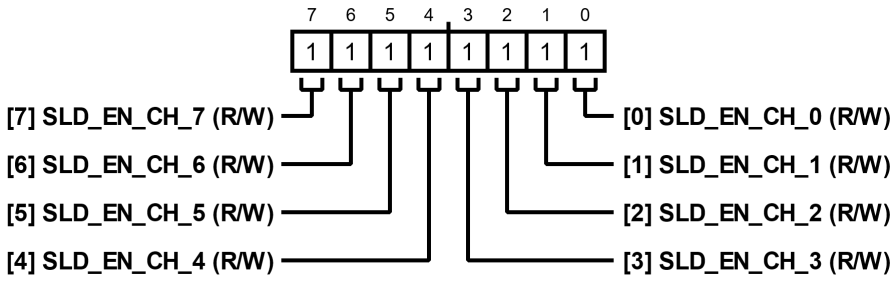


*Figure 96. Address: 0xD1, Reset: 0xFF, Name: SW_LDAC_EN_0*

*Table 38. Bit Descriptions for SW_LDAC_EN_0*

| Bits | Bit Name | Description | Reset | Access |
|---|---|---|---|---|
| 7 | SLD_EN_CH_7 | Software LDAC Enable Channel 7. Enable/disable software LDAC functionality on Channel 7.<br>0: Disable software LDAC on Channel 7.<br>1: Enable software LDAC on Channel 7. | 0x1 | R/W |
| 6 | SLD_EN_CH_6 | Software LDAC Enable Channel 6. Enable/disable software LDAC functionality on Channel 6.<br>0: Disable software LDAC on Channel 6.<br>1: Enable software LDAC on Channel 6. | 0x1 | R/W |
| 5 | SLD_EN_CH_5 | Software LDAC Enable Channel 5. Enable/disable software LDAC functionality on Channel 5.<br>0: Disable software LDAC on Channel 5.<br>1: Enable software LDAC on Channel 5. | 0x1 | R/W |
| 4 | SLD_EN_CH_4 | Software LDAC Enable Channel 4. Enable/disable software LDAC functionality on Channel 4.<br>0: Disable software LDAC on Channel 4.<br>1: Enable software LDAC on Channel 4. | 0x1 | R/W |
| 3 | SLD_EN_CH_3 | Software LDAC Enable Channel 3. Enable/disable software LDAC functionality on Channel 3.<br>0: Disable software LDAC on Channel 3.<br>1: Enable software LDAC on Channel 3. | 0x1 | R/W |
| 2 | SLD_EN_CH_2 | Software LDAC Enable Channel 2. Enable/disable software LDAC functionality on Channel 2.<br>0: Disable software LDAC on Channel 2.<br>1: Enable software LDAC on Channel 2. | 0x1 | R/W |
| 1 | SLD_EN_CH_1 | Software LDAC Enable Channel 1. Enable/disable software LDAC functionality on Channel 1.<br>0: Disable software LDAC on Channel 1.<br>1: Enable software LDAC on Channel 1. | 0x1 | R/W |
| 0 | SLD_EN_CH_0 | Software LDAC Enable Channel 0. Enable/disable software LDAC functionality on Channel 0.<br>0: Disable software LDAC on Channel 0.<br>1: Enable software LDAC on Channel 0. | 0x1 | R/W |

### DAC Register

16-bit data defines the output of OUTn pin, where n is the channel number

DAC_CH0: 0xD2-0xD3
DAC_CH1: 0xD4-0xD5
DAC_CH2: 0xD6-0xD7
DAC_CH3: 0xD8-0xD9
DAC_CH4: 0xDA-0xDB
DAC_CH5: 0xDC-0xDD
DAC_CH6: 0xDE-0xDF
DAC_CH7: 0xE0-0xE1

## REGISTERS

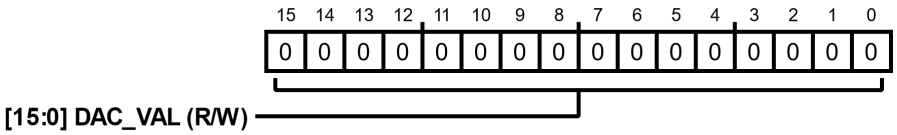


*Figure 97. Address: 0xD2 to 0xE0 (Increments of 2), Reset: 0x0000, Name: DAC_CHn*

*Table 39. Bit Descriptions for DAC_CHn*

| Bits | Bit Name | Description | Reset | Access |
|---|---|---|---|---|
| [15:0] | DAC_VAL | DAC Value. The 16-bit data defines the voltage or current of OUTn pin, where n is the channel number. | 0x0 | R/W |

### Multiple DAC Register

Data written to this register also writes all DAC_CHn selected in MULTI_DAC_Sel_0

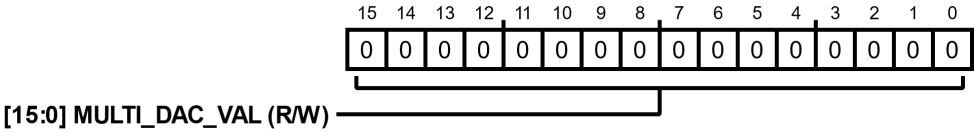


*Figure 98. Address: 0xE2, Reset: 0x0000, Name: MULTI_DAC_CH*

*Table 40. Bit Descriptions for MULTI_DAC_CH*

| Bits | Bit Name | Description | Reset | Access |
|---|---|---|---|---|
| [15:0] | MULTI_DAC_VAL | Multiple DAC Value. Data written to all DAC_CHn selected in MULTI_DAC_Sel_0. Read data always returns the latest data written. | 0x0 | R/W |

### Multiple DAC Select 0 Register

Select which DAC_CHn is written when a write operation is executed on Multi_DAC_CH; applies DAC_CH0 to DAC_CH7 only

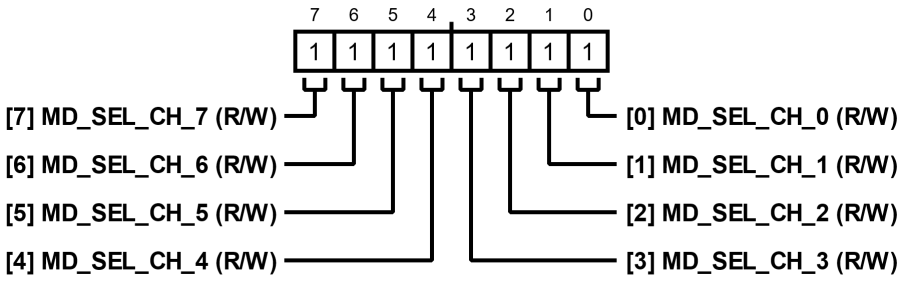


*Figure 99. Address: 0xE4, Reset: 0xFF, Name: MUTLI_DAC_SEL_0*

*Table 41. Bit Descriptions for MULTI_DAC_SEL_0*

| Bits | Bit Name | Description | Reset | Access |
|---|---|---|---|---|
| 7 | MD_SEL_CH_7 | Multiple DAC Select CHannel 7. If selected, write operation on MULTI_DAC_CH also writes DAC_CH7 with the same data. If deselected, write operation on MULTI_DAC_CH does not affect DAC_CH7.<br>0: MD Sel 0. Deselect DAC_CH7 for MULTI_DAC_CH operation.<br>1: MD Sel 1. Select DAC_CH7 for MULTI_DAC_CH operation. | 0x1 | R/W |

## REGISTERS

*Table 41. Bit Descriptions for MULTI_DAC_SEL_0 (Continued)*

| Bits | Bit Name | Description | Reset | Access |
|---|---|---|---|---|
| 6 | MD_SEL_CH_6 | Multiple DAC Select CHannel 6. If selected, write operation on MULTI_DAC_CH also writes DAC_CH6 with the same data. If deselected, write operation on MULTI_DAC_CH does not affect DAC_CH6.<br>0: MD Sel 0. Deselect DAC_CH6 for MULTI_DAC_CH operation.<br>1: MD Sel 1. Select DAC_CH6 for MULTI_DAC_CH operation. | 0x1 | R/W |
| 5 | MD_SEL_CH_5 | Multiple DAC Select CHannel 5. If selected, write operation on MULTI_DAC_CH also writes DAC_CH5 with the same data. If deselected, write operation on MULTI_DAC_CH does not affect DAC_CH5.<br>0: MD Sel 0. Deselect DAC_CH5 for MULTI_DAC_CH operation.<br>1: MD Sel 1. Select DAC_CH5 for MULTI_DAC_CH operation. | 0x1 | R/W |
| 4 | MD_SEL_CH_4 | Multiple DAC Select CHannel 4. If selected, write operation on MULTI_DAC_CH also writes DAC_CH4 with the same data. If deselected, write operation on MULTI_DAC_CH does not affect DAC_CH4.<br>0: MD Sel 0. Deselect DAC_CH4 for MULTI_DAC_CH operation.<br>1: MD Sel 1. Select DAC_CH4 for MULTI_DAC_CH operation. | 0x1 | R/W |
| 3 | MD_SEL_CH_3 | Multiple DAC Select CHannel 3. If selected, write operation on MULTI_DAC_CH also writes DAC_CH3 with the same data. If deselected, write operation on MULTI_DAC_CH does not affect DAC_CH3.<br>0: MD Sel 0. Deselect DAC_CH3 for MULTI_DAC_CH operation.<br>1: MD Sel 1. Select DAC_CH3 for MULTI_DAC_CH operation. | 0x1 | R/W |
| 2 | MD_SEL_CH_2 | Multiple DAC Select CHannel 2. If selected, write operation on MULTI_DAC_CH also writes DAC_CH2 with the same data. If deselected, write operation on MULTI_DAC_CH does not affect DAC_CH2.<br>0: MD Sel 0. Deselect DAC_CH2 for MULTI_DAC_CH operation.<br>1: MD Sel 1. Select DAC_CH2 for MULTI_DAC_CH operation. | 0x1 | R/W |
| 1 | MD_SEL_CH_1 | Multiple DAC Select CHannel 1. If selected, write operation on MULTI_DAC_CH also writes DAC_CH1 with the same data. If deselected, write operation on MULTI_DAC_CH does not affect DAC_CH1.<br>0: MD Sel 0. Deselect DAC_CH1 for MULTI_DAC_CH operation.<br>1: MD Sel 1. Select DAC_CH1 for MULTI_DAC_CH operation. | 0x1 | R/W |
| 0 | MD_SEL_CH_0 | Multiple DAC Select CHannel 0. If selected, write operation on MULTI_DAC_CH also writes DAC_CH0 with the same data. If deselected, write operation on MULTI_DAC_CH does not affect DAC_CH0.<br>0: MD Sel 0. Deselect DAC_CH0 for MULTI_DAC_CH operation.<br>1: MD Sel 1. Select DAC_CH0 for MULTI_DAC_CH operation. | 0x1 | R/W |

### Software LDAC Trigger A Register

Initiates transfer of INPUT_CHn to DAC_CHn and takes effect on enabled channels identified by SW_LDAC_EN_0 only (this register is a copy of SW_LDAC_TRIG_B)

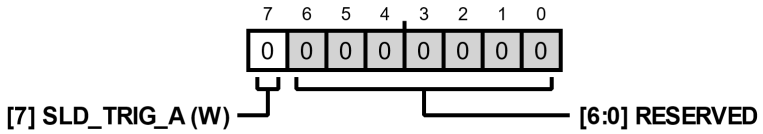


*Figure 100. Address: 0xE5, Reset: 0x00, Name: SW_LDAC_TRIG_A*

*Table 42. Bit Descriptions for SW_LDAC_TRIG_A*

| Bits | Bit Name | Description | Reset | Access |
|---|---|---|---|---|
| 7 | SLD_TRIG_A | Software LDAC Trigger A. When set, initiates transfer of INPUT_CHn to DAC_CHn, where n is the channel number as enabled by SW_LDAC_EN_0. Writing 0 does not have any effect. | 0x0 | W |
| [6:0] | RESERVED | Reserved. | 0x0 | R |

## REGISTERS

### Multiple Input Register

Data written to this register also writes all INPUT_CHn selected in MULTI_INPUT_SEL_0; read data always return the latest data written

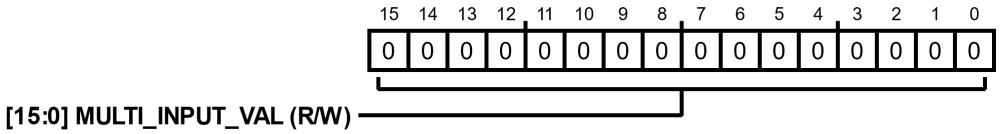


*Figure 101. Address: 0xE6, Reset: 0x0000, Name: MULTI_INPUT_CH*

*Table 43. Bit Descriptions for MULTI_INPUT_CH*

| Bits | Bit Name | Description | Reset | Access |
|---|---|---|---|---|
| [15:0] | MULTI_INPUT_VAL | Multiple Input Value. Data to be written to all INPUT_CHn selected in Multi_INPUT_SEL_0. Read data always returns the latest data written. | 0x0 | R/W |

### Multiple Input Select 0 Register

Select which MULTI_INPUT_SEL_CHn is written during a write operation on MULTI_INPUT_CH; this applies only to MULTI_INPUT_SEL_CH0 to MULTI_INPUT_SEL_CH7

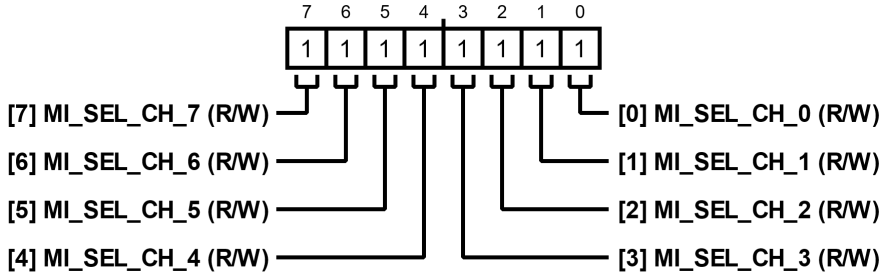


*Figure 102. Address: 0xE8, Reset: 0xFF, Name: MULTI_INPUT_SEL_0*

*Table 44. Bit Descriptions for MULTI_INPUT_SEL_0*

| Bits | Bit Name | Description | Reset | Access |
|---|---|---|---|---|
| 7 | MI_SEL_CH_7 | Multiple Input Select Channel 7. If selected, write operation on MULTI_INPUT_CH also writes INPUT_CH7 with the same data. If deselected, write operation on MULTI_DAC_CH does not affect INPUT_CH7.<br>0: MI Sel 0. Deselect INPUT_CH7 for MULTI_INPUT_CH operation.<br>1: MI Sel 1. Select INPUT_CH7 for MULTI_INPUT_CH operation. | 0x1 | R/W |
| 6 | MI_SEL_CH_6 | Multiple Input Select Channel 6. If selected, write operation on MULTI_INPUT_CH also writes INPUT_CH6 with the same data. If deselected, write operation on MULTI_DAC_CH does not affect INPUT_CH6.<br>0: MI Sel 0. Deselect INPUT_CH6 for MULTI_INPUT_CH operation.<br>1: MI Sel 1. Select INPUT_CH6 for MULTI_INPUT_CH operation. | 0x1 | R/W |
| 5 | MI_SEL_CH_5 | Multiple Input Select Channel 5. If selected, write operation on MULTI_INPUT_CH also writes INPUT_CH5 with the same data. If deselected, write operation on MULTI_DAC_CH does not affect INPUT_CH5.<br>0: MI Sel 0. Deselect INPUT_CH5 for MULTI_INPUT_CH operation.<br>1: MI Sel 1. Select INPUT_CH5 for MULTI_INPUT_CH operation. | 0x1 | R/W |
| 4 | MI_SEL_CH_4 | Multiple Input Select Channel 4. If selected, write operation on MULTI_INPUT_CH also writes INPUT_CH4 with the same data. If deselected, write operation on MULTI_DAC_CH does not affectINPUT_CH4.<br>0: MI Sel 0. Deselect INPUT_CH4 for MULTI_INPUT_CH operation.<br>1: MI Sel 1. Select INPUT_CH4 for MULTI_INPUT_CH operation. | 0x1 | R/W |

## REGISTERS

*Table 44. Bit Descriptions for MULTI_INPUT_SEL_0 (Continued)*

| Bits | Bit Name | Description | Reset | Access |
|---|---|---|---|---|
| 3 | MI_SEL_CH_3 | Multiple Input Select Channel 3. If selected, write operation on MULTI_INPUT_CH also writes INPUT_CH3 with the same data. If deselected, write operation on MULTI_DAC_CH does not affect INPUT_CH3.<br>0: MI Sel 0. Deselect INPUT_CH3 for MULTI_INPUT_CH operation.<br>1: MI Sel 1. Select INPUT_CH3 for MULTI_INPUT_CH operation. | 0x1 | R/W |
| 2 | MI_SEL_CH_2 | Multiple Input Select Channel 2. If selected, write operation on MULTI_INPUT_CH also writes INPUT_CH2 with the same data. If deselected, write operation on MULTI_DAC_CH does not affect INPUT_CH2.<br>0: MI Sel 0. Deselect INPUT_CH2 for MULTI_INPUT_CH operation.<br>1: MI Sel 1. Select INPUT_CH2 for MULTI_INPUT_CH operation. | 0x1 | R/W |
| 1 | MI_SEL_CH_1 | Multiple Input Select Channel 1. If selected, write operation on MULTI_INPUT_CH also writes INPUT_CH1 with the same data. If deselected, write operation on MULTI_DAC_CH does not affect INPUT_CH1.<br>0: MI Sel 0. Deselect INPUT_CH1 for MULTI_INPUT_CH operation.<br>1: MI Sel 1. Select INPUT_CH1 for MULTI_INPUT_CH operation. | 0x1 | R/W |
| 0 | MI_SEL_CH_0 | Multiple Input Select Channel 0. If selected, write operation on MULTI_INPUT_CH also writes MULTI_INPUT_SEL_CH0 with the same data. If deselected, write operation on MULTI_DAC_CH does not affect MULTI_INPUT_SEL_CH0.<br>0: MI Sel 0. Deselect MULTI_INPUT_SEL_CH0 for MULTI_INPUT_CH operation.<br>1: MI Sel 1. Select MULTI_INPUT_SEL_CH0 for MULTI_INPUT_CH operation. | 0x1 | R/W |

### Software LDAC Trigger B Register

Initiates transfer of INPUT_CHn to DAC_CHn and only takes effect on enabled channels identified by SW_LDAC_EN_0, SW_LDAC_EN_1, SW_LDAC_EN_2, and SW_LDAC_EN_3 (this register is a copy of SW_LDAC_TRIG_A)

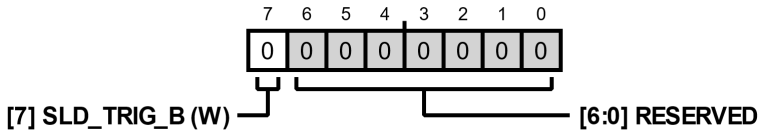


*Figure 103. Address: 0xE9, Reset: 0x00, Name: SW_LDAC_TRIG_B*

*Table 45. Bit Descriptions for SW_LDAC_TRIG_B*

| Bits | Bit Name | Description | Reset | Access |
|---|---|---|---|---|
| 7 | SLD_TRIG_B | Software LDAC Trigger B. When set, initiates transfer of INPUT_CHn to DAC_CHn, where n is the channel number as enabled by SW_LDAC_EN_0. Writing 0 has no effect. | 0x0 | W |
| [6:0] | RESERVED | Reserved. | 0x0 | R |

### Input Register

Writing to this register does not affect the device output; a hardware LDAC or software LDAC is required to push data from INPUT_CHn to DAC_CHn, which also updates the output

INPUT_CH0: 0xEA-0xEB
INPUT_CH1: 0xEC-0xED
INPUT_CH2: 0xEE-0xEF
INPUT_CH3: 0xF0-0xF1
INPUT_CH4: 0xF2-0xF3
INPUT_CH5: 0xF4-0xF5
INPUT_CH6: 0xF6-0xF7
INPUT_CH7: 0xF8-0xF9

## REGISTERS

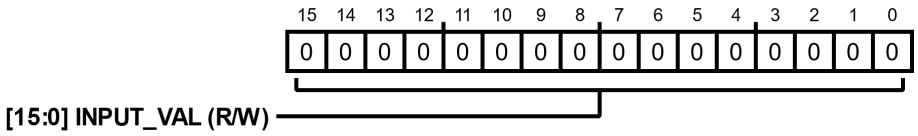


*Figure 104. Address: 0xEA to 0xF8 (Increments of 2), Reset: 0x0000, Name: INPUT_CHn*

*Table 46. Bit Descriptions for INPUT_CHn*

| Bits | Bit Name | Description | Reset | Access |
|---|---|---|---|---|
| [15:0] | INPUT_VAL | Input Value. 16-bit INPUT_CHn data where n is the channel number. | 0x0 | R/W |

### Channel Voltage Mode Enable

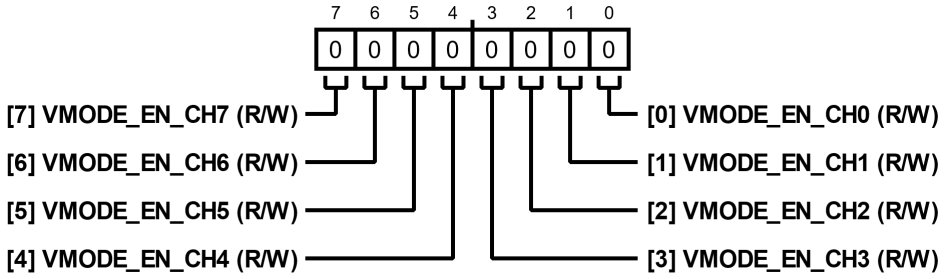


*Figure 105. Address: 0xFF, Reset: 0x00, Name: CHN_VMODE_EN*

*Table 47. Bit Descriptions for CHN_VMODE_EN*

| Bits | Bit Name | Description | Reset | Access |
|---|---|---|---|---|
| 7 | VMODE_EN_CH7 | VMODE enable register..<br>0: Set DAC to IMODE.<br>1: Set DAC to VMODE. | 0x0 | R/W |
| 6 | VMODE_EN_CH6 | VMODE enable register.<br>0: Set DAC to IMODE.<br>1: Set DAC to VMODE. | 0x0 | R/W |
| 5 | VMODE_EN_CH5 | VMODE enable register<br>0: Set DAC to IMODE.<br>1: Set DAC to VMODE. | 0x0 | R/W |
| 4 | VMODE_EN_CH4 | VMODE enable register.<br>0: Set DAC to IMODE.<br>1: Set DAC to VMODE. | 0x0 | R/W |
| 3 | VMODE_EN_CH3 | VMODE enable register.<br>0: Set DAC to IMODE.<br>1: Set DAC to VMODE. | 0x0 | R/W |
| 2 | VMODE_EN_CH2 | VMODE enable register.<br>0: Set DAC to IMODE.<br>1: Set DAC to VMODE. | 0x0 | R/W |
| 1 | VMODE_EN_CH1 | VMODE enable register.<br>0: Set DAC to IMODE.<br>1: Set DAC to VMODE. | 0x0 | R/W |
| 0 | VMODE_EN_CH0 | VMODE enable register<br>0: Set DAC to IMODE. | 0x0 | R/W |

## REGISTERS

*Table 47. Bit Descriptions for CHN_VMODE_EN (Continued)*

| Bits | Bit Name | Description | Reset | Access |
|---|---|---|---|---|
| | | 1: Set DAC to VMODE. | | |

## OUTLINE DIMENSIONS

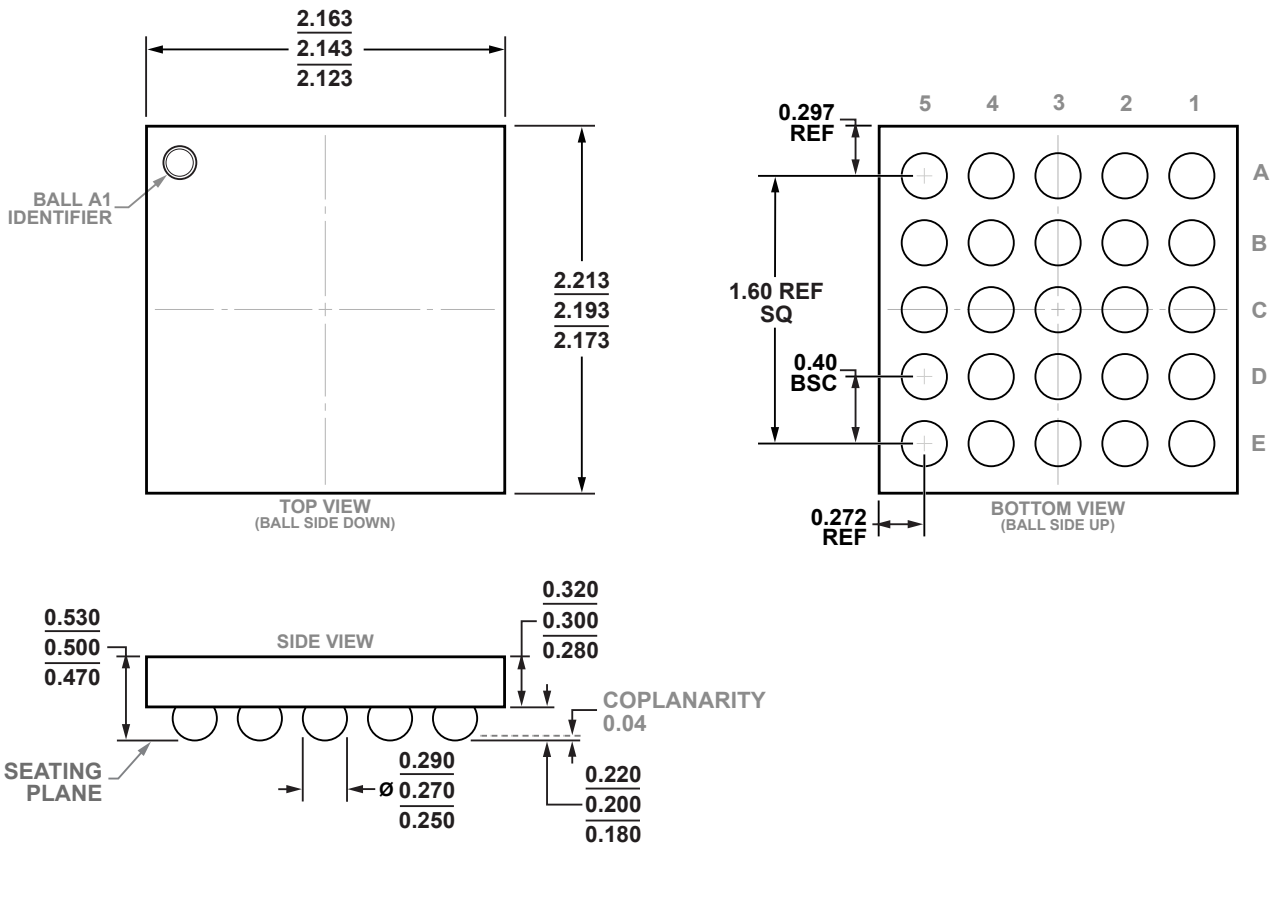


*Figure 106. 25-Ball Wafer Level Chip Scale Package [WLCSP]*
*(CB-25-11)*
*Dimensions shown in millimeters.*

## ORDERING GUIDE

| Model[1] | Temperature Range | Package Description | Packing Quantity | Package Option |
|---|---|---|---|---|
| AD5710RBCBZ-RL7 | −40 to 125°C | 25-Ball WLCSP (2.143mm × 2.193mm × 0.500mm) | Reel, 1500 | CB-25-11 |

[1] Z=RoHS Compliant Part