

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

### FEATURES

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

### APPLICATIONS

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

### GENERAL DESCRIPTION

The AD5706R is a 4-channel, 16-bit resolution, low noise programmable current output, digital-to-analog converter (DAC). The device is ideal for applications such as photonics control and current-mode biasing.

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

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

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

### FUNCTIONAL BLOCK DIAGRAM

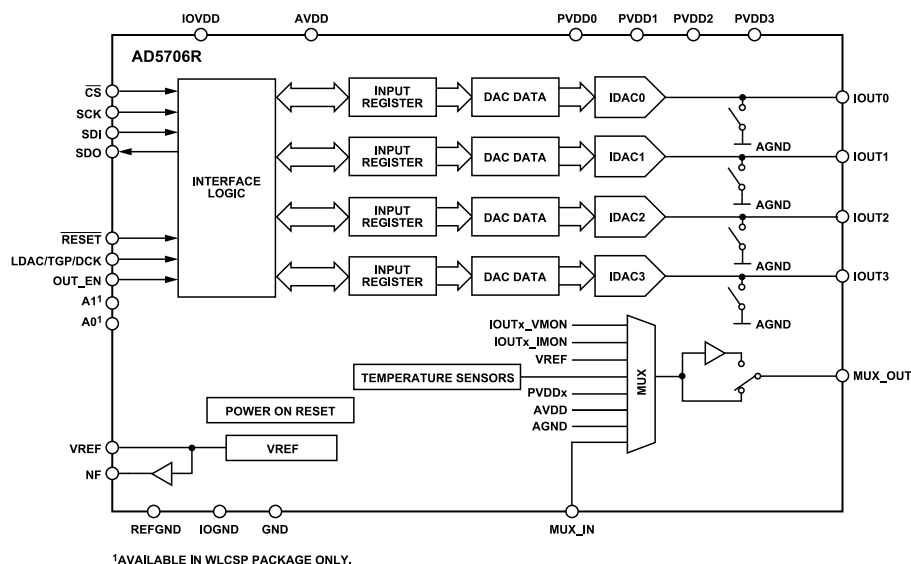


Figure 1. Functional Block Diagram

**TABLE OF CONTENTS**

Features.....	1	Device Addressing.....	32
Applications.....	1	Error Detection.....	33
General Description.....	1	Cyclic Redundancy Check (CRC) Error Detection.....	34
Functional Block Diagram .....	1	Overtemperature Warning Alert.....	36
Specifications.....	3	Device Reset.....	36
DC Specifications.....	3	Scratchpad and User Spare Registers.....	36
AC Specifications.....	5	Device Identification.....	36
Timing Specifications.....	6	DAC Operations.....	37
Absolute Maximum Ratings.....	11	DAC Registers (Data and Input).....	37
Thermal Performance.....	11	IOUT Output Stages.....	37
Thermal Resistance.....	11	DAC Update Modes.....	38
ESD Caution.....	11	Load DAC Function.....	39
Pin Configuration and Function Descriptions.....	12	Toggle DAC Function.....	40
Typical Performance Characteristics.....	14	Dither Function.....	41
DC Performance.....	14	Multi-DAC Function.....	46
AC Performance.....	16	Summary of DAC Modes.....	47
Miscellaneous.....	19	Applications Information.....	48
Terminology.....	21	Recommended External Components.....	48
Theory of Operation.....	24	Power Supply Initialization.....	48
Device Information.....	24	Combining Channels to Increase Current Range .....	48
Precision Reference.....	24	Combining Channels to Reduce Output Noise.....	48
Analog Multiplexer and Diagnostic Monitoring.....	24	Microprocessor Interfacing.....	48
Multiplexer Input and Daisy Chaining.....	26	Layout and Assembly Guidelines.....	49
Power Supplies.....	26	Thermal Considerations.....	49
Digital Interface.....	27	Registers.....	51
Instruction Phase.....	27	Register Summary.....	51
Data Phase.....	27	Register Bitwise Summary.....	52
SPI Frame Synchronization.....	27	Register Details.....	55
Multibyte Registers.....	28	Outline Dimensions.....	74
Register Map Address Direction.....	29	Ordering Guide.....	75
Streaming Mode.....	29		
Single Instruction Mode.....	31		

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

## SPECIFICATIONS

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

## DC SPECIFICATIONS

Table 1. DC Specifications

Parameter <sup>1</sup>	Min	Typ	Max	Unit	Test Conditions/Comments
OPERATING TEMPERATURE RANGE					
	-40			$^{\circ}\text{C}$	Minimum operating ambient temperature ( $T_{A\_MIN}$ )
			125	$^{\circ}\text{C}$	Maximum operating junction temperature ( $T_{J\_MAX}$ )
DAC STATIC PERFORMANCE					
Resolution	16			Bits	All output current ranges
Integral Nonlinearity (INL) <sup>2</sup>	-32		+32	$\text{LSB}_{16}$ <sup>3</sup>	
Differential Nonlinearity (DNL) <sup>2</sup>	-1	$\pm 0.25$	+1	$\text{LSB}_{16}$	Per channel, guaranteed monotonic
Total Unadjusted Error (TUE)	-1		+1	%FSR	
Offset Error <sup>2</sup>	-0.4		+0.4	%FSR	
Offset Error Drift		5		$\text{ppm}/^{\circ}\text{C}$	
Gain Error <sup>2</sup>	-0.9		+0.9	%FSR	
Gain Error Drift		10		$\text{ppm}/^{\circ}\text{C}$	
DC Crosstalk <sup>4</sup>		110		$\mu\text{A}$	Result of a 638mW change in dissipated power, with internal reference
DC Power Supply Rejection Ratio (PSRR)					50mA range, DAC code = 3/4 full scale
AVDD		5		$\mu\text{A}/\text{V}$	AVDD = $3.3\text{V} \pm 10\%$
PVDDx		3		$\mu\text{A}/\text{V}$	PVDD = $3.3\text{V} \pm 10\%$
DAC OUTPUT CHARACTERISTICS					
Output Current ( $I_{\text{OUT}}$ ) Ranges <sup>5</sup>					
	0		+50	mA	
	0		+150	mA	
	0		+200	mA	
	0		+300	mA	
Output Compliance Voltage <sup>6</sup>	0		PVDDx - 0.3	V	
Ground Switch Resistance		10		$\Omega$	Sinking 100mA from IOUTx to GND
INTERNAL REFERENCE OUTPUT					
Voltage ( $V_{\text{REF}}$ )	2.49	2.5	2.51	V	Internal reference enabled, decoupling $1\mu\text{F}$ $T_A = 25^{\circ}\text{C}$
Temperature Coefficient		15		$\text{ppm}/^{\circ}\text{C}$	Across operating temperature range
Output Current Load Capability		+5		mA	Across operating temperature range
Line Regulation		2		$\mu\text{V}/\text{V}$	Due to change in AVDD
Load Regulation, Sourcing		2		$\text{ppm}/\text{mA}$	$\Delta V_{\text{OUT}}/\Delta I_{\text{LOAD}}$
EXTERNAL REFERENCE INPUT					
Input Voltage	2.495	2.5	2.505	V	Internal reference disabled (default) $T_A = 25^{\circ}\text{C}$
Input Impedance		16		k $\Omega$	
TEMPERATURE SENSING DIODE					
Accuracy		5		$^{\circ}\text{C}$	
Reading		1.395		V	MUX_OUT reading at $T_A = 25^{\circ}\text{C}$
Temperature Coefficient		-3.8		$\text{mV}/^{\circ}\text{C}$	
Range	-40			$^{\circ}\text{C}$	$T_A$
			125	$^{\circ}\text{C}$	$T_J$

## SPECIFICATIONS

Table 1. DC Specifications (Continued)

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

<sup>1</sup> See the [Terminology](#) section.<sup>2</sup> Linearity is defined from Code 384 to Code 65535. Offset Error is computed at Code 384.<sup>3</sup> LSB size varies with current output range.<sup>4</sup> Crosstalk is measured with a step change from 0mA to 300mA on all aggressor channels, with PVDD = 3.3V and R<sub>LOAD</sub> = 3.9Ω. Total power dissipation change is 638mW. The monitored channel is held at midscale on 300mA range, with AVDD = 3.5V, PVDD = 3.3V, and R<sub>LOAD</sub> = 8.33Ω.<sup>5</sup> See [Table 15](#).<sup>6</sup> When sourcing current, the output compliance voltage is the maximum voltage at the IOUTx pin, for which the output current is within 0.1% of the measured full-scale range.<sup>7</sup> See the [Timing Specifications](#) section for more details.<sup>8</sup> PVDD must either be tied to AVDD or sourced to a lower level than AVDD.

## SPECIFICATIONS

## AC SPECIFICATIONS

Table 2. AC Specifications

Parameter <sup>1</sup>	Min	Typ	Max	Unit	Test Conditions/Comments
DAC DYNAMIC PERFORMANCE					
Output Current Settling Time					DAC code changed from zero scale to full scale and vice-versa, with settling to $\pm 4\text{LSB}_{16}$ <sup>2</sup>
		9		$\mu\text{s}$	0mA to +200mA range
		11		$\mu\text{s}$	0mA to +50mA range
Slew Rate					$I_{\text{OUT}} = 10\%$ to $90\%$ of FSR and vice-versa
		285		$\text{mA}/\mu\text{s}$	0mA to +200mA range
		67		$\text{mA}/\mu\text{s}$	0mA to +50mA range
Digital-to-Analog Glitch Energy		6		$\text{pA}\cdot\text{sec}$	
Digital Feedthrough (Current Area)		0.03		$\text{nA}\cdot\text{sec}$	
Digital Crosstalk		0.03		$\text{nA}\cdot\text{sec}$	
DAC to DAC Crosstalk (Current Area)					For 200mA range on adjacent DAC channel
		40		$\text{pA}\cdot\text{sec}$	DAC code change from 1/4 scale to 3/4 scale in a step change and vice-versa
		70		$\text{pA}\cdot\text{sec}$	DAC code changed from zero scale to full scale in a step change and vice-versa
MUX to DAC Crosstalk (Current Area)					DAC code = midscale
		0.03		$\text{nA}\cdot\text{sec}$	0mA to +300mA range
AC PSRR					PVDD=3V, refer Figure 37 and Figure 38.
PVDDx		-72		$\text{dB}$	$f = 100\text{kHz}$ , 0mA to +300mA range
		-80		$\text{dB}$	$f = 100\text{kHz}$ , 0mA to +50mA range
AVDD		-82		$\text{dB}$	$f = 10\text{kHz}$
Output Noise Spectral Density (NSD)					DAC code = 3/4 full scale, no external filter
		7.7		$\text{nA}/\sqrt{\text{Hz}}$	1 kHz, 0mA to +300mA range
		6		$\text{nA}/\sqrt{\text{Hz}}$	10 kHz, 0mA to +300mA range
		5.3		$\text{nA}/\sqrt{\text{Hz}}$	1 kHz, 0mA to +200mA range
		4.1		$\text{nA}/\sqrt{\text{Hz}}$	10 kHz, 0mA to +200mA range
		4.2		$\text{nA}/\sqrt{\text{Hz}}$	1 kHz, 0mA to +150mA range
		3.2		$\text{nA}/\sqrt{\text{Hz}}$	10 kHz, 0mA to +150mA range
		2.2		$\text{nA}/\sqrt{\text{Hz}}$	1 kHz, 0mA to +50mA range
		1.2		$\text{nA}/\sqrt{\text{Hz}}$	10 kHz, 0mA to +50mA range
Output Noise (0.1Hz to 10Hz)					DAC code = 3/4 full scale, no external filter
		620		$\text{nA rms}$	0mA to +300mA range
		425		$\text{nA rms}$	0mA to +200mA range
		332		$\text{nA rms}$	0mA to +150mA range
		148		$\text{nA rms}$	0mA to +50mA range
REFERENCE OUTPUT					
Output Voltage Noise (0.1Hz to 10Hz)		4		$\mu\text{V rms}$	Internal reference enabled, decoupling $1\mu\text{F}$
Output Voltage Noise Density		112		$\text{nV}/\sqrt{\text{Hz}}$	$f = 1\text{kHz}$
Turn On Settling Time		500		$\mu\text{s}$	Decoupling $1\mu\text{F}$ , settling to $\pm 0.1\%$
MUX OUTPUT CHARACTERISTICS					
-3dB Input Bandwidth		2.2		$\text{MHz}$	Buffered, $C_{\text{LOAD}} = 3\text{pF}$
		1.4		$\text{MHz}$	Nonbuffered, standalone, $C_{\text{LOAD}} = 3\text{pF}$
Maximum Output Capacitive Load		100		$\text{pF}$	Buffered

<sup>1</sup> See the Terminology section.<sup>2</sup> LSB size varies with current output range.

## SPECIFICATIONS

## TIMING SPECIFICATIONS

AVDD = 2.9V to 3.6V, PVDDx = 1.65V to AVDD, IOVDD = 1.14V to 1.89V, and operating temperature range = -40°C (T<sub>A</sub>) to +125°C (T<sub>J</sub>), unless otherwise noted. All timing values are specified for a standalone device.

All digital input signals are specified with rise time (t<sub>R</sub>) = fall time (t<sub>F</sub>) = 1ns/V (10% to 90% of IOVDD) and timed from a voltage level of (V<sub>IL</sub> + V<sub>IH</sub>)/2.

## Serial Interface Timing

See Figure 2 to Figure 5 for related timing diagrams.

Table 3. Digital Interface Timing Specifications

Parameter	Description	Min	Typ	Max	Unit
t <sub>SCK</sub>	SCK cycle time, write operation. <sup>1</sup>	20			ns
	SCK cycle time, write and read operations. <sup>2</sup>	40			ns
f <sub>SCK</sub>	SCK frequency, write operation. <sup>1</sup>			50	MHz
	SCK frequency, wire and read operations. <sup>2</sup>			25	MHz
t <sub>SCKH_PW</sub>	SCK high time.	8			ns
t <sub>SCKL_PW</sub>	SCK low time.	8			ns
t <sub>CSL_SCKH</sub>	$\overline{\text{CS}}$ falling edge to SCK rising edge setup time	6			ns
t <sub>CSH_SCKH</sub>	$\overline{\text{CS}}$ rising edge to SCK rising edge setup time.	10			ns
t <sub>SCKH_CSH</sub>	Rising edge of SCK <sup>3</sup> to rising edge of $\overline{\text{CS}}$ .	10			ns
t <sub>SCKH_CSL</sub>	Rising edge of SCK to falling edge of $\overline{\text{CS}}$ .	10			ns
t <sub>SDI_SET</sub>	SDI data setup time.	5			ns
t <sub>SDI_HOL</sub>	SDI data hold time.	5			ns
t <sub>CSH_PW</sub>	$\overline{\text{CS}}$ pulse width high.	10			ns
t <sub>SCKL_SDO</sub>	SCK falling edge to SDO data available delay.			15	ns
t <sub>SDO_VALID</sub>	SCK falling Edge to SDO data remains valid.			12	ns
t <sub>CSH_SDOZ</sub>	$\overline{\text{CS}}$ rising edge to SDO disabled.			14	ns
t <sub>SCKL_SDOEN</sub>	SCK falling edge to SDO enabled.			14	ns
t <sub>RESETL_PW</sub>	RESET pulse width low.	20			ns
t <sub>RESET_ACT</sub> <sup>4</sup>	RESET assertion (low level triggered) to channel output getting reset, or rising edge of SCK <sup>3</sup> to channel output getting reset.		<100		ns
t <sub>RESET_DELAY</sub> <sup>5</sup>	RESET pulse delay upon power-on reset.			200	μs

<sup>1</sup> With send status and CRC disabled.

<sup>2</sup> With send status and CRC enabled.

<sup>3</sup> The 8th or 16th SCK, whichever shifts in the final bit of register data in the data phase.

<sup>4</sup> Channel output settling behaviour not taken into consideration.

<sup>5</sup> Guaranteed by design; not production tested.

SPECIFICATIONS

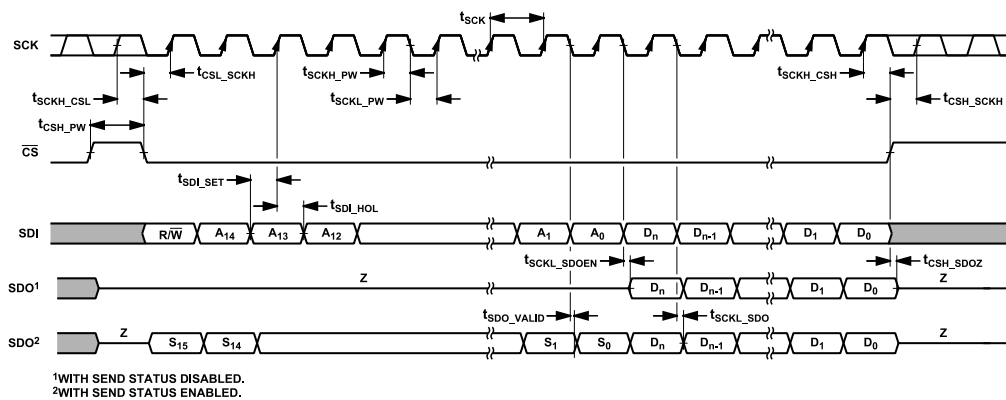


Figure 2. Register Read/Write Timing Diagram (CPOL = 0 and CPHA = 0)

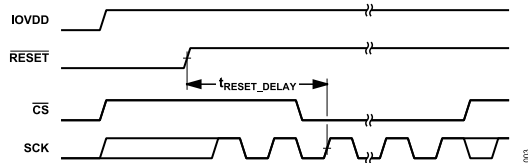


Figure 3. Power On Reset (POR) Timing Diagram

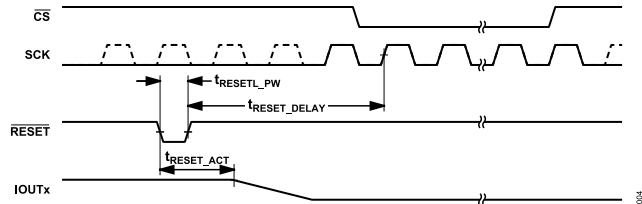


Figure 4. Hardware Reset Timing Diagram

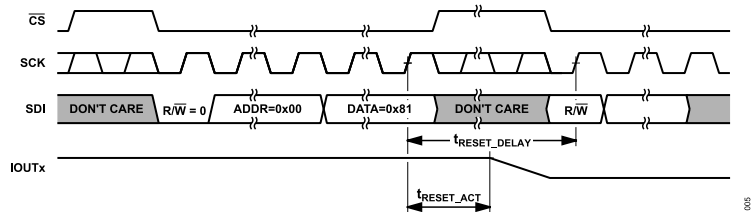


Figure 5. Software Reset Timing Diagram

## SPECIFICATIONS

## DAC Output Update Timing

See Figure 6 to Figure 15 for related timing diagrams.

Table 4. DAC Output Update Timing Specifications

Parameter <sup>1</sup>	Description	Min	Typ	Max	Unit
$t_{OUTEN\_ACT}^2$	Delay from OUT_EN low level trigger to DAC channel in high-Z state (measured on 200mA range)		350		ns
$t_{OUTENL\_SCKH}^3$	Delay from the OUT_EN low level trigger to the rising edge of SCK <sup>4</sup> for transition of IOUTx from high-Z state to normal operation	0			ns
$t_{SCKH\_OUTENL}^3$	Delay from the rising edge of SCK <sup>4</sup> to OUT_EN low for transition of IOUTx to high-Z state	5			ns
$t_{OUTEN\_PW}^3$	OUT_EN pulse width low	6			μs
$t_{DAC\_ACT}^2$	Direct DAC update activation time (measured on 200mA range)		230		ns
$t_{LDAC\_PW}$	LDAC/TGP/DCK pulse width (low or high)	20			ns
$t_{LDACT\_SCKH}$	Delay from the active edge of LDAC/TGP/DCK to the rising edge of SCK <sup>4</sup> in a SPI write transaction during hardware trigger modes	5			ns
$t_{SCKH\_LDACT}$	Delay from the rising edge of SCK <sup>4</sup> to the following active edge of LDAC/TGP/DCK in a SPI write transaction during hardware trigger modes	5			ns
$t_{HW\_LDAC\_TRIG}$	Time between the two consecutive active trigger edges of LDAC/TGP/DCK during hardware trigger modes	80			ns
$t_{SW\_LDAC\_TRIG}$	Delay from one rising edge of SCK <sup>4</sup> to another rising edge of SCK <sup>4</sup> (in the consecutive data phase) in a SPI write transaction during software trigger modes	$16 \times t_{SCK}$			
$t_{LDACT\_SCKH\_RD}$	Delay from active edge of LDAC/TGP/DCK to the rising edge of SCK <sup>4</sup> in a SPI read transaction during hardware trigger modes	5			ns
$t_{SCKH\_LDACT\_RD}$	Delay from the rising edge of SCK <sup>4</sup> to the next active edge of LDAC/TGP/DCK in a SPI read transaction during hardware trigger modes	5			ns
$t_{FUNC\_MODE\_EN1}$	Delay from the active trigger edge of LDAC/TGP/DCK to the rising edge of SCK <sup>4</sup> for change in DAC function modes	5			ns
$t_{FUNC\_MODE\_EN2}$	Delay from the rising edge of SCK <sup>4</sup> to the active trigger edge of LDAC/TGP/DCK for change in DAC function modes	5			ns

<sup>1</sup> See the [DAC Operations](#) section for a list of DAC update modes.

<sup>2</sup> The timing does not include the analog IOUT settling time.

<sup>3</sup> Guaranteed by design; not production tested.

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

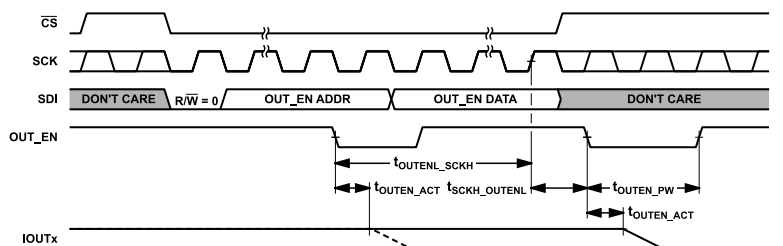


Figure 6. IOUTx Enable/Disable Timing Diagram Using OUT\_EN Pin

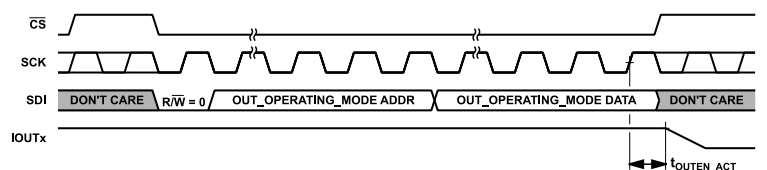


Figure 7. IOUTx Enable/Disable Timing Diagram Using SPI Write

## SPECIFICATIONS

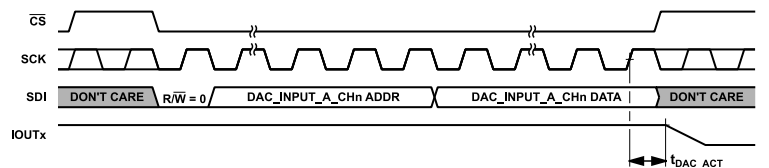


Figure 8. Asynchronous DAC Update Timing Diagram

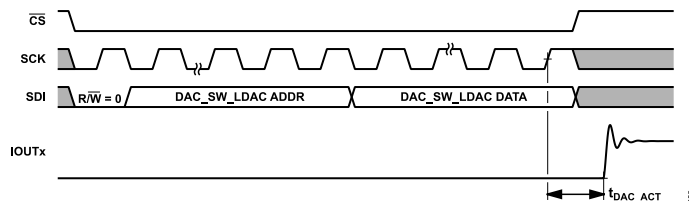


Figure 9. Software LDAC Timing Diagram

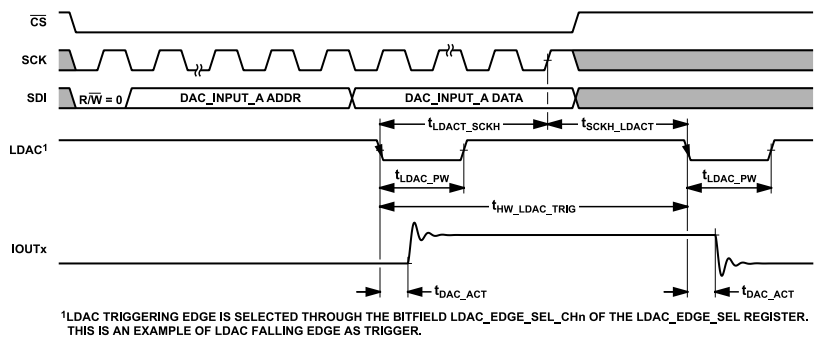
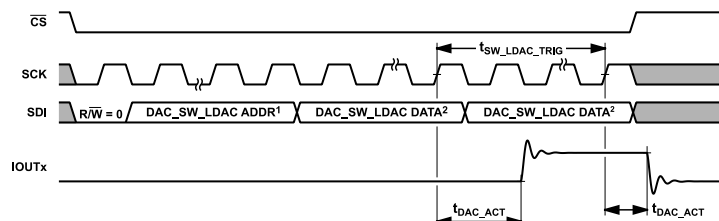
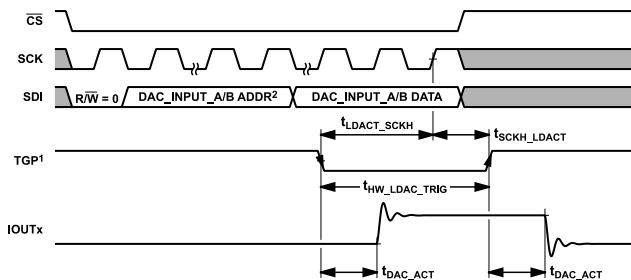


Figure 10. Hardware LDAC Timing Diagram



- <sup>1</sup>ALL SW FUNCTIONS SHARE THE DAC\_SW\_LDAC REGISTER FOR SW TRIGGER.  
<sup>2</sup>INSTEAD OF CREATING TRANSITIONS THROUGH SPI WRITES, SW TOGGLE NEED THE USER TO WRITE '1'b1 TO THE DAC\_INPUT\_A\_SW\_LDAC\_CHn TO INITIATE A DAC UPDATE.  
<sup>3</sup>DAC\_INPUT\_A IS ALWAYS THE SOURCE TO BE USED FOR THE DAC UPDATE OF THE FIRST SW/HW TOGGLE EVENT AFTER ENTERING TOGGLE MODE.

Figure 11. Software DAC Update Timing Diagram in Streaming Mode



- <sup>1</sup>TGP TRIGGER EDGE IS SELECTED THROUGH THE BITFIELD LDAC\_EDGE\_SEL\_CHn OF THE LDAC\_EDGE\_SEL REGISTER. THIS IS AN EXAMPLE OF BOTH EDGES AS TRIGGER EVENTS.  
<sup>2</sup>DAC\_INPUT\_A IS ALWAYS THE FIRST SOURCE TO BE USED FOR DAC UPDATE FOR THE FIRST SW/HW TOGGLE EVENT AFTER ENTERING TOGGLE MODE.

Figure 12. Hardware Toggle Timing Diagram

SPECIFICATIONS

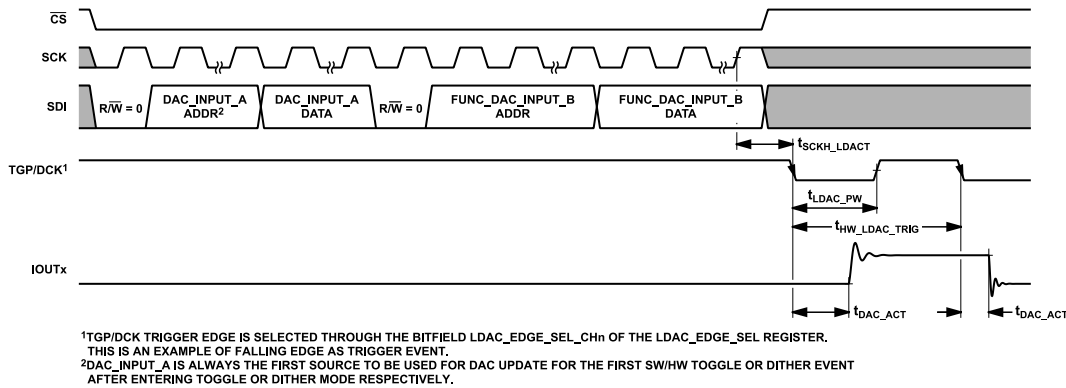


Figure 13. Hardware Toggle and Dither Timing Diagram in Single Instruction Mode

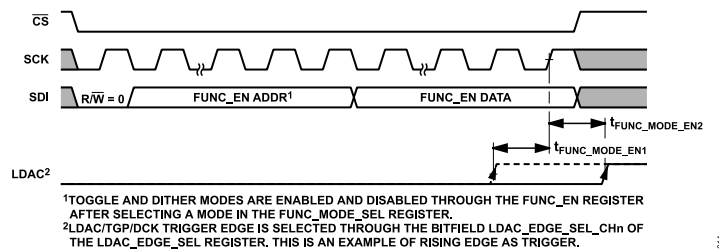


Figure 14. DAC Mode Function Enable Timing Diagram

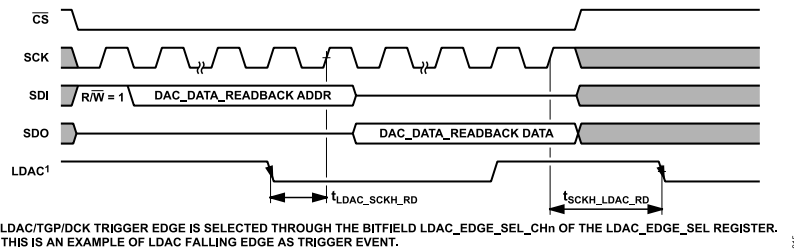


Figure 15. DAC Data Read Back Timing Diagram

## ABSOLUTE MAXIMUM RATINGS

$T_A = 25^\circ\text{C}$ , unless otherwise noted

**Table 5. Absolute Maximum Ratings**

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

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

<sup>2</sup> when in External Reference mode

<sup>3</sup> In LFCSP package, IOGND is same as GND.

<sup>4</sup> See the [Pin Configuration and Function Descriptions](#) section for a list of the digital input and digital output pins.

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

Absolute maximum ratings apply individually only, not in combination.

## THERMAL PERFORMANCE

The AD5706R can be damaged when the junction temperature ( $T_J$ ) limits are exceeded. See [Table 1](#) for the maximum operating junction temperature specification ( $T_{J\text{ MAX}}$ ). Monitoring ambient temperature does not guarantee that  $T_J$  is within the specified maximum temperature limits. In applications with high power dissipation and poor thermal resistance,  $T_J$  must be monitored using the internal temperature sensor.

## THERMAL RESISTANCE

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

Thermal resistance values specified in [Table 6](#) are calculated based on JEDEC specifications and must be used in compliance with JESD51-12. The worst-case junction temperature is reported.

**Table 6. Thermal Resistance**

Package Type <sup>1</sup>	$\theta_{JA}$	$\theta_{JB}$	$\Psi_{JB}$	$\Psi_{JT}$	$\Psi_{JC\text{-TOP}}$	Unit
WLCSP	42.26	7.9	8.1	0.06	0.16	$^\circ\text{C/W}$
LFCSP	39.02	7.88	10.28	0.56	23.32	$^\circ\text{C/W}$

<sup>1</sup> The values in [Table 6](#) are calculated based on the standard JEDEC 2S2P thermal test board with 4 thermal vias. See JEDEC JESD51 series

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

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

Using the parameters listed in [Table 6](#) in accordance with JEDEC standards in the JESD51 series is recommended.

## ESD CAUTION



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

## PIN CONFIGURATION AND FUNCTION DESCRIPTIONS

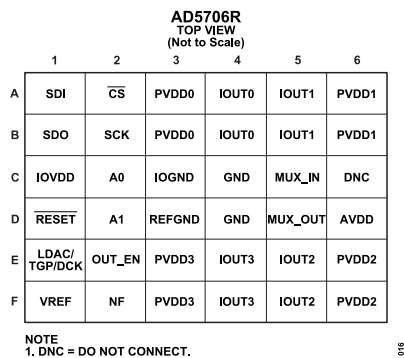


Figure 16. WLCSP Pin Configuration

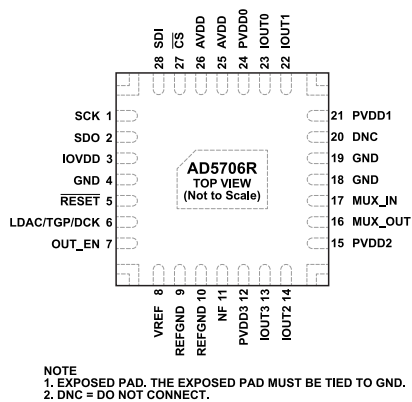


Figure 17. LFCSP Pin Configuration

Table 7. AD5706R Pin Function Descriptions

Pin No.				
WLCSP	LFCSP	Mnemonic	Type <sup>1</sup>	Description
D5	16	MUX_OUT	AO	Analog Output with Internal Buffer. An external analog-to-digital converter (ADC) reads voltages on this pin for diagnostic purposes.
C5	17	MUX_IN	AI	Analog Input. An external input within 0 to VREF range can be fed from another AD5706R MUX_OUT or any other analog output for daisy chaining purposes.
A4, B4	23	IOUT0	AO	Current Output of IDAC0 Channel.
A5, B5	22	IOUT1	AO	Current Output of IDAC1 Channel.
E5, F5	14	IOUT2	AO	Current Output of IDAC2 Channel.
E4, F4	13	IOUT3	AO	Current Output of IDAC3 Channel.
A3, B3	24	PVDD0	S	Power Supply for IDAC0 Channel.
A6, B6	21	PVDD1	S	Power Supply for IDAC1 Channel.
E6, F6	15	PVDD2	S	Power Supply for IDAC2 Channel.
E3, F3	12	PVDD3	S	Power Supply for IDAC3 Channel.
D6	25, 26	AVDD	S	Analog Power Supply.
C1	3	IOVDD	S	Logic Power Supply. IOVDD must be between 1.14V and 1.89V. This pin supplies power to the serial interface circuit blocks on the device.
C4, D4	4, 18, 19	GND	S	Analog Supply Ground Pin.
C3	-	IOGND <sup>2</sup>	S	Digital Supply Ground Pin.
D3	9, 10	REFGND	S	Analog Ground for DAC Core(s) and Internal Reference. REFGND and GND must be tied together at 0V using a low impedance connection.
F1	8	VREF	AI/O	2.5V Voltage Reference Input/Output. Connect a 1μF capacitor between this pin and REFGND.

## PIN CONFIGURATION AND FUNCTION DESCRIPTIONS

Table 7. AD5706R Pin Function Descriptions (Continued)

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

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

<sup>2</sup> IOGND and GND are tied internally in LFCSP package.

## TYPICAL PERFORMANCE CHARACTERISTICS

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

## DC PERFORMANCE

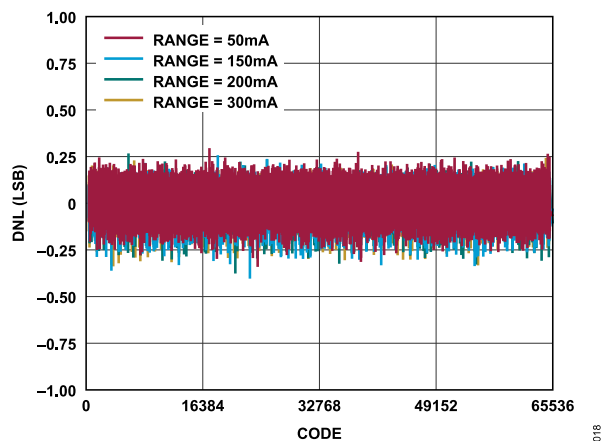


Figure 18. DAC DNL Error vs. DAC Code

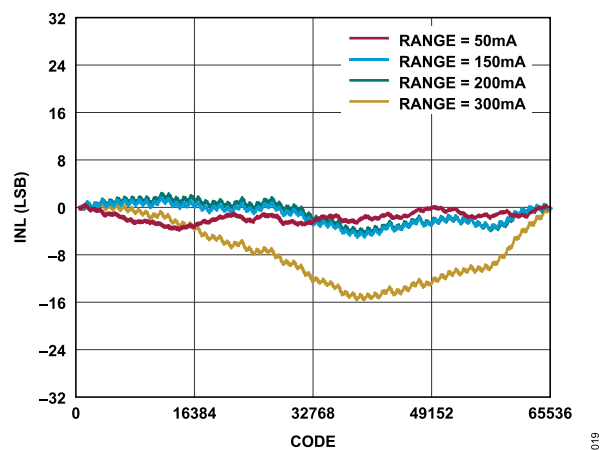


Figure 19. DAC INL Error vs. DAC Code

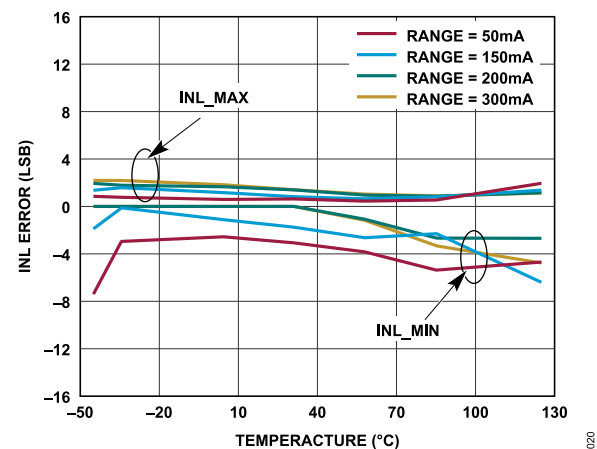


Figure 20. DAC INL Error vs. Temperature

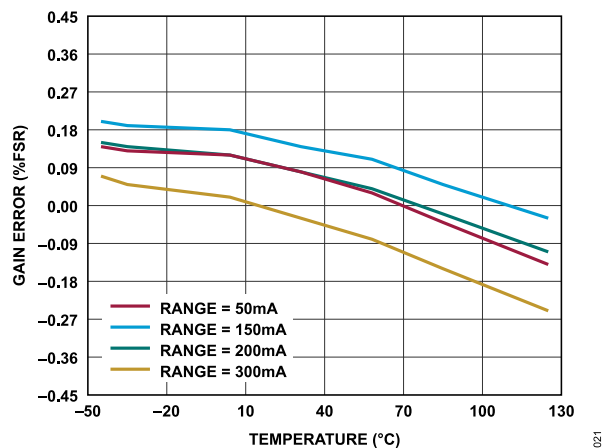


Figure 21. DAC Gain Error vs. Temperature

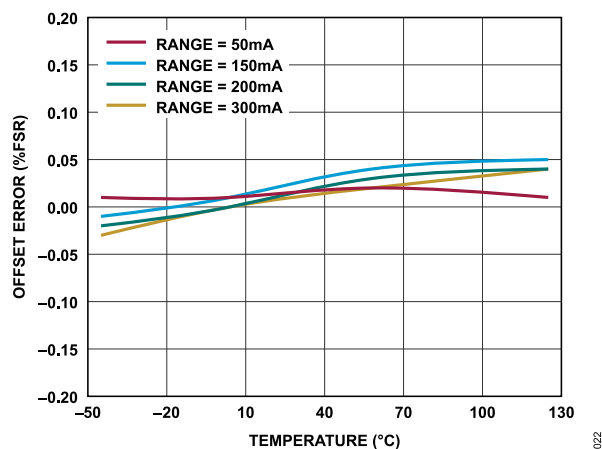


Figure 22. DAC Offset Error vs. Temperature

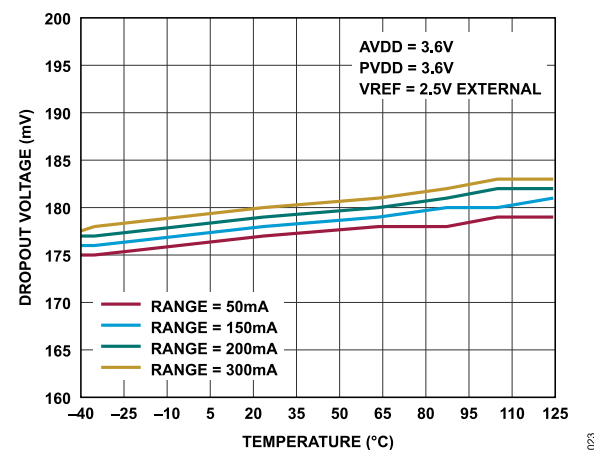


Figure 23. PVDD Dropout Voltage vs. Temperature at PVDD = 3.6V

TYPICAL PERFORMANCE CHARACTERISTICS

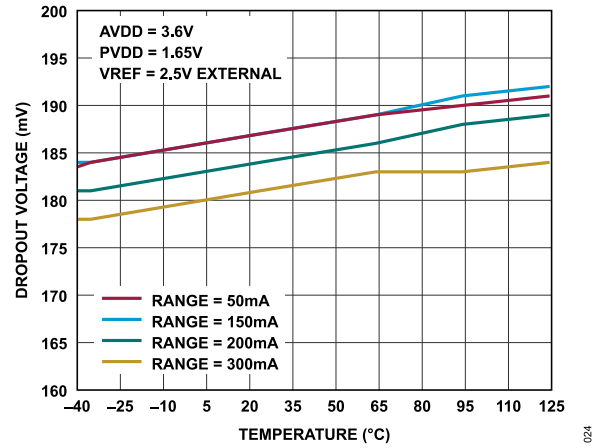


Figure 24. PVDD Dropout Voltage vs. Temperature at PVDD = 1.65V

## TYPICAL PERFORMANCE CHARACTERISTICS

## AC PERFORMANCE

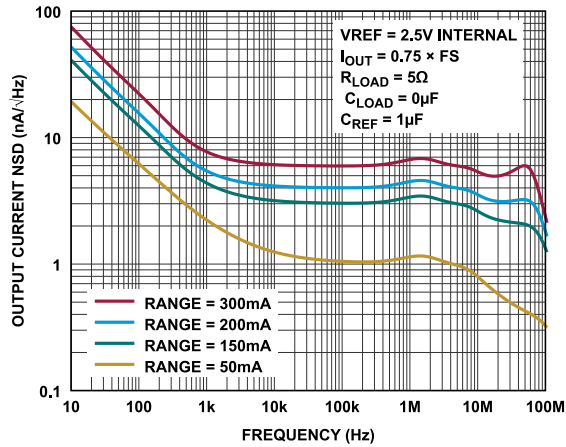


Figure 25. DAC Output Noise Spectral Density vs. Frequency

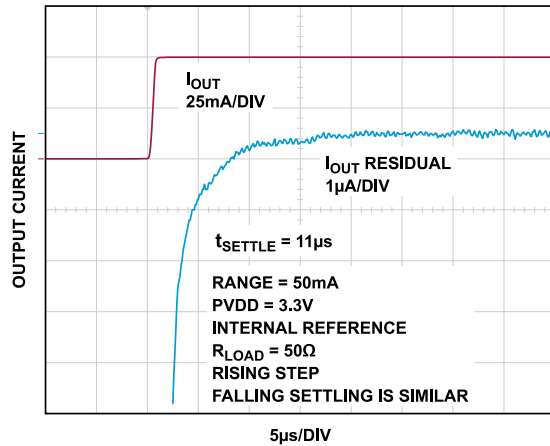


Figure 26. DAC Output Settling Time, 50mA Range

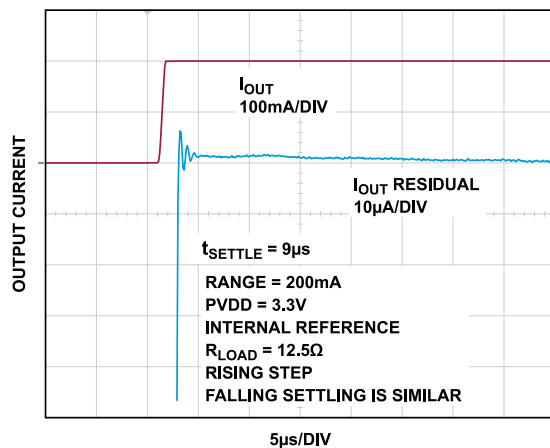


Figure 27. DAC Output Settling Time, 200mA Range

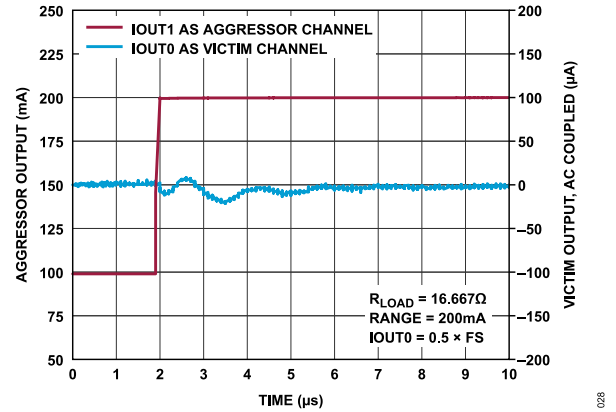


Figure 28. DAC Channel-to-Channel Crosstalk, Positive Output Transition

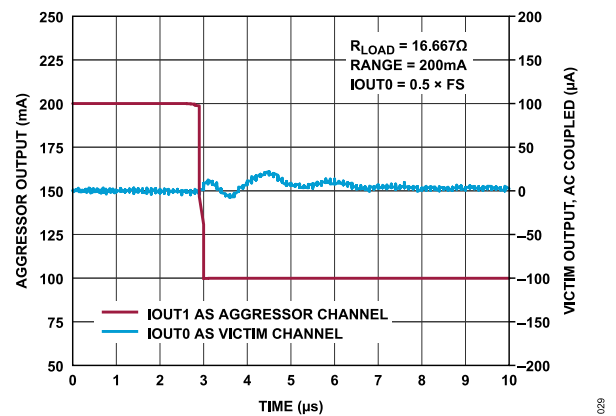


Figure 29. DAC Channel-to-Channel Crosstalk, Negative Output Transition

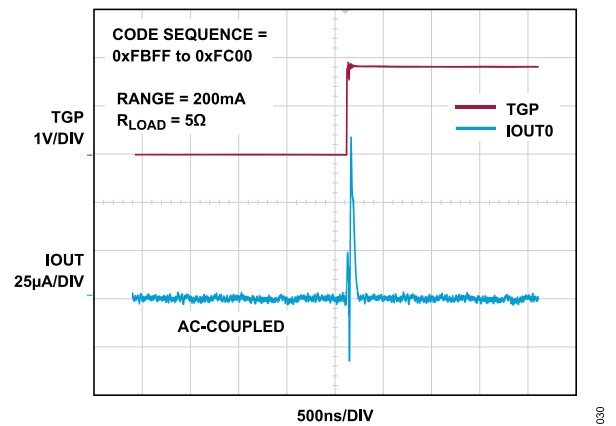


Figure 30. DAC Glitch on Positive Output Transition

## TYPICAL PERFORMANCE CHARACTERISTICS

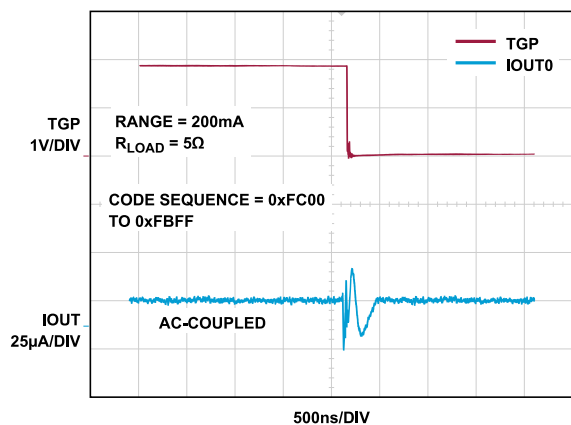


Figure 31. DAC Glitch on Negative Output Transition

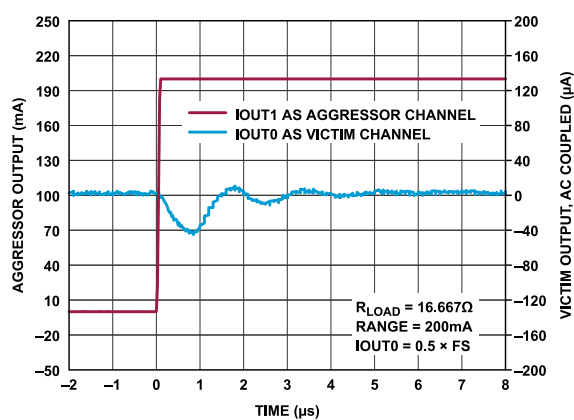


Figure 32. Analog Crosstalk, Positive Output Transition

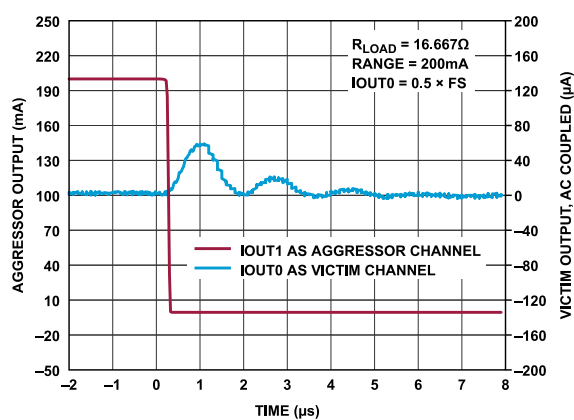


Figure 33. Analog Crosstalk, Negative Output Transition

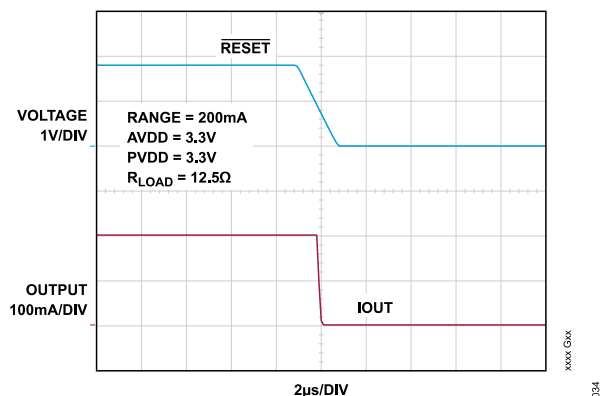


Figure 34. DAC Response Time on Device Reset

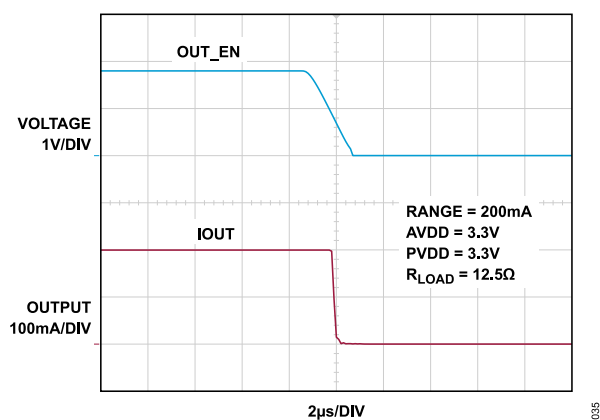


Figure 35. DAC Response Time on Output Disable

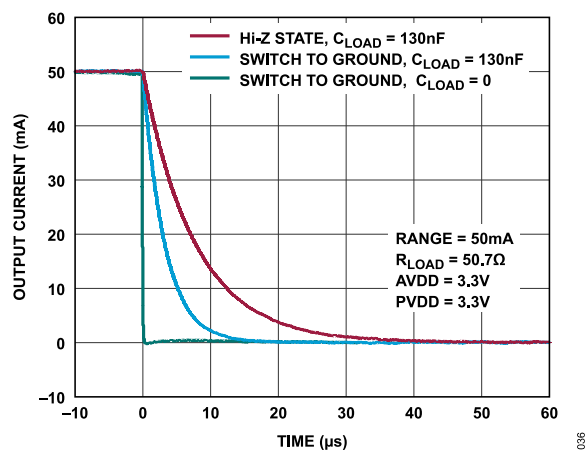


Figure 36. DAC Response on Output Disable over Time

## TYPICAL PERFORMANCE CHARACTERISTICS

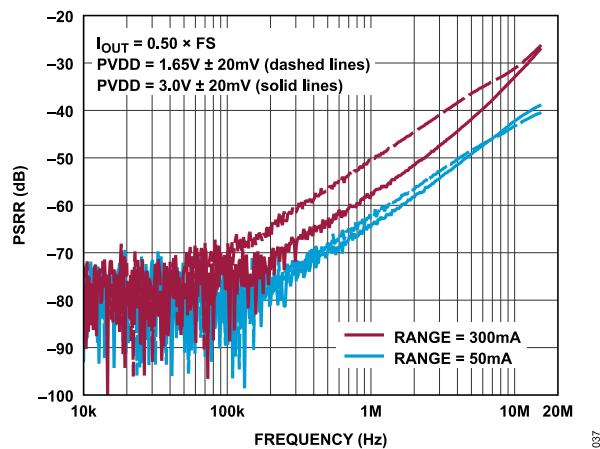
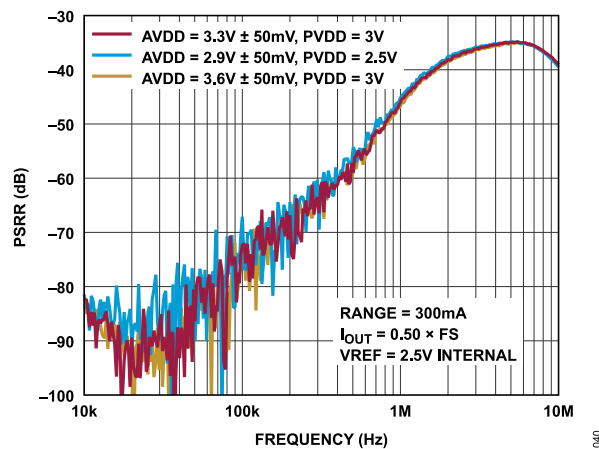
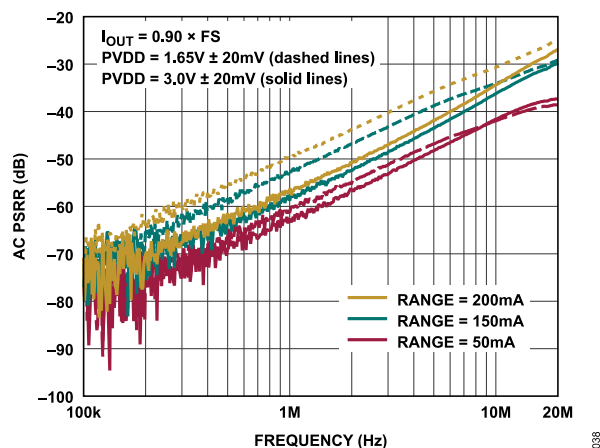
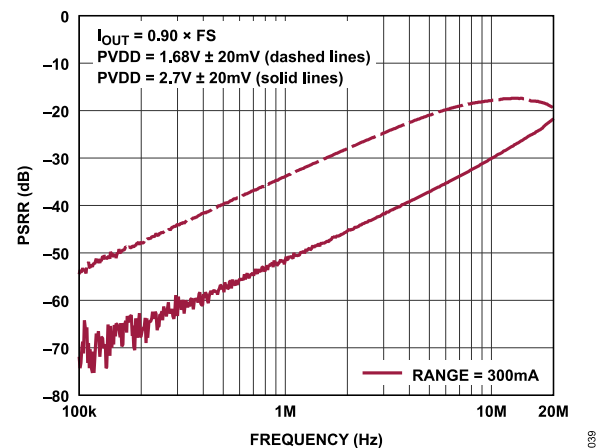
Figure 37. DAC AC PSRR vs. Frequency at DAC Output =  $0.50 \times FS$ 

Figure 40. DAC AC PSRR vs. Frequency Across AVDD Supply

Figure 38. DAC AC PSRR vs. Frequency at DAC Output =  $0.90 \times FS$ Figure 39. DAC AC PSRR vs. Frequency at DAC Output =  $0.90 \times FS$  of 300mA Range

## TYPICAL PERFORMANCE CHARACTERISTICS

## MISCELLANEOUS

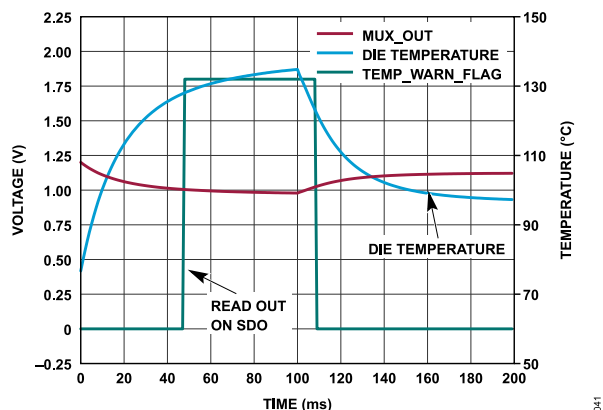


Figure 41. Temperature Sensor Response over Time

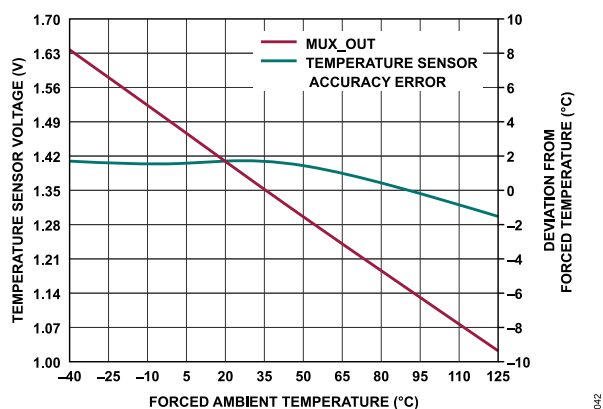


Figure 42. Temperature Sensor Voltage vs. Forced Ambient Temperature

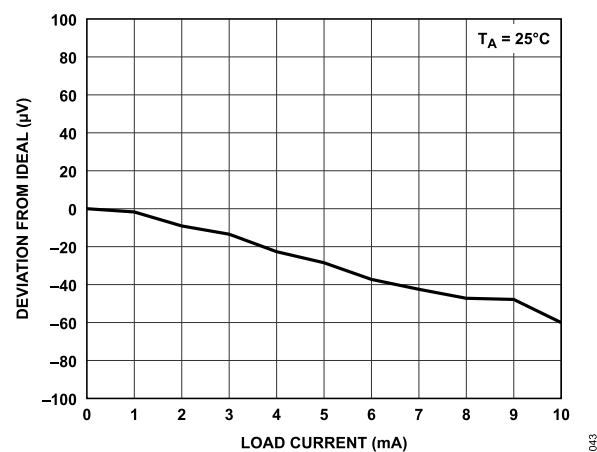


Figure 43. VREF Accuracy Error vs. Load Current

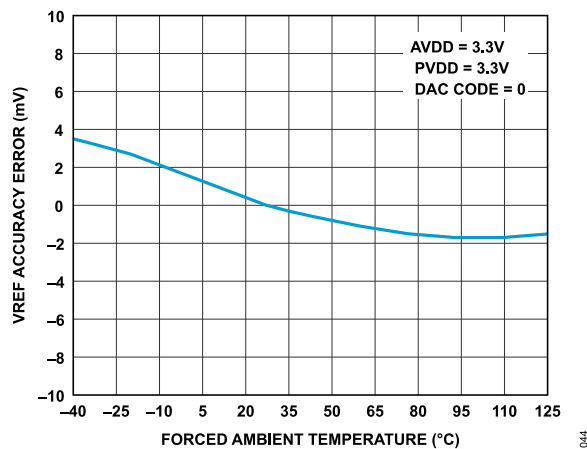


Figure 44. VREF Accuracy Error vs. Forced Ambient Temperature

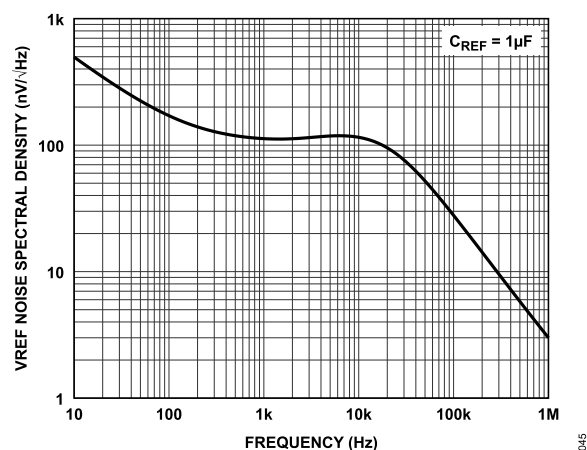


Figure 45. VREF Noise Spectral Density vs. Frequency

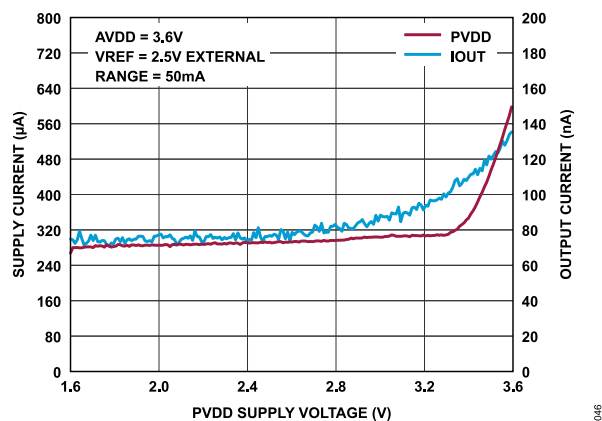


Figure 46. PVDD and IOUT Response in High-Z vs. PVDD Supply Voltage

TYPICAL PERFORMANCE CHARACTERISTICS

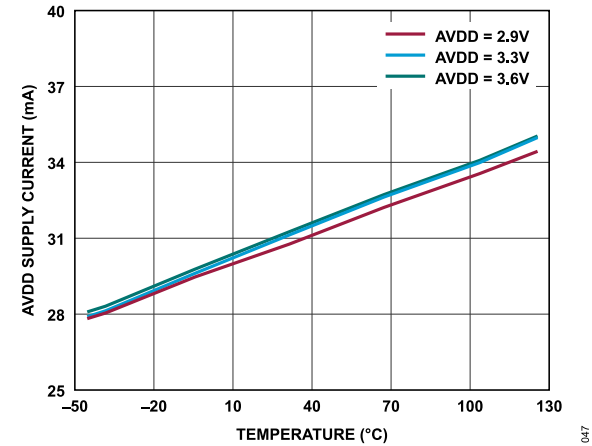


Figure 47. AVDD Supply Current vs. Temperature

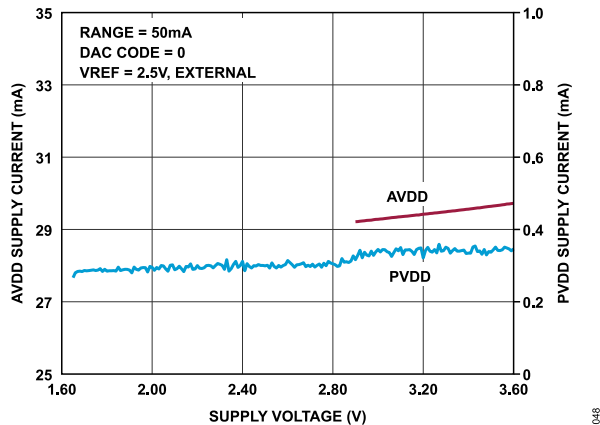


Figure 48. AVDD Supply Current vs. Supply Voltage

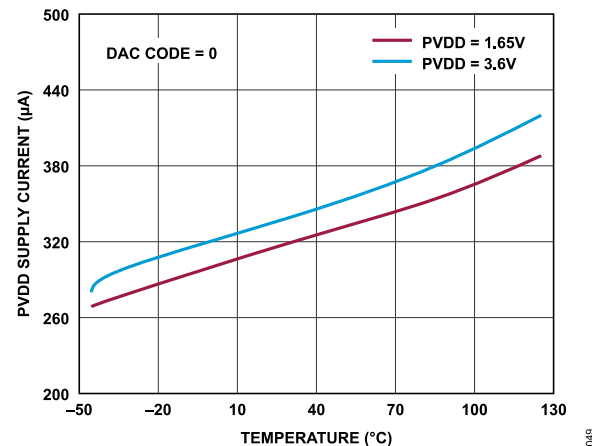


Figure 49. PVDD Supply Current vs. Temperature

## TERMINOLOGY

### DAC

#### Full-Scale Range (FSR)

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

#### Total Unadjusted Error (TUE)

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

#### Relative Accuracy or Integral Nonlinearity (INL)

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

#### Differential Nonlinearity (DNL)

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

#### Gain Error

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

#### Gain Error Drift

Gain error drift is a measurement of the change in gain error with changes in temperature. It is expressed in ppm of FSR/ $^{\circ}$ C.

#### Offset Error

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

#### Offset Error Drift

Offset error drift is a measurement of the change in offset error with a change in temperature. For a current output DAC, the offset error drift is expressed in  $\mu$ A/ $^{\circ}$ C.

#### DC Power Supply Rejection Ratio (PSRR)

PSRR indicates how the output of the DAC is affected by DC changes in the supply voltage. For a current output DAC, PSRR

is the ratio of the change in IOUTx to a change in AVDDx for a full-scale output of the DAC and it's measured in  $\mu$ A/V.

#### AC Power Supply Rejection Ratio (PSRR)

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

#### Output Settling Time

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

#### Digital-to-Analog Glitch Energy

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

#### Digital Feedthrough

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

#### DC Crosstalk

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

#### Digital Crosstalk

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

#### DAC to DAC Crosstalk

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

## TERMINOLOGY

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

### Analog Crosstalk

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

### Output Noise Spectral Density (NSD)

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

### Multiplexer to DAC Crosstalk

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

## TERMINOLOGY

## REFERENCE

## Line Regulation

Line regulation refers to the change in output voltage in response to a given change in supply voltage and is expressed in  $\mu\text{V/V}$ .

## Load Regulation

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

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

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

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

where:

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

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

$V_{\text{REF}, \text{NOM}}$  is the nominal reference voltage output at ambient temperature ( $25^\circ\text{C}$ ).

$\text{TEMP\_RANGE}$  is the difference between the maximum and minimum operating temperature of the reference.

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

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

# THEORY OF OPERATION

## DEVICE INFORMATION

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

## PRECISION REFERENCE

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

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

To achieve optimal noise performance, consider the following:

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

## ANALOG MULTIPLEXER AND DIAGNOSTIC MONITORING

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

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

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

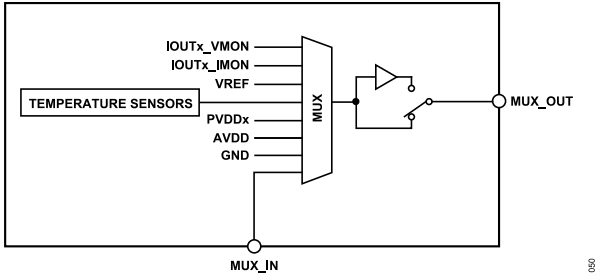


Figure 50. Analog Multiplexer

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

Complete the following to enable diagnostic monitoring:

- ▶ Set the MUX\_OUT\_EN field in the MUX\_OUT\_SEL register.
- ▶ Use the MUX\_PARAM\_SEL register to select which internal signal or channel is routed to the MUX\_OUT pin.

By default, the GND channel is selected in the MUX block. The available channel selections and their corresponding register field values are listed in Table 8.

Table 8. Multiplexer Channels

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

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

## THEORY OF OPERATION

the internal buffer is disabled, the MUX switch impedance can vary between 1kΩ and 20kΩ. In this configuration, the MUX\_OUT pin should only be connected to high-impedance inputs to ensure signal integrity. The available MUX\_OUT configurations are summarized in Table 9, and the corresponding full-scale output voltages for each monitored channel are provided in Table 10.

**Table 9. MUX\_OUT Configuration**

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

**Table 10. Full Scale Output Voltages**

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

### IOU Output Current Monitoring

When the appropriate MUX channel is selected via the MUX\_PAR-AM\_SEL register, the current at the IOUx pin is represented as a voltage on the MUX\_OUT pin. The output voltage range for this representation is 0V to 1.25V.

The output current can be calculated using the following formula:

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

where:

$I_{OUT}$  is the output current at the IOUx pin.

$V_{MUX\_OUT}$  is the voltage at the MUX\_OUT pin.

Note that the above equation is valid only when the IOUx pin is properly loaded and operating within compliance voltage limits. It is not valid if the output is open or dropping out.

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

### Two-Point Calibration

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

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

where:

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

$I_{OUT,X}$  = Expected output currents at DAC code  $DAC\_CODE_X$ .

### IOU Output Voltage Monitoring

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

The conversion from MUX\_OUT voltage to the actual IOUx pin voltage follows the equation:

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

where:

$V_{OUT}$  is the actual voltage at the IOUx pin.

$V_{MUX\_OUT}$  is the voltage measured at the MUX\_OUT pin.

The preceding equation is valid only when the IOUx pin is properly loaded and operating within its compliance voltage range. It is not valid if the output is open or dropping out.

### Internal Die Temperature Monitoring

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

By selecting the appropriate MUX channel in the MUX\_PAR-AM\_SEL register, the output of a temperature sensor can be routed to the MUX\_OUT pin for external measurement.

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

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

where:

$T_J$  is the junction temperature in units of  $^{\circ}\text{C}$ .

$V_{OUT}$  is the voltage at the MUX\_OUT pin.

## THEORY OF OPERATION

### Power Supplies Monitoring

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

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

It is recommended to bypass the internal buffer for GND channel measurement at the MUX\_OUT.

### MULTIPLEXER INPUT AND DAISY CHAINING

The device includes an external input to the analog multiplexer, accessible via the MUX\_IN pin (see Figure 50). This feature enables daisy-chaining of the MUX\_OUT paths when multiple devices are connected to the same SPI bus.

By connecting the MUX\_OUT of one device to the MUX\_IN of the next, a single ADC channel can be shared across multiple AD5706R devices. This configuration simplifies system design and reduces the number of required ADC inputs.

An example of a two-device daisy-chain configuration is shown in Figure 51 for two devices, and the concept can be extended to support more than two devices.

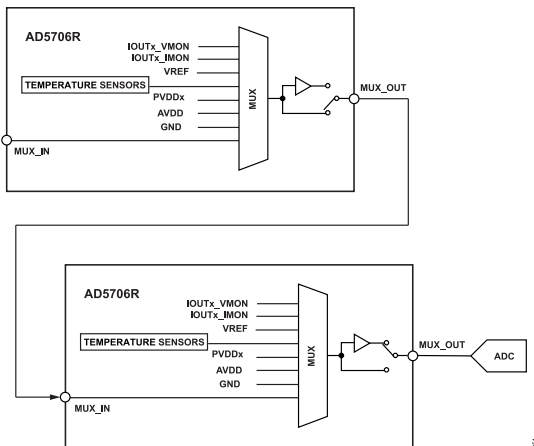


Figure 51. Input Daisy Chaining for Multiple AD5706R Devices

### POWER SUPPLIES

The AD5706R device requires the following supply rails:

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

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

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

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

## DIGITAL INTERFACE

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

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

The SPI interface of the AD5706R supports the transmission mode with CPOL = 0 and CPHA = 0. In this mode, the data on SDI is latched on the rising edges of SCK, while the data is shifted out on SDO on the falling edges of SCK. The data on SDO is valid after the time  $t_{SCK\_SDO}$  from the falling edge of SCK and remains valid for the duration  $t_{SDO\_VALID}$  after the next falling edge of SCK (refer to Table 3 and Figure 2).

Communication with the device is divided into two distinct phases.

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

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

Figure 52 illustrates an SPI transaction phase with a 15-bit address instruction phase. A SPI frame can contain one or more register read and write transactions.

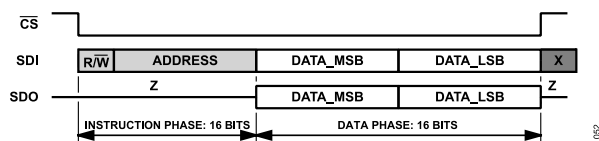


Figure 52. SPI Transaction Phases

### INSTRUCTION PHASE

The instruction phase begins immediately after the falling edge of  $\overline{CS}$ , which initiates the SPI transaction. This phase includes a read/write bit ( $R/\overline{W}$ ) followed by an address word. Setting  $R/\overline{W}$  high selects a read instruction (see Figure 54), while setting  $R/\overline{W}$  low selects a write instruction (see Figure 53). By default, the address word is 15 bits long (15-bit addressing). The register address sent during the instruction phase serves as the starting address for subsequent read or write operations.

### DATA PHASE

The data phase follows directly after the instruction phase, as illustrated in Figure 53 and Figure 54. During a write instruction, data is clocked into the device via the SDI pin on the rising edges of SCK and is written to the selected register location. Conversely, during a read instruction, data from the selected register location is shifted out via the SDO pin on the falling edges of SCK.

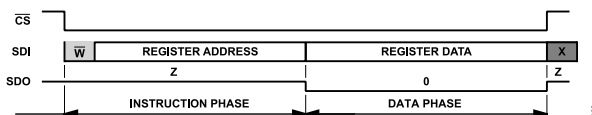


Figure 53. Basic SPI Write Frame

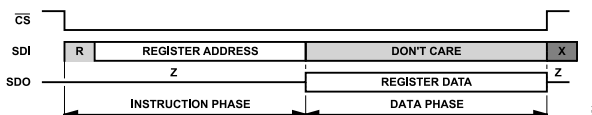


Figure 54. Basic SPI Read Frame

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

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

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

### SPI FRAME SYNCHRONIZATION

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

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

Figure 2 illustrates the detailed timing diagram for executing register read and write via the SPI interface. For the specific timing specifications, refer to Table 3.

DIGITAL INTERFACE

MULTIBYTE REGISTERS

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

- ▶ DAC\_INPUT\_A\_CHn
- ▶ FUNC\_DAC\_INPUT\_B\_CHn
- ▶ MULTI\_DAC\_INPUT\_A\_CHn
- ▶ DAC\_DATA\_READBACK\_CHn

When writing to a multibyte register, all bytes must be accessed in a single SPI transaction. Consequently, the STRICT\_REGISTER\_ACCESS bit in the INTERFACE\_CONFIG\_C register is read only and set to 1. If an SPI write transaction to a multibyte register is attempted on a per byte basis, the register contents are not updated, and the REGISTER\_PARTIAL\_ACCESS\_ERR bit in the INTERFACE\_STATUS\_A register is set. A write transaction to a multibyte register is completed after the 16th SCK rising edge of the data phase, which shifts in the final bit of the register data.

Table 11. Multibyte Registers

Bit	MSB								LSB							
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
16	DATA [15:14] <sup>1</sup>		DATA [13:8]						DATA [7:0]							

<sup>1</sup> These bits are don't care in the dither mode when assigned for FUNC\_DAC\_INPUT\_B\_CHn registers. Refer to the [Dither Function](#) section.

## DIGITAL INTERFACE

## REGISTER MAP ADDRESS DIRECTION

The address direction options control whether the address automatically decrements (address descending) or increments (address ascending) when accessing multiple bytes of data in a single data phase, such as when accessing multibyte registers (see [Figure 55](#) and [Figure 56](#)) or multiple registers with streaming mode.

Address direction is selected using the ADDR\_ASCENSION bit in the INTERFACE\_CONFIG\_A register. If ADDR\_ASCENSION is set to 0, the address decrements after each byte is accessed. If ADDR\_ASCENSION is set to 1, the address increments after each byte is accessed.

The order of the bytes in the SPI frames depends on the ADDR\_ASCENSION bit as follows:

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

For example, the DAC\_INPUT\_A\_CH0 register is two bytes long, with the least significant byte at Address 0x60 and the most significant byte at Address 0x61. If the ADDR\_ASCENSION bit is set to decrement, the address must be 0x61, and the first byte must be the most significant one. Refer to the [Registers](#) section for a complete list of registers.

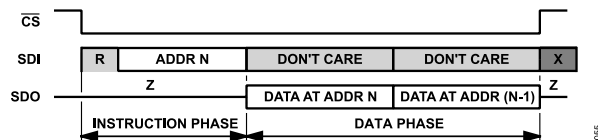


Figure 55. Multibyte Register Address Descending Read Access

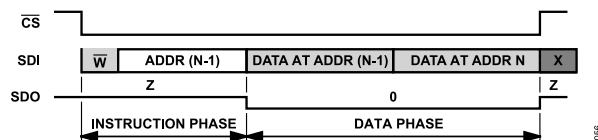


Figure 56. Multibyte Register Address Ascending Write Access

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

## STREAMING MODE

When the SINGLE\_INST bit in the INTERFACE\_CONFIG\_B register is set to 0, single instruction mode is disabled, and streaming mode is enabled. In streaming mode, multiple registers with adjacent addresses can be accessed with a single instruction and data phase, allowing efficient access to contiguous memory regions

(for example, during initial device configuration). The device is configured in streaming mode by default.

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

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

[Figure 57](#) shows the instruction and data phase when using streaming mode with address descending to write to all the DAC input registers. In this example, the starting address is the most significant byte of the DAC\_INPUT\_A\_CH3 register at Address 0x67. To determine the number of data bytes to transfer to consecutive addresses,  $\overline{CS}$  must be brought high at the end of the write transaction (in [Figure 57](#), the end of the write transaction occurs after Address 0x60).

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

## Looping in Streaming Mode

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

If LOOP\_COUNT is set to 0, looping is disabled, and the following occurs:

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

DIGITAL INTERFACE

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

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

Figure 58 shows an example where data is written to the two DAC input registers in streaming mode with the address descending.

The address in the instruction phase is 0x67 and LOOP\_COUNT is set to four, but  $\overline{CS}$  is held low for six bytes, which means that the DAC\_INPUT\_A\_CH3 register is written at the start and end of the transaction. The user must carefully determine when to end the transaction by deasserting  $\overline{CS}$ .

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

By default, LOOP\_COUNT automatically clears itself after the instruction where it is applied. If the user wants the programmed value to hold indefinitely, the user should first set KEEP\_STREAM\_LENGTH\_VAL bit to 1 before programming the LOOP\_COUNT bits.

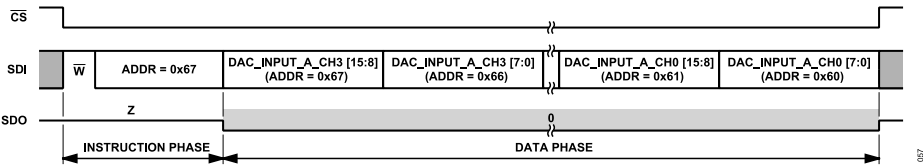


Figure 57. Streaming Mode Write Register Access with Address Descending (CRC Error Detection Disabled)

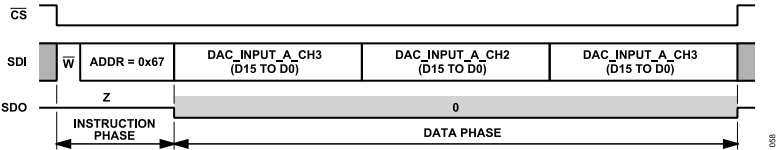


Figure 58. Streaming Mode Write Register Access with Looping and Address Descending (LOOP\_COUNT = 4)

DIGITAL INTERFACE

SINGLE INSTRUCTION MODE

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

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

Figure 59 illustrates an example of an SPI transaction in single instruction mode with the following register accesses:

- 1. Enables internal reference (BANDGAP\_CONTROL register).
- 2. Enables IOUT0 output (OUT\_OPERATING\_MODE register).
- 3. Sets the output range of IOUT0 (OUT\_RANGE\_CH0 register).
- 4. Reads the TEMP\_WARN\_STAT register.

In the Figure 59, BANDGAP\_CONTROL is a single byte register, while OUT\_OPERATING\_MODE, OUT\_RANGE\_CH0, and TEMP\_WARN\_STAT are two-byte registers. Therefore, the data phases containing OUT\_OPERATING\_MODE, OUT\_RANGE\_CH0, and TEMP\_WARN\_STAT have two bytes, making the SPI transactions for these data phases 32 bits wide. In contrast, the SPI transaction for the one-byte register BANDGAP\_CONTROL is 24 bits wide.

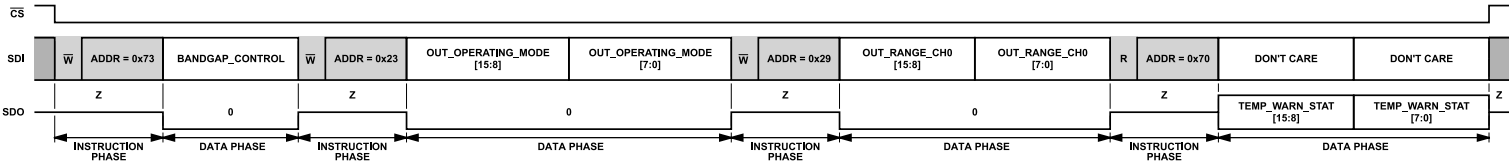


Figure 59. Single Instruction Mode Register Access Example with Address Descending (CRC Error Detection Disabled)

## DIGITAL INTERFACE

## DEVICE ADDRESSING

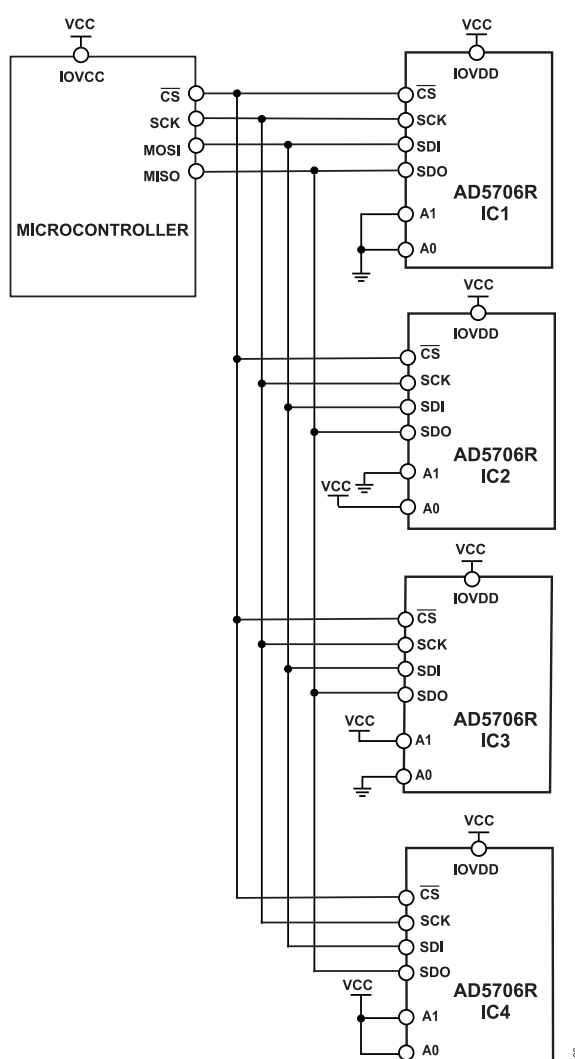
The AD5706R has two address pins, A (1:0), in the WLCSP package. With four unique addresses, it is possible to keep a maximum of four devices on the same SPI bus as shown in [Figure 60](#). Care should be taken to ensure that every device on the same SPI bus has a unique device address on the address pins.

To communicate with any AD5706R, the instruction phase in the SPI frame has three MSB bits, A (14:12), which correspond to the device addresses as indicated in [Figure 2](#) and [Table 3](#). This allows for selecting and writing to a specific device using the corresponding address bits in the instruction phase. An SPI read transaction must address a specific device to read out data on the SDO line.

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 device ignores the SPI read and write transactions if the address pins do not match the address bits in the instruction phase.

**Table 12. Device Addressing Truth Table**

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



**Figure 60. Communication with Multiple AD5706R Devices on a Single SPI Bus**

## DIGITAL INTERFACE

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

## ERROR DETECTION

### Interface\_Status\_A Register

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

### Register Does Not Exist Error

When the host attempts to access an invalid register address, the ADDRESS\_INVALID\_ERR bit is set in the INTERFACE\_STATUS\_A register. To clear this error, write 1 to this bit.

### Partial Register Access Error

The REGISTER\_PARTIAL\_ACCESS\_ERR bit in the INTERFACE\_STATUS\_A register is set when a multibyte register is partially accessed for read or write, meaning that the transaction ends before all bytes of a multibyte register have been accessed. To clear this error, write 1 to the REGISTER\_PARTIAL\_ACCESS\_ERR bit.

### Write to Read only Register Error

If the host tries to write to a read only register, the WR\_TO\_RD\_ONLY\_REG\_ERR bit field is asserted in the INTERFACE\_STATUS\_A register. To clear this error, write 1 to the WR\_TO\_RD\_ONLY\_REG\_ERR bit.

### Invalid/No CRC Received Error

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

### Clock Counting Error

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

## Interface Not Ready Status

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

## Device Status During Instruction Phase

The device has an internal status register that contains all its major status bits, and it can be appended to SPI frames during transactions. The status bits in the internal status register are mapped to the INTERFACE\_STATUS\_A and TEMP\_WARN\_STATUS register bits. When the SEND\_STATUS bit in INTERFACE\_CONFIG\_C register is enabled, the contents of the status register are read out on the SDO pin as shown in Figure 61. It is recommended to use this feature on a single device to avoid issuing separate SPI commands to read the mapped registers. This feature should not be used for multiple devices sharing the same SPI bus.

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

The internal status register is 16-bit wide and is described in Table 13. The readout contents of the status register depend on the SHORT\_INSTRUCTION bit of the INTERFACE\_CONFIG\_B register. If the SHORT\_INSTRUCTION is set to 1, the Bits[7:0] of the status register are read out on the SPI SDO. If the SHORT\_INSTRUCTION is set to 0 (15-bit addressing), the Bits[15:0] of the status register are read out on the SPI SDO.

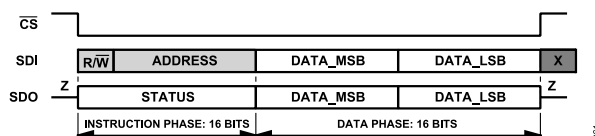


Figure 61. SPI Transaction Phases with SEND\_STATUS Bit Enabled and 15-Bit Addressing Mode

Table 13. Internal Status Register Bit Details

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

## DIGITAL INTERFACE

## CYCLIC REDUNDANCY CHECK (CRC) ERROR DETECTION

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

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

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

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

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

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

Figure 62 and Figure 63 show how a CRC code is appended to the write or read, respectively, for the digital host or the device to validate the data. For register writes, the host must generate the CRC using the calculation described in Equation 7. For register reads, the host must also send the correct CRC byte that is checked by the device. The first byte of data sent contributes to the CRC calculation; therefore, a value of 0x00 is recommended. In the same read transaction, the device provides the CRC code for the digital host to verify.

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

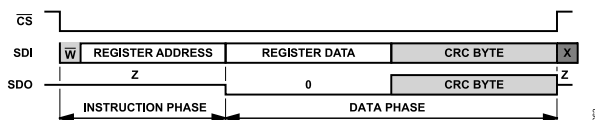


Figure 62. Basic SPI Write Frame (CRC Error Detection Enabled)

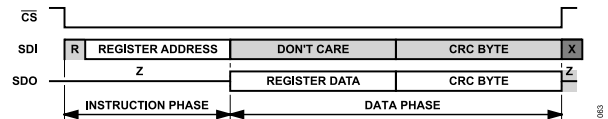


Figure 63. Basic SPI Read Frame (CRC Error Detection Enabled)

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

Table 14 shows the seed value used in the CRC code calculation and how it is transmitted for both single instruction mode and streaming mode. When using single instruction mode, every CRC code in a SPI frame uses 0xA5 as the seed value to prevent stuck-at fault conditions for Address 0x0000.

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

Figure 64 shows an example where CRC is enabled in the first SPI transaction, which demonstrates how the CRC codes are provided by the device during a read transaction and by the host during a write transaction. In this example, the device is configured in single instruction mode with the address descending. This example shows the following sequence:

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

DIGITAL INTERFACE

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

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

When CRC error detection is enabled, do not use streaming mode, including looping, if the range of registers to be addressed includes

unused and reserved registers. See [Figure 65](#) and [Figure 66](#) for examples of write and read SPI transactions in streaming mode, particularly for multibyte registers.

Table 14. CRC Seed Values

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

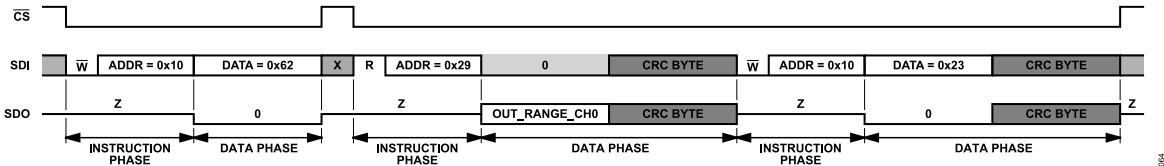


Figure 64. CRC Code SPI Transactions Example (Address Descending, Single Instruction Mode)

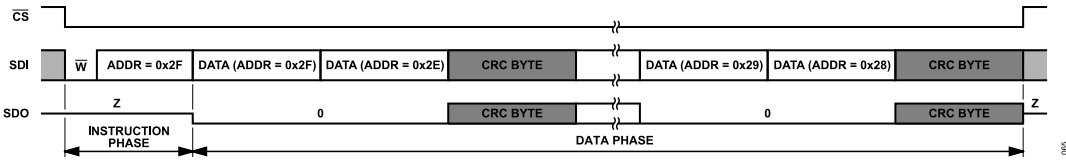


Figure 65. SPI Write Transaction Example in Streaming Mode for Multibyte Registers with CRC Enabled (Address Descending)

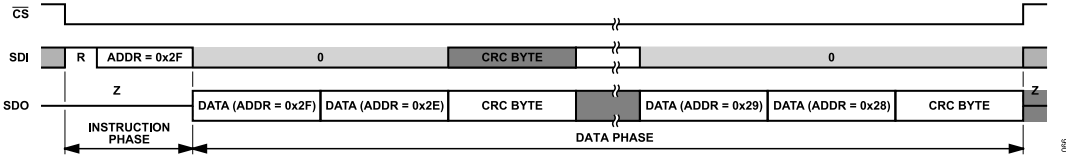


Figure 66. SPI Read Transaction Example in Streaming Mode for Multibyte Registers with CRC Enabled (Address Descending)

## DIGITAL INTERFACE

### OVERTEMPERATURE WARNING ALERT

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

Mask bits in the TEMP\_WARN\_INT\_EN register can be used to mask the corresponding TEMP\_WARN\_FLAG\_CH\_x bits in the TEMP\_WARN\_STAT register. By default, the TEMP\_WARN\_FLAG\_CHx bits are disabled.

These status bits can also be routed out during the instruction phase of any SPI transaction through the SEND\_STATUS bit of the INTERFACE\_CONFIG\_C register. For more details, refer to the [Device Status During Instruction Phase](#) section.

There is no autoshtutdown functionality based on the over-temperature warning alert. It is recommended to read out the TEMP\_WARN\_STAT register to respond to the over-temperature warning alert.

### DEVICE RESET

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

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

A hardware reset is initiated by a falling edge on the  $\overline{\text{RESET}}$  pin and the device remains in a reset state for as long as the  $\overline{\text{RESET}}$  pin is held low. [Figure 4](#) shows the hardware reset timing diagram, with the minimum RESET pulse width, represented by  $t_{\text{RESET\_PW}}$  in [Figure 4](#) and [Table 3](#).

A software reset is initiated by setting both the SW\_RESET and RESET\_SW bits in the INTERFACE\_CONFIG\_A register to 1 in the same data phase. [Figure 5](#) shows the software reset timing diagram.

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

The device requires a minimum of 200µs between any reset event and a register read/write transaction as shown in [Figure 4](#) and [Figure 5](#) and represented by  $t_{\text{RESET\_DELAY}}$  in [Table 3](#). If an SPI

transaction is attempted before the device is ready, it may not succeed and the NOT\_READY\_ERR bit in the INTERFACE\_STATUS\_A register is set. This is also a R/W1C type.

To verify complete initialization, check the NOT\_READY\_ERR bit in the INTERFACE\_STATUS\_A register and the POWERUP\_COMPLETED bit in the DIGITAL\_STATUS register. If any error bit is flagged, perform a device reset.

### SCRATCHPAD AND USER SPARE REGISTERS

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

### DEVICE IDENTIFICATION

The following registers contain identification information about the AD5706R:

- ▶ VENDOR register to identify Analog Devices as the vendor of the device
- ▶ CHIP\_TYPE register to identify the category of Analog Devices products the device belongs to
- ▶ PRODUCT\_ID register to be used in conjunction with CHIP\_TYPE to identify a device
- ▶ CHIP\_GRADE register to record the device revision and performance grade
- ▶ SPI\_REVISION register for the information on the SPI interface revision

The device identifies as follows:

- ▶ VENDOR\_ID = 0x0456
- ▶ CHIP\_TYPE = 0x08
- ▶ PRODUCT\_ID = 0x4130
- ▶ CHIP\_GRADE = 0x0
- ▶ SPI\_REVISION = 0x85

## DAC OPERATIONS

The DAC core of each channel of the AD5706R has double buffered registers for the DAC code. Data for each channel can be written directly to the DAC register for an asynchronous DAC update, or to the input registers (DAC\_INPUT\_A\_CHn or FUNC\_DAC\_INPUT\_B\_CHn) without changing the DAC outputs. A load DAC command issued to the device via the LDAC pin or the DAC\_SW\_LDAC bit transfers input register content into the DAC register, updating the DAC output.

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

## DAC REGISTERS (DATA AND INPUT)

Each DAC channel has an associated 16-bit DAC\_DATA\_READBACK\_CHn, DAC\_INPUT\_A\_CHn, and FUNC\_DAC\_INPUT\_B\_CHn registers.

- ▶ The DAC\_DATA\_READBACK\_CHn is a read-only register that contains the undecoded DAC code that is loaded to the DAC core.
- ▶ The DAC\_INPUT\_A\_CHn register is used for storing DAC data that can be loaded to the DAC\_DATA\_READBACK\_CHn register at a later time through the load DAC functions. It is also utilized for the toggle and dither tone generation functionalities.
- ▶ Lastly, the FUNC\_DAC\_INPUT\_B\_CHn register is used for storing DAC data that is used for the toggle DAC and dither tone generation functionalities.

## IOUT OUTPUT STAGES

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

The full-scale current range is selected via the bits OUT\_RANGE\_CH, on a per channel basis. At power-up, the device initializes with all channel outputs (IOUT0 to IOUT3) in off mode (high-Z mode). The range and code of each channel are then fully programmable through SPI write to registers OUT\_RANGE\_CHn (Address 0x28 to Address 0x2F) as shown in Table 15 and Figure 67.

Table 15. Output Ranges

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

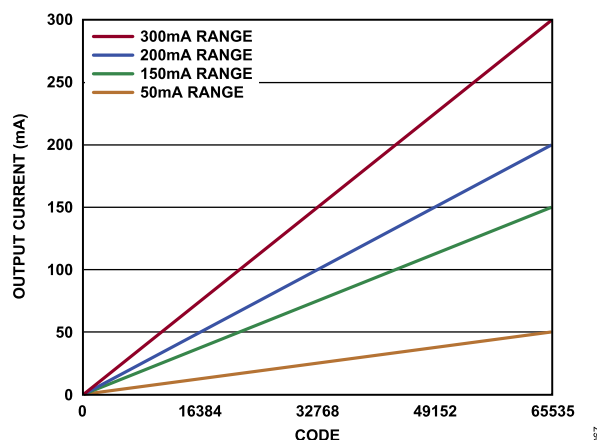


Figure 67. 16-bit IDAC Transfer Function

## IOUT in High-Z Mode

On power-up or on device reset, the outputs of each channel are in high-Z (off) mode (by default) and the DAC core is active. Additionally, there is an internal ground switch for each channel, where the output is shorted to GND through a typical 10Ω switch internally. Before the DAC ground switch is closed, the DAC output for the associated channel is set to high-Z to avoid GND switch burnout (break-before-make topology). However, the bias circuitry for each IDAC channel remains powered up, and only the output is either in off state or is connected to GND.

The user can control the DAC output stage through the SPI Write or through OUT\_EN pin. When changing between output modes on a DAC channel, the output stage of the channel is set to high-Z to prevent glitches on the output. The output enable functionality for each channel is controlled by specific register bits: OUT\_OP\_MODE\_CHn, OUT\_SWITCH\_EN\_CHn, and HW\_SHUTDOWN\_EN\_CHn. These bits configure the output stage to either high-Z state or connect to ground. In specific use cases, the host controller may read the alarm and other fault status of the AD5706R or other devices on the system and accordingly decides upon status of the DAC output stages through the control of OUT\_EN pin or through SPI write. Alternatively, the user can tie the OUT\_EN pin to a resistor divider junction between the AVDD and GND, and any fault on the AVDD can disable the DAC output. There are certain bit field names for the output states per channel, as mentioned in the Table 16.

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

## DAC OPERATIONS

**Table 16. Output in High-Z Mode and Ground Switch Control**

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

### IOU Voltage Compliance

When sourcing current, each IOUTx pin must comply with the voltage headroom required by its respective positive supply PVDDx. The minimum headroom voltage when sourcing at full scale is listed in [Table 1](#) as output compliance voltage.

### DAC UPDATE MODES

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

[Table 23](#) summarizes the DAC update modes described in this section and succeeding subsections. All DAC update modes described assume the respective output stages are enabled.

### DAC Functionality Control

The Bitfield FUNC\_EN\_CHn and Bitfield FUNC\_MODE\_CHn of the FUNC\_EN and FUNC\_MODE\_SEL registers, respectively, are used to control which DAC update features are active at a time for each DAC channel.

When FUNC\_EN\_CH is 1'b0, only the load DAC function is enabled. The user can select a DAC function through the FUNC\_MODE\_CH bitfield. However, only by writing 1'b1 to the bitfield FUNC\_EN\_CH enables the selected DAC function.

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

The functions, namely, load DAC, toggle DAC, and dither tone generation, are logically exclusive with each other. This means that when one function is active for a particular channel, the other two are not. These functions can be executed either using LDAC pin or with an SPI write command. All DAC functions assume that the output stages of respective DAC channels are enabled. See the definition of FUNC\_EN\_CH and FUNC\_MODE\_CH bits in [Table 17](#) below.

**Table 17. DAC Functionality Control bits**

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

### Global Software and Hardware Triggers

All channels share the same triggers for each of their respective software and hardware modes. For instance, SW LDAC, SW toggle, and SW dither share the trigger of SPI write of 1'b1 to the

## DAC OPERATIONS

DAC\_SW\_LDAC\_CHn bitfield of the DAC\_SW\_LDAC register. This means that the channels are independent of each other and trigger their corresponding SW modes functionalities with just one SPI write. This is the same for the HW triggers: LDAC, TGP, and DCK. This is true in AD5706R since the three mentioned signals share the same pin.

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

### LOAD DAC FUNCTION

DAC channels can be updated with the load DAC function. The load DAC functionality can be divided into three modes: asynchronous load, hardware LDAC (through LDAC pin), and software LDAC (through SPI write to the DAC\_SW\_LDAC register) as illustrated in Figure 68. For the AD5706R, the LDAC pin is shared with the TGP (toggle) and DCK (dither clock) pins.

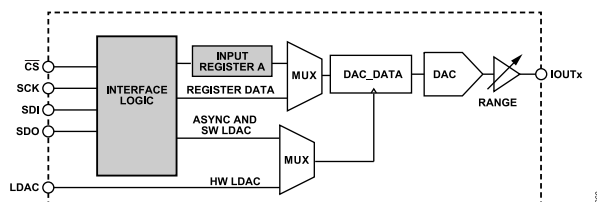


Figure 68. Load DAC Function

Table 18. Load DAC Function Configuration

FUNC_EN_CH n bit	FUNC_MODE _CHn bit	LDAC_SYNC		DAC Function
		ASYNC_CH n bit	LDAC_HW_SW _CHn bit	
1'b1	x	x	x	None are active
1'b0 (default)	x	0 (default)	x	Only asynchronous load is active
1'b0 (default)	x	1	0 (default)	Only HW LDAC is active
1'b0 (default)	x	1	1	Only SW LDAC is active

The load DAC function is enabled by default or by writing 1'b0 to the Bitfield FUNC\_EN\_CH assigned to each channel on their corresponding FUNC\_EN register. During load DAC mode, the user can choose which mode is active through the Bitfield LDAC\_SYNC\_ASYNC\_CHn and Bitfield LDAC\_HW\_SW\_CHn on the LDAC\_SYNC\_ASYNC and LDAC\_HW\_SW registers, respectively. See Table 18 for reference.

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

### Asynchronous Load DAC

During asynchronous load DAC update, writing to a DAC\_INPUT\_A\_CHn register immediately sends the written data to the corresponding DAC\_DATA\_READBACK\_CHn register of that channel and triggers a DAC update. The DAC output is updated after the last SCK rising edge (16th rising edge of SCK in the data phase) as shown in Figure 8. It is to be noted that writing to the FUNC\_DAC\_INPUT\_B\_CHn register has no effect on the DAC\_DATA\_READBACK\_CHn register or the DAC output of any respective channel in the asynchronous load DAC mode.

### Software LDAC (SPI Write)

The software LDAC (SW LDAC) mode utilizes a SPI write of 1'b1 to the Bitfield DAC\_SW\_LDAC\_CHn on the DAC\_SW\_LDAC register as trigger to the load DAC operations. This can be enabled per channel by writing 1'b1 to the LDAC\_SYNC\_ASYNC\_CHn and 1'b1 to the LDAC\_HW\_SW\_CHn bitfields of the corresponding LDAC\_SYNC\_ASYNC and LDAC\_HW\_SW registers of each channel, respectively.

Writing 1'b1 to the DAC\_SW\_LDAC\_CHn bitfield of a channel in the DAC\_SW\_LDAC register transfers the data that is currently written to its respective DAC\_INPUT\_A\_CHn register, given that it is in SW LDAC mode, to their corresponding DAC\_DATA\_READBACK\_CHn register. This triggers a DAC update. See Figure 9.

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

The Bitfield DAC\_SW\_LDAC\_CHn auto clears upon write of 1'b1 to the register. Thus, reading back from the DAC\_SW\_LDAC register only returns 0s.

### Hardware LDAC (LDAC Pin)

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

- Idle high/low: While the LDAC pin is held high or low, the data can be written to the DAC\_INPUT\_A\_CHn registers without triggering a DAC update.
- Active edge: The active edge for HW LDAC is selected through the LDAC\_HW\_EDGE\_SEL\_CHn bitfields on the LDAC\_EDGE\_SEL register. This is also true for other HW triggers (toggle, dither). See Table 19 for reference.

## DAC OPERATIONS

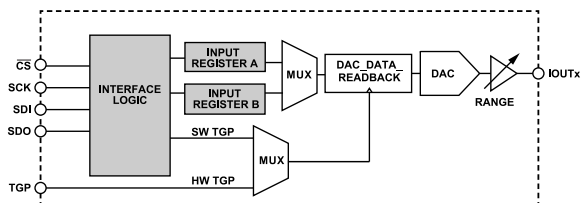
**Table 19. HW Active Edge Configuration**

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

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

### TOGGLE DAC FUNCTION

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



**Figure 69. Toggle DAC Function**

**Table 20. Toggle DAC Function Configuration**

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

The toggle DAC function is enabled by writing 1'b0 to the Bitfield FUNC\_MODE\_CHn on the FUNC\_MODE\_SEL register, and then, writing 1'b1 to the FUNC\_EN\_CHn bitfield in the FUNC\_EN register. During toggle DAC mode, the user can choose which trigger is active (HW/SW) by writing to the corresponding LDAC\_HW\_SW register. See [Table 20](#) for reference.

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

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

### Software Toggle (SPI Write)

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

#### First Write

After enabling SW toggle mode, writing 1'b1 to a Bitfield DAC\_SW\_LDAC\_CHn of the DAC\_SW\_LDAC register transfers the data that is currently written to its corresponding DAC\_INPUT\_A\_CHn register, given that it is in SW toggle mode, to its corresponding DAC\_DATA\_READBACK\_CHn registers. This triggers a DAC update. See [Figure 11](#).

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

The Bitfield DAC\_SW\_LDAC\_CHn auto clears upon write of 1'b1 to the register. Thus, reading back from the DAC\_SW\_LDAC register only returns 0s.

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

### Succeeding Writes

Succeeding writes of 1'b1 to the Bitfield DAC\_SW\_LDAC\_CHn transfer alternately between the DAC\_INPUT\_A\_CHn and FUNC\_DAC\_INPUT\_B\_CHn registers (starting with FUNC\_DAC\_INPUT\_B\_CHn for the second write). For instance, the second write transfers data from FUNC\_DAC\_INPUT\_B\_CHn to the corresponding DAC\_DATA\_READBACK\_CHn register, from DAC\_INPUT\_A\_CHn to the third write, and then from FUNC\_DAC\_INPUT\_B\_CHn again to the next.

### Hardware Toggle (TGP Pin)

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

Same as the HW LDAC, the active edge for HW toggle is selected through the LDAC\_HW\_EDGE\_SEL\_CHn bitfields on the LDAC\_EDGE\_SEL\_CHn register. See [Table 19](#) again as reference.

The HW toggle functionality is detailed in the following sections.

## DAC OPERATIONS

### Idle High/Low

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

### First Edge

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

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

### Succeeding Edges

Succeeding active edges transfers alternately between the DAC\_INPUT\_A\_CHn and FUNC\_DAC\_INPUT\_B\_CHn registers (starting with FUNC\_DAC\_INPUT\_B\_CHn for the second write). For example, the second write transfers data from FUNC\_DAC\_INPUT\_B\_CHn to the corresponding DAC\_DATA\_READBACK\_CHn register, from DAC\_INPUT\_A\_CHn to the third write, and then from FUNC\_DAC\_INPUT\_B\_CHn again to the next. See [Figure 70](#).

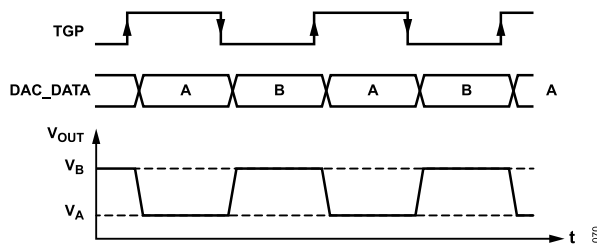


Figure 70. Toggle Timing with Both Edges Selected as Active Trigger

### Recommended Flow for Toggle DAC Function

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

1. Select the toggle DAC function for the desired channels by writing 1'b0 (default) to their respective FUNC\_MODE\_CHn bitfields on the FUNC\_MODE\_SEL register. If FUNC\_EN\_CHn bit for the corresponding channel is 1'b1, then it must be reset to 1'b0 before enabling the FUNC\_MODE\_CHn for the toggle DAC function.
2. Enable the toggle DAC function for the desired channels by writing 1'b1 to the corresponding FUNC\_EN\_CHn bitfields on the FUNC\_EN register.
3. Select whether the trigger to be used by the toggle DAC function is via the TGP pin (HW toggle) or via SPI write (SW

toggle) by writing to the LDAC\_HW\_SW\_CHn bitfields in the LDAC\_HW\_SW register. Write 1'b0 for HW toggle or 1'b1 for SW toggle.

4. For HW toggle mode, select which edges are going to be used as active edges for the TGP pin by writing to the corresponding LDAC\_HW\_EDGE\_SEL\_CHn bitfields on the LDAC\_EDGE\_SEL register.
5. Write to the target DAC\_INPUT\_A\_CHn and FUNC\_DAC\_INPUT\_B\_CHn registers.
6. For HW toggle mode, apply an edge transition to the TGP pin for the set active edges. For SW toggle mode, write to the corresponding DAC\_SW\_LDAC\_CHn bitfield on the DAC\_SW\_LDAC register.
7. Repeat Step 6 to functionally output a square wave.
8. Disable the toggle DAC function for the desired channels by writing 1'b0 to the corresponding FUNC\_EN\_CHn bitfields on the FUNC\_EN register.

### DITHER FUNCTION

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

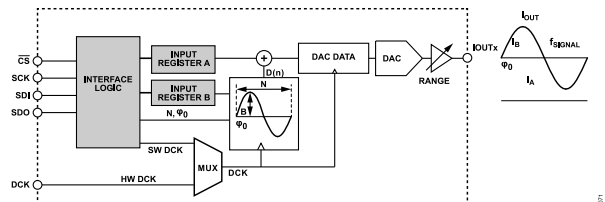


Figure 71. Dither Tone Function

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

$$D(n) = B \times \sin\left(\frac{2\pi n}{N} + \phi_0\right) \quad (8)$$

where:

$n = \{0, 1, 2, \dots, (N-1)\}$ .

$N$  = dither period = {4, 8, 16, 32, 64} samples. Refer to register FUNC\_DITHER\_PERIOD\_CHn in [Table 21](#).

$\phi_0$  = initial phase offset = {0°, 90°, 180°, 270°}. Refer to register FUNC\_DITHER\_PHASE\_CHn in [Table 21](#).

$B$  = dither amplitude =  $[0, 2^{(M-2)} - 1]$ . It is configured through the FUNC\_DAC\_INPUT\_B\_CHn register. Refer to [Table 11](#). The 2 MSB bits of data are disregarded on the register FUNC\_DAC\_INPUT\_B\_CHn while configuring the dither amplitude.

$M$  = DAC resolution = 16.

$D(n)$  = output range =  $[-B, B]$ .

The signal parameters,  $N$  and  $\phi_0$ , can also be selected for each individual channel through the FUNC\_DITHER\_PERIOD\_CHn and

## DAC OPERATIONS

FUNC\_DITHER\_PHASE\_CHn registers, respectively. This allows the precise control of signal frequencies and phase relationships between dithered DAC channels.

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

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

Each channel also utilizes the DAC\_INPUT\_A\_CHn and FUNC\_DAC\_INPUT\_B\_CHn registers, where the DAC\_INPUT\_A\_CHn register is used for the offset of the sine wave, while the FUNC\_DAC\_INPUT\_B\_CHn register is used for scaling the amplitude of the sine wave.

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

where:

A is an M-bit unsigned data obtained from register DAC\_INPUT\_A\_CHn with range  $[0, 2^M - 1]$ .

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

M is DAC resolution of value 16.

At phases  $90^\circ$  and  $270^\circ$ , the DAC\_OUTPUT\_CODE is same as DAC\_INPUT\_A\_CHn for the first (N/4) samples. It is to be noted that the DAC\_OUTPUT\_CODE range is bounded to  $[0, 2^M - 1]$ , where M is the DAC resolution.

**Table 21. Dither Tone Generation Module Registers**

Register	Bitfield	Reset Value	Description
FUNC_DITHER_PHASE_CHn	DITHER_PHASE_CHn	2'b00	Configures the initial phase offset ( $\phi_0$ ) in terms of degrees of the sine wave: <ul style="list-style-type: none"> <li>▶ 2'b00 : <math>0^\circ</math> (default)</li> <li>▶ 2'b01 : <math>90^\circ</math></li> <li>▶ 2'b10 : <math>180^\circ</math></li> <li>▶ 2'b11 : <math>270^\circ</math></li> </ul>
FUNC_DITHER_PERIOD_CHn	DITHER_PERIOD_CHn	3'b000	Configures the period (N) in terms of samples of the sine wave: <ul style="list-style-type: none"> <li>▶ 3'b000 : 4 samples (default)</li> <li>▶ 3'b001 : 8 samples</li> <li>▶ 3'b010 : 16 samples</li> <li>▶ 3'b011 : 32 samples</li> <li>▶ 3'b100 to 3'b111 : 64 samples</li> </ul>

## DAC OPERATIONS

### Example of Dither Operation

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

For the AD5706R, the DC value to be written to DAC\_INPUT\_A\_CH0 is d21845 = 0x5555.

The allowable range for dither amplitude is 1/4th the output current range. FUNC\_DAC\_INPUT\_B\_CH0 is loaded with

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

For  $N = 64$ , set the DCK frequency to  $64 \times 10kHz = 640kHz$ . Set the initial phase value to 0 in the register DITHER\_PHASE\_CH0. See [Table 21](#).

### Dither Tone Generation

The dither mode can be enabled per channel by writing 1'b1 to the respective FUNC\_MODE\_CHn bitfield in the FUNC\_MODE\_SEL register, and then, writing 1'b1 to the FUNC\_EN\_CHn bitfield in the FUNC\_EN register.

The dither clock (DCK), that is responsible for sending the dither code output DAC\_OUTPUT\_CODE to the DAC\_DATA\_READBACK\_CHn register, is controlled/generated via the DCK pin (HW dither) or via SPI writes (SW dither). This option is selected through the LDAC\_HW\_SW\_CHn bitfield on the LDAC\_HW\_SW register. See the [Table 22](#) for reference. The multi-DAC function is also active during dither mode.

**Table 22. Dither Tone Generation Function Configuration**

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

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

### First Trigger

The DAC\_OUTPUT\_CODE value of the DAC update of the first SW/HW trigger event after entering dither mode is always DAC\_INPUT\_A\_CHn register. This is because  $A + \sin(0)$  is  $A$  (in reference to [Equation 10](#)).

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

### Software Dither (SPI Write)

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

Writing 1'b1 to the bitfield DAC\_SW\_LDAC\_CHn of the DAC\_SW\_LDAC register transfers the data that is currently written to a DAC\_INPUT\_A\_CHn register to their corresponding DAC\_DATA\_READBACK\_CHn register. This triggers a DAC update. See [Figure 11](#). It is to be noted that the channels that are on other SW modes (SW LDAC and SW toggle) also trigger their corresponding SW functions when writing 1'b1 to the bitfield DAC\_SW\_LDAC\_CHn of the DAC\_SW\_LDAC register.

The bitfield DAC\_SW\_LDAC\_CHn auto clears upon write of 1'b1 to the register. Thus, reading back from the DAC\_SW\_LDAC register only returns 0s.

### Hardware Dither (DCK Pin)

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

### Idle High/Low

While the DCK pin is held high or low, no change in the value of the DAC\_OUTPUT\_CODE output happens.

### Active Edge

Same as the HW LDAC and HW toggle, the active edge for HW dither is selected through the LDAC\_HW\_EDGE\_SEL\_CHn bitfields on the LDAC\_EDGE\_SEL register. See [Table 19](#) again as reference.

An active edge on the DCK pin updates the DAC\_DATA\_READBACK\_CHn register (with their respective DCK active edge) with the calculated data based on the DAC\_INPUT\_A\_CHn and FUNC\_DAC\_INPUT\_B\_CHn registers. It is to be noted that the channels that are on other HW modes (HW LDAC and HW toggle) also trigger their corresponding HW functions when triggering on their set active edges.

### Disabling the Dither Operation

Dither operation can be disabled by resetting the FUNC\_EN\_CHn bit to 0. After this bit is reset, the dither operation is effectively disabled and regular DAC codes can then be written to the DAC channel.

## DAC OPERATIONS

### Recommended Flow for Dither Tone Generation Function

The following is the recommended flow for the dither generation:

1. Select the dither tone generation function for the desired channels by writing 1'b1 to their respective FUNC\_MODE\_CHn bitfields on the FUNC\_MODE\_SEL register. If FUNC\_EN\_CHn bit for the corresponding channel is 1'b1, then it must be reset to 1'b0 before enabling the FUNC\_MODE\_CHn for the dither tone generation function.
2. Enable the dither tone generation function for the desired channels by writing 1'b1 to the corresponding FUNC\_EN\_CHn bitfields on the FUNC\_EN register.
3. Select whether the trigger to be used by the dither tone generation function is via the DCK pin (HW dither) or via SPI writes (SW dither) by writing to the LDAC\_HW\_SW\_CHn bitfields in the LDAC\_HW\_SW register. Write 1'b0 for HW dither or 1'b1 for SW dither.
4. For HW DCK, select which edges are going to be used as active edges for the DCK pin by writing to the corresponding LDAC\_HW\_EDGE\_SEL\_CHn bitfields on the LDAC\_EDGE\_SEL register.
5. Set the initial phase offset and period in the DITHER\_PERIOD and DITHER\_PHASE bits on the FUNC\_DITHER\_PERIOD\_CHn and FUNC\_DITHER\_PHASE\_CHn registers, respectively.
6. Write to the target DAC\_INPUT\_A\_CHn registers to set the vertical offset of the sine wave.
7. Write to the target FUNC\_DAC\_INPUT\_B\_CHn registers to set the amplitude of the sine wave.
8. For HW dither, apply an edge transition to the DCK pin for the set active edge; or, for SW dither, write 1'b1 to the corresponding DAC\_SW\_LDAC\_CHn bitfield on the DAC\_SW\_LDAC register.
9. Repeat Step 8 to functionally output a sine wave.
10. Disable the dither generation for the desired channels by writing 1'b0 through their corresponding FUNC\_EN\_CHn bitfields on the FUNC\_EN register.

## DAC OPERATIONS

## Normalized Waveforms

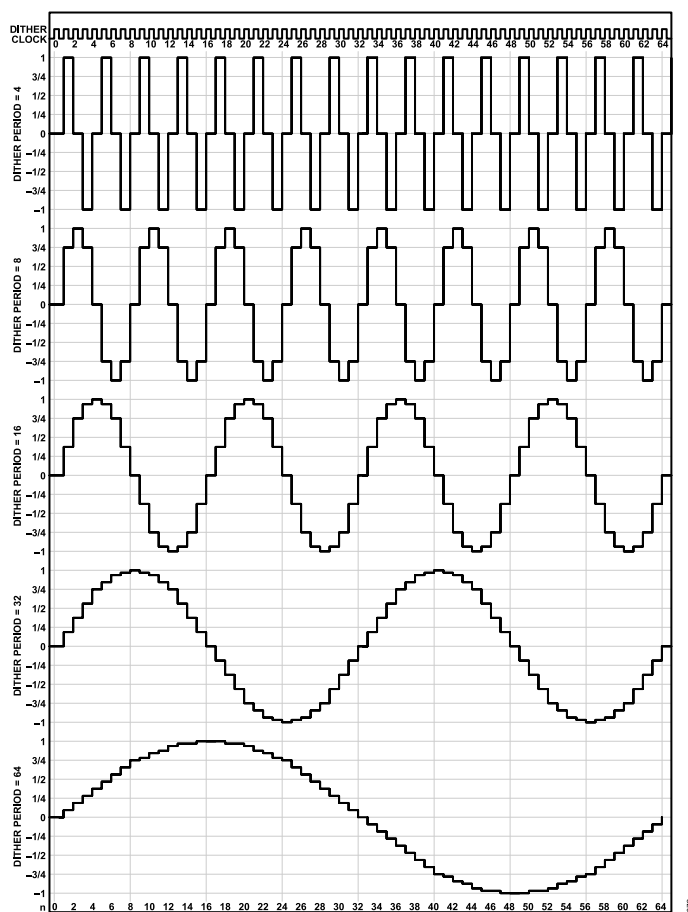
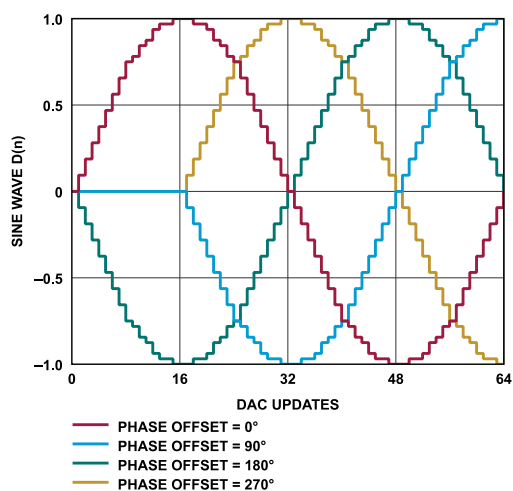
Figure 72. Normalized Sinusoidal Waveforms for Various Values of Dither Period  $N$ 

Figure 73. Normalized Sinusoidal Waveforms for Various Values of Dither Phase Offset

## DAC OPERATIONS

### MULTI-DAC FUNCTION

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

Data written to the MULTI\_DAC\_INPUT\_A register is also written to the DAC\_INPUT\_A\_CHn registers whose MULTI\_DAC\_CH\_SEL bits are asserted. For example, when bits 0 and 1 in the MULTI\_DAC\_CH\_SEL register are set to 1'b1 (channel 0 and channel 1 are selected), and then the data is written to MULTI\_DAC\_INPUT\_A register, the same data is written to the DAC\_INPUT\_A\_CH0 and DAC\_INPUT\_A\_CH1 registers concurrently.

The function of transferring written data from the MULTI\_DAC\_INPUT\_A register to the selected DAC\_INPUT\_A\_CHn registers is active for all modes.

### Asynchronous Load DAC Mode

Writing to the MULTI\_DAC\_INPUT\_A register transfers the written data to the corresponding DAC\_INPUT\_A\_CHn registers of channels selected in the MULTI\_DAC\_CH\_SEL register. Additionally, the selected channels that are in asynchronous load DAC mode also transfers the written data to their respective DAC\_DATA\_READBACK\_CHn registers, and this triggers a DAC update.

The functionality and timing is same as referenced in the [Asynchronous Load DAC](#) section.

### Software LDAC Mode

Writing 1'b1 to the Bitfield MULTI\_DAC\_SW\_LDAC of the MULTI\_DAC\_SW\_LDAC register triggers a SW LDAC for all channels that are in SW LDAC mode and are selected in the MULTI\_DAC\_CH\_SEL register. This transfers the contents of the MULTI\_DAC\_INPUT\_A register to the respective DAC\_INPUT\_A\_CHn registers and their corresponding DAC\_DATA\_READBACK\_CHn registers and triggers a DAC update for each selected channel.

The sequence is as follows:

1. For all desired DAC channels to be triggered with a SW LDAC update, set the corresponding DAC function mode to any SW modes (SW LDAC, SW toggle, and SW dither).
2. Write 1'b1 to all the bits assigned to the desired channels in the MULTI\_DAC\_CH\_SEL register.
3. Set the value of the DAC\_INPUT\_A\_CHn registers of selected channels by writing to the MULTI\_DAC\_INPUT\_A register. Note that if there are channels that are in asynchronous load DAC mode and are also selected in the MULTI\_DAC\_CH\_SEL register, this triggers a DAC update.

4. Write 1'b1 to the bitfield MULTI\_DAC\_SW\_LDAC of the MULTI\_DAC\_SW\_LDAC to trigger a corresponding SW LDAC to the selected channels depending on their SW modes as follows:
  - a. SW LDAC: transfer the content of the DAC\_INPUT\_A\_CHn registers to their corresponding DAC\_DATA\_READBACK\_CHn registers and triggers a DAC update.
  - b. SW toggle: toggle between transferring data from DAC\_INPUT\_A\_CHn and FUNC\_DAC\_INPUT\_B\_CHn registers to the corresponding DAC\_DATA\_READBACK\_CHn register and triggers a DAC update.
  - c. SW dither: output a step of a sine wave and trigger a DAC update.
  - d. Other modes: no updates to the DAC\_DATA\_READBACK\_CHn registers and the DAC outputs.

The Bitfield MULTI\_DAC\_SW\_LDAC\_CHn auto clears upon write of 1'b1 to the register. Thus, reading back from the MULTI\_DAC\_SW\_LDAC register only returns 0s.

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

### Hardware LDAC Mode

When the HW LDAC pin transitions to an active edge (selected in the LDAC\_EDGE\_SEL\_CHn register), a HW LDAC is triggered for all the channels that are in HW LDAC mode and are selected in the MULTI\_DAC\_CH\_SEL register. This transfers the contents of the MULTI\_DAC\_INPUT\_A register to the respective DAC\_INPUT\_A\_CHn registers and their corresponding DAC\_DATA\_READBACK\_CHn registers and triggers a DAC update for each selected channel.

The sequence is as follows:

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

## DAC OPERATIONS

- a. HW LDAC: transfer the content of the DAC\_INPUT\_A\_CHn register to their corresponding DAC\_DATA\_READBACK\_CHn register and trigger a DAC update.
- b. HW toggle: toggle between transferring data from DAC\_INPUT\_A\_CHn and FUNC\_DAC\_INPUT\_B\_CHn registers to the corresponding DAC\_DATA\_READBACK\_CHn register and trigger a DAC update.
- c. HW DCK: output a step of a sine wave and trigger a DAC update.
- d. Other modes: no updates to the DAC\_DATA\_READBACK\_CHn registers and the DAC outputs.

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

## SUMMARY OF DAC MODES

Table 23. DAC Modes in Terms of DAC Configuration Bits

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

## APPLICATIONS INFORMATION

## RECOMMENDED EXTERNAL COMPONENTS

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

Typically, capacitor values are considered with a  $\pm 5\%$  tolerance unless otherwise specified.

Table 24. External Passive Components

Pin Name	Passive Components Required
RESET	10k $\Omega$ to IOVDD
NF	0.1 $\mu$ F capacitor to AGND
VREF	1 $\mu$ F capacitor to AGND
AVDD	1 $\mu$ F capacitor per pin to AGND
PVDDx	1 $\mu$ F capacitor per pin to AGND
IOVDD	1 $\mu$ F capacitor to AGND

## POWER SUPPLY INITIALIZATION

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

The sequence described in the [Device Reset](#) section requires a specific duration identified as  $t_{\text{RESET\_DELAY}}$  in [Table 3](#). During  $t_{\text{RESET\_DELAY}}$ , avoid using the digital interface to communicate with the device, including toggling any digital asynchronous pins.

If the digital host attempts an SPI transaction before the device is ready, the transaction is invalid and the NOT\_READY\_ERR bit in the INTERFACE\_STATUS\_A register is set. The NOT\_READY\_ERR bit is a R/W1C type of bit. Check the NOT\_READY\_ERR bit in the INTERFACE\_STATUS\_A register and the POWERUP\_COMPLETED bit in the DIGITAL\_STATUS register to verify initialization. If any error bit is flagged, perform a device reset.

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

## COMBINING CHANNELS TO INCREASE CURRENT RANGE

The maximum current that can be sourced from IDACx is 300mA. It is possible to increase the current source capability by directly connecting the two or more channels together. For example combining IOUT0 with IOUT1 can create a full scale output current of 400mA as shown in [Figure 74](#). When combining channels, ensure the following:

- ▶ The output compliance voltage stays within the range specified in [Table 1](#).
- ▶ The output voltage stays within the absolute maximum ratings specified in [Table 5](#).

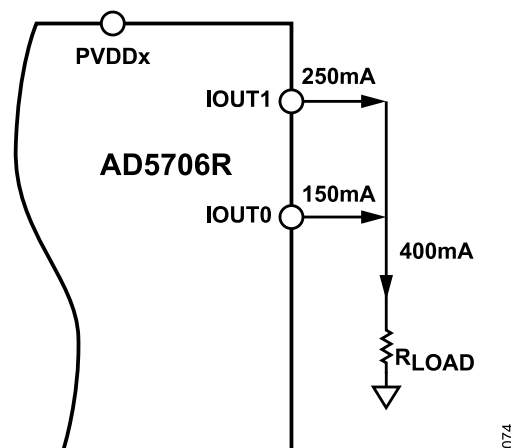


Figure 74. Increasing the Current Range by Summing Channels

## COMBINING CHANNELS TO REDUCE OUTPUT NOISE

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

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

$$NSD_{IOUT} \approx \sqrt{NSD_{IOUT0}^2 + NSD_{IOUT1}^2} \quad (12)$$

$$= \sqrt{2.25^2 + 2.25^2} \approx 3.18\text{nA}/\sqrt{\text{Hz}}$$

Thus, the NSD is improved from 4.49nA/ $\sqrt{\text{Hz}}$  to an estimation of 3.18nA/ $\sqrt{\text{Hz}}$  by combining both channels.

## MICROPROCESSOR INTERFACING

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

The SPI interface of the AD5706R is designed for easy connection to industry-standard DSPs and microcontrollers. [Figure 75](#) shows the AD5706R connected to the ADuCM320. The ADuCM320 has an integrated SPI port that can be connected directly to the SPI pins of the AD5706R.

## APPLICATIONS INFORMATION

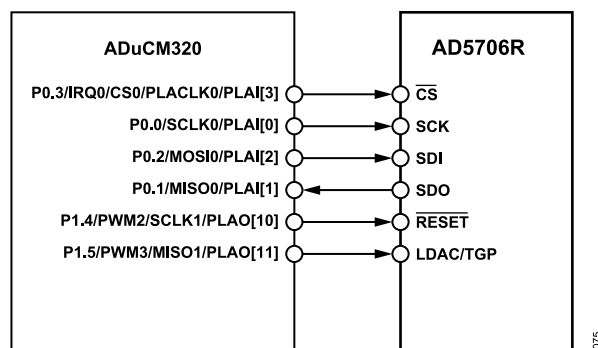


Figure 75. ADuCM320 SPI Interface

## LAYOUT AND ASSEMBLY GUIDELINES

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

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

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

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

Table 25. Thermal Considerations for AD5706R WLCSP Package

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

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  $>0.8\text{mm}$ , 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  $150^\circ\text{C}$  (see [Table 5](#)) and a maximum operating junction temperature ( $T_{J\_MAX}$ ) of  $125^\circ\text{C}$  (see [Table 1](#)). To meet the specified performance, the AD5706R must be operated at a junction temperature no greater than  $125^\circ\text{C}$ . The junction temperature is directly affected by the power dissipated across the AD5706R and the ambient temperature.

Most of the power dissipation of the AD5706R comes from the IOUT channels. [Table 1](#) specifies the output current ranges for each IOUT channel and the maximum power supply voltages. It is important to understand the effects of power dissipation on the package and the effects the package has on the junction temperature. The part is packaged in a 36-ball  $2.58\text{mm} \times 2.58\text{mm}$  WLCSP. The thermal resistance values are specified in [Table 6](#). [Table 25](#) 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 25. Thermal Considerations for AD5706R WLCSP Package (Continued)

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

## REGISTERS

The AD5706R has programmable user configuration registers that are used to configure the device. The [Register Summary](#) section contains the complete list of the user configuration registers.

The access column specifies whether the register comprises only of read only bits (R) or a mix of read only and read/write bits (R/W). Read only bits cannot be overwritten by an SPI write transaction, whereas read/write bits can. Attempting to write to a register that contains only read only bits sets the WR\_TO\_RD\_ONLY\_REG\_ERR flag in the INTERFACE\_STATUS\_A register. At-

tempting to access an invalid register address sets the ADDR\_INVALID\_ERR flag in the INTERFACE\_STATUS\_A register. The error flags in the INTERFACE\_STATUS\_A register are also read or write 1 to clear (R/W1C) and are only reset when an SPI write transaction writes a 1 to their location.

The [Register Summary](#) section and the [Register Bitwise Summary](#) section show the size of the register and its bitfields. See the [Multi-byte Registers](#) section for a detailed description of how multibyte registers can be accessed.

## REGISTER SUMMARY

Table 26. Register Summary

Address	Name	Description	Reset	Access
0x00	<a href="#">INTERFACE_CONFIG_A</a>	Interface Configuration A.	0x10	R/W
0x01	<a href="#">INTERFACE_CONFIG_B</a>	Interface Configuration B.	0x00	R/W
0x02	<a href="#">DEVICE_CONFIG</a>	Device Configuration.	0x00	R
0x03	<a href="#">CHIP_TYPE</a>	Chip Type.	0x08	R
0x04	<a href="#">PRODUCT_ID_L</a>	Product ID Low.	0x30	R
0x05	<a href="#">PRODUCT_ID_H</a>	Product ID High.	0x41	R
0x06	<a href="#">CHIP_GRADE</a>	Chip Grade.	0x00	R
0x0A	<a href="#">SCRATCH_PAD</a>	Scratch Pad.	0x00	R/W
0x0B	<a href="#">SPI_REVISION</a>	SPI Revision.	0x85	R
0x0C	<a href="#">VENDOR_L</a>	Vendor ID Low.	0x56	R
0x0D	<a href="#">VENDOR_H</a>	Vendor ID High.	0x04	R
0x0E	<a href="#">STREAM_MODE</a>	Stream Mode.	0x00	R/W
0x0F	<a href="#">TRANSFER_CONFIG</a>	Transfer Configuration.	0x00	R/W
0x10	<a href="#">INTERFACE_CONFIG_C</a>	Interface Configuration C.	0x2F	R/W
0x11	<a href="#">INTERFACE_STATUS_A</a>	Interface Status A.	0x00	R/W
0x14	<a href="#">MULTI_DAC_CH_SEL</a>	Multi-DAC Channel Select.	0x0000	R/W
0x16	<a href="#">LDAC_SYNC_ASYNC</a>	Sync-Async Load.	0x0000	R/W
0x18	<a href="#">LDAC_HW_SW</a>	Hardware-Software Trigger.	0x0000	R/W
0x1A to 0x20 by 2	<a href="#">LDAC_EDGE_SEL_CHn</a>	LDAC Edge Select.	0x0000	R/W
0x22	<a href="#">OUT_OPERATING_MODE</a>	Output Operating Mode.	0x0000	R/W
0x24	<a href="#">OUT_SWITCH_EN</a>	Output Switch Enable.	0x0000	R/W
0x26	<a href="#">OUT_EN</a>	OUT_EN Pin Enable.	0x0000	R/W
0x28 to 0x2E by 2	<a href="#">OUT_RANGE_CHn</a>	Output Range.	0x0000	R/W
0x30	<a href="#">FUNC_EN</a>	Function Enable.	0x0000	R/W
0x32 to 0x38 by 2	<a href="#">FUNC_MODE_SEL_CHn</a>	Function Mode Selector.	0x0000	R/W
0x3A to 0x40 by 2	<a href="#">FUNC_DAC_INPUT_B_CHn</a>	DAC Input B.	0x0000	R/W
0x42 to 0x48 by 2	<a href="#">FUNC_DITHER_PERIOD_CHn</a>	Dither Settings.	0x0000	R/W
0x4A to 0x50 by 2	<a href="#">FUNC_DITHER_PHASE_CHn</a>	Dither Settings.	0x0000	R/W
0x54	<a href="#">MUX_OUT_SEL</a>	MUX Out Select.	0x0000	R/W
0x56	<a href="#">MUX_OUT_CONTROL</a>	MUX Out Control.	0x0001	R/W
0x58	<a href="#">TEMP_WARN_INT_EN</a>	Temperature Warning Interrupt Enabled.	0x0000	R/W
0x5A	<a href="#">MULTI_DAC_SW_LDAC</a>	Software LDAC for Multiple DAC Input.	0x0000	R/W
0x5C	<a href="#">MULTI_DAC_INPUT_A</a>	Multiple DAC Input.	0x0000	R/W

## REGISTERS

Table 26. Register Summary (Continued)

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

## REGISTER BITWISE SUMMARY

Table 27. Register Bitwise Summary

Reg	Name	Bits	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset	RW
0x00	INTERFACE_CONFIG_A	[7:0]	SW_RESET	RESERVED	ADDR_EXTENSION	SDO_ENABLE	RESERVED			RESET_SW	0x10	R/W
0x01	INTERFACE_CONFIG_B	[7:0]	SINGLE_INSTR	RESERVED			SHORT_INSTRUCTION	RESERVED			0x00	R/W
0x02	DEVICE_CONFIG	[7:0]	RESERVED						OPERATING_MODE		0x00	R
0x03	CHIP_TYPE	[7:0]	RESERVED				CHIP_TYPE				0x08	R
0x04	PRODUCT_ID_L	[7:0]	PRODUCT_ID[7:0]								0x30	R
0x05	PRODUCT_ID_H	[7:0]	PRODUCT_ID[15:8]								0x41	R
0x06	CHIP_GRADE	[7:0]	GRADE				DEVICE_REVISION				0x00	R
0x0A	SCRATCH_PAD	[7:0]	SCRATCH_VALUE								0x00	R/W
0x0B	SPI_REVISION	[7:0]	SPI_TYPE		VERSION						0x85	R
0x0C	VENDOR_L	[7:0]	VID[7:0]								0x56	R
0x0D	VENDOR_H	[7:0]	VID[15:8]								0x04	R
0x0E	STREAM_MODE	[7:0]	LOOP_COUNT								0x00	R/W
0x0F	TRANSFER_CONFIG	[7:0]	RESERVED					KEEP_STREAM_LENGTH_VAL	RESERVED		0x00	R/W
0x10	INTERFACE_CONFIG_C	[7:0]	CRC_ENABLE		STRICT_REGISTER_ACCESS	SEND_STATUS	ACTIVE_INTERFACE_MODE		CRC_ENABLEB		0x2F	R/W
0x11	INTERFACE_STATUS_A	[7:0]	NOT_READY_ERR	RESERVED		CLOCK_COUNT_ERR	CRC_ERR	WR_TO_RD_ONLY_REG_ERR	REGISTER_PARTIAL_ACCESS_ERR	ADDRESS_INVALID_ERR	0x00	R/W
0x14	MULTI_DAC_CH_SEL	[7:0]	RESERVED				MULTI_DAC_SEL_CH_3	MULTI_DAC_SEL_CH_2	MULTI_DAC_SEL_CH_1	MULTI_DAC_SEL_CH_0	0x00	R/W
0x15		[15:8]	RESERVED								0x00	
0x16	LDAC_SYNC_ASYNC	[7:0]	RESERVED				LDAC_SYNC_ASYNC_CH_3	LDAC_SYNC_ASYNC_CH_2	LDAC_SYNC_ASYNC_CH_1	LDAC_SYNC_ASYNC_CH_0	0x00	R/W
0x17		[15:8]	RESERVED								0x00	
0x18	LDAC_HW_SW	[7:0]	RESERVED				LDAC_HW_SW_CH_3	LDAC_HW_SW_CH_2	LDAC_HW_SW_CH_1	LDAC_HW_SW_CH_0	0x00	R/W

## REGISTERS

Table 27. Register Bitwise Summary (Continued)

Reg	Name	Bits	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset	RW	
0x19		[15:8]	RESERVED								0x00		
0x1A to 0x20 by 2	LDAC_EDGE_SEL_CHn	[7:0]	RESERVED						LDAC_HW_EDGE_SEL_CH		0x00	R/W	
0x1B to 0x21 by 2		[15:8]	RESERVED										0x00
0x22	OUT_OPERATING_MODE	[7:0]	RESERVED				OUT_OP_MODE_CH_3	OUT_OP_MODE_CH_2	OUT_OP_MODE_CH_1	OUT_OP_MODE_CH_0	0x00	R/W	
0x23		[15:8]	RESERVED								0x00		
0x24	OUT_SWITCH_EN	[7:0]	RESERVED				OUT_SWITCH_EN_CH_3	OUT_SWITCH_EN_CH_2	OUT_SWITCH_EN_CH_1	OUT_SWITCH_EN_CH_0	0x00	R/W	
0x25		[15:8]	RESERVED								0x00		
0x26	OUT_EN	[7:0]	RESERVED				HW_SHUTDOWN_EN_CH_3	HW_SHUTDOWN_EN_CH_2	HW_SHUTDOWN_EN_CH_1	HW_SHUTDOWN_EN_CH_0	0x00	R/W	
0x27		[15:8]	RESERVED								0x00		
0x28 to 0x2E by 2	OUT_RANGE_CHn	[7:0]	RESERVED						OUT_RANGE_CH		0x00	R/W	
0x29 to 0x2F by 2		[15:8]	RESERVED										0x00
0x30	FUNC_EN	[7:0]	RESERVED				FUNC_EN_CH_3	FUNC_EN_CH_2	FUNC_EN_CH_1	FUNC_EN_CH_0	0x00	R/W	
0x31		[15:8]	RESERVED								0x00		
0x32 to 0x38 by 2	FUNC_MODE_SEL_CHn	[7:0]	RESERVED								FUNC_MODE_CH	0x00	R/W
0x33 to 0x39 by 2		[15:8]	RESERVED								0x00		
0x3A to 0x40 by 2	FUNC_DAC_INPUT_B_CHn	[7:0]	DAC_INPUT_B_CH[7:0]								0x00	R/W	
0x3B to 0x41 by 2		[15:8]	DAC_INPUT_B_CH[15:8]										0x00
0x42 to 0x48 by 2	FUNC_DITHER_PERIOD_CHn	[7:0]	RESERVED					DITHER_PERIOD_CH			0x00	R/W	
0x43 to		[15:8]	RESERVED										0x00

## REGISTERS

Table 27. Register Bitwise Summary (Continued)

Reg	Name	Bits	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset	RW
0x49 by 2												
0x4A to 0x50 by 2	FUNC_DITHER_PHASE_CHn	[7:0]	RESERVED						DITHER_PHASE_CH		0x00	R/W
0x4B to 0x51 by 2		[15:8]	RESERVED									
0x54	MUX_OUT_SEL	[7:0]	MUX_OUT_EN	RESERVED		MUX_PARAM_SEL					0x00	R/W
0x55		[15:8]	RESERVED								0x00	
0x56	MUX_OUT_CONTROL	[7:0]	RESERVED							BUFFER_EN	0x01	R/W
0x57		[15:8]	RESERVED								0x00	
0x58	TEMP_WARN_INT_EN	[7:0]	RESERVED				TEMP_WARN_INT_EN_CH_3	TEMP_WARN_INT_EN_CH_2	TEMP_WARN_INT_EN_CH_1	TEMP_WARN_INT_EN_CH_0	0x00	R/W
0x59		[15:8]	RESERVED								0x00	
0x5A	MULTI_DAC_SW_LDAC	[7:0]	RESERVED							MULTI_DAC_SW_LDAC	0x00	R/W
0x5B		[15:8]	RESERVED								0x00	
0x5C	MULTI_DAC_INPUT_A	[7:0]	MULTI_DAC_INPUT_A[7:0]								0x00	R/W
0x5D		[15:8]	MULTI_DAC_INPUT_A[15:8]								0x00	
0x5E	DAC_SW_LDAC	[7:0]	RESERVED				DAC_SW_LDAC_CH_3	DAC_SW_LDAC_CH_2	DAC_SW_LDAC_CH_1	DAC_SW_LDAC_CH_0	0x00	R/W
0x5F		[15:8]	RESERVED								0x00	
0x60 to 0x66 by 2	DAC_INPUT_A_CHn	[7:0]	DAC_INPUT_A_CH[7:0]								0x00	R/W
0x61 to 0x67 by 2		[15:8]	DAC_INPUT_A_CH[15:8]								0x00	
0x68 to 0x6E by 2	DAC_DATA_READBACK_CHn	[7:0]	DAC_DATA_READBACK_CH[7:0]								0x00	R
0x69 to 0x6F by 2		[15:8]	DAC_DATA_READBACK_CH[15:8]								0x00	
0x70	TEMP_WARN_STAT	[7:0]	RESERVED				TEMP_WARN_FLAG_CH_3	TEMP_WARN_FLAG_CH_2	TEMP_WARN_FLAG_CH_1	TEMP_WARN_FLAG_CH_0	0x00	R/W
0x71		[15:8]	RESERVED								0x00	
0x72	DIGITAL_STATUS	[7:0]	RESERVED				FUSE_ERROR	INTERFACE_ERROR	POWERUP_COMPLETE	RESET_OCCURRED	0x01	R/W

## REGISTERS

Table 27. Register Bitwise Summary (Continued)

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

## REGISTER DETAILS

## Interface Configuration A Register

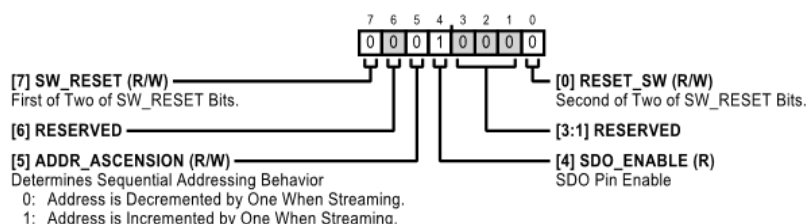


Figure 76. Address: 0x00, Reset: 0x10, Name: INTERFACE\_CONFIG\_A

Table 28. Bit Descriptions for INTERFACE\_CONFIG\_A

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

## Interface Configuration B Register

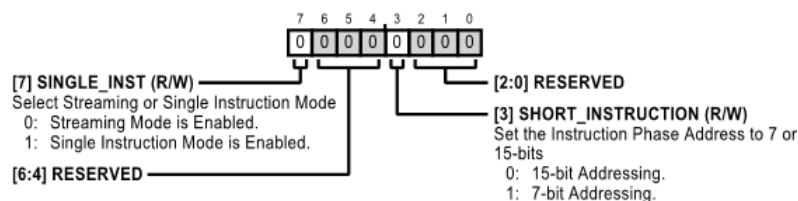


Figure 77. Address: 0x01, Reset: 0x00, Name: INTERFACE\_CONFIG\_B

Table 29. Bit Descriptions for INTERFACE\_CONFIG\_B

Bits	Bit Name	Description	Reset	Access
7	SINGLE_INST	Select Streaming or Single Instruction Mode.	0x0	R/W

## REGISTERS

Table 29. Bit Descriptions for INTERFACE\_CONFIG\_B (Continued)

Bits	Bit Name	Description	Reset	Access
		0: Streaming mode is enabled. The address increments/decrements as successive data bytes are received. 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. 0: 15-bit addressing. 1: 7-bit addressing.	0x0	R/W
[2:0]	RESERVED	Reserved.	0x0	R

## Device Configuration Register

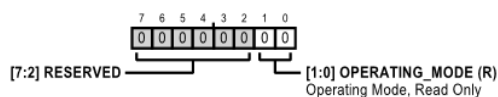


Figure 78. Address: 0x02, Reset: 0x00, Name: DEVICE\_CONFIG

Table 30. Bit Descriptions for DEVICE\_CONFIG

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

## Chip Type Register

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

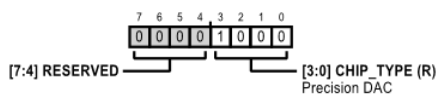


Figure 79. Address: 0x03, Reset: 0x08, Name: CHIP\_TYPE

Table 31. Bit Descriptions for CHIP\_TYPE

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

## Product ID Low Register

Low byte of the Product ID

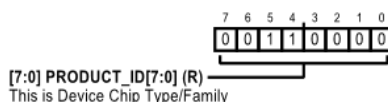


Figure 80. Address: 0x04, Reset: 0x30, Name: PRODUCT\_ID\_L

Table 32. Bit Descriptions for PRODUCT\_ID\_L

Bits	Bit Name	Description	Reset	Access
[7:0]	PRODUCT_ID[7:0]	This is Device Chip Type/Family. The product ID should be used in conjunction with chip type to identify a product.	0x30	R

# REGISTERS

## Product ID High Register

High byte of the Product ID

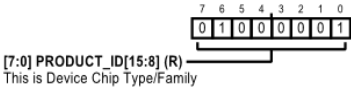


Figure 81. Address: 0x05, Reset: 0x41, Name: PRODUCT\_ID\_H

Table 33. Bit Descriptions for PRODUCT\_ID\_H

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

## Chip Grade Register

Identifies product variations and device revisions

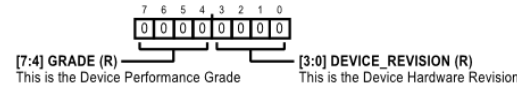


Figure 82. Address: 0x06, Reset: 0x00, Name: CHIP\_GRADE

Table 34. Bit Descriptions for CHIP\_GRADE

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

## Scratch Pad Register

This may be used to test writes and reads



Figure 83. Address: 0x0A, Reset: 0x00, Name: SCRATCH\_PAD

Table 35. 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

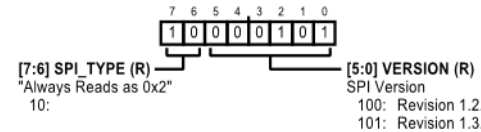


Figure 84. Address: 0x0B, Reset: 0x85, Name: SPI\_REVISION

## REGISTERS

Table 36. Bit Descriptions for SPI\_REVISION

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

## Vendor ID Low Register

Low byte of the Vendor ID

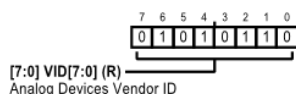


Figure 85. Address: 0x0C, Reset: 0x56, Name: VENDOR\_L

Table 37. Bit Descriptions for VENDOR\_L

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

## Vendor ID High Register

High byte of the Vendor ID

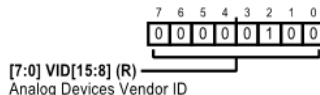


Figure 86. Address: 0x0D, Reset: 0x04, Name: VENDOR\_H

Table 38. 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

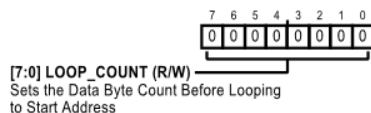


Figure 87. Address: 0x0E, Reset: 0x00, Name: STREAM\_MODE

Table 39. Bit Descriptions for STREAM\_MODE

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

## REGISTERS

## Transfer Configuration Register

Controls how data moves between controller and target registers

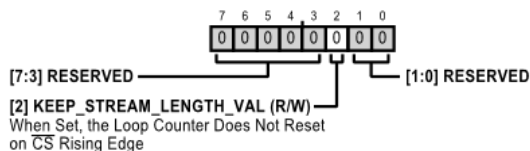


Figure 88. Address: 0x0F, Reset: 0x00, Name: TRANSFER\_CONFIG

Table 40. 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 CS rising edge.	0x0	R/W
[1:0]	RESERVED	Reserved.	0x0	R

## Interface Configuration C Register

Additional Interface Configuration Settings

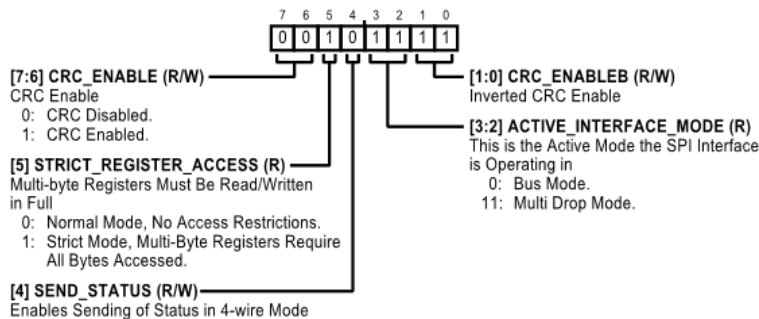


Figure 89. Address: 0x10, Reset: 0x2F, Name: INTERFACE\_CONFIG\_C

Table 41. 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. 0: CRC disabled. 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 multi-byte register must be read/written in full. 0: Normal mode, No Access Restrictions. 1: Strict mode, multibyte registers require all bytes accessed.	0x1	R
4	SEND_STATUS	Enables Sending of Status in 4-Wire Mode. When set, status information is sent by the device on SDO during the instruction phase. When clear, no status is sent during the instruction phase.	0x0	R/W
[3:2]	ACTIVE_INTERFACE_MODE	This is the Active Mode the SPI Interface is Operating in. 00: Bus mode. 11: Multi drop mode.	0x3	R
[1:0]	CRC_ENABLEB	Inverted CRC Enable. This must be written with the inverted value of the CRC_ENABLE.	0x3	R/W

## REGISTERS

## Interface Status A Register

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

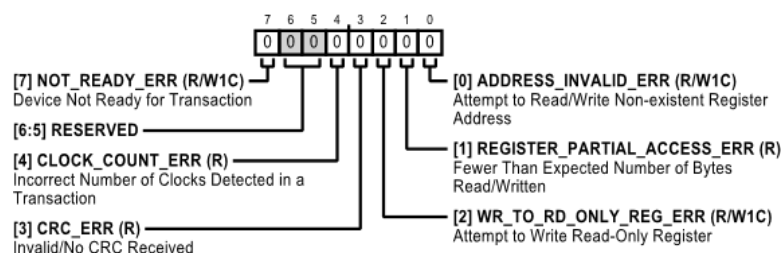


Figure 90. Address: 0x11, Reset: 0x00, Name: INTERFACE\_STATUS\_A

Table 42. Bit Descriptions for INTERFACE\_STATUS\_A

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

## Multi-DAC Channel Select Register

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

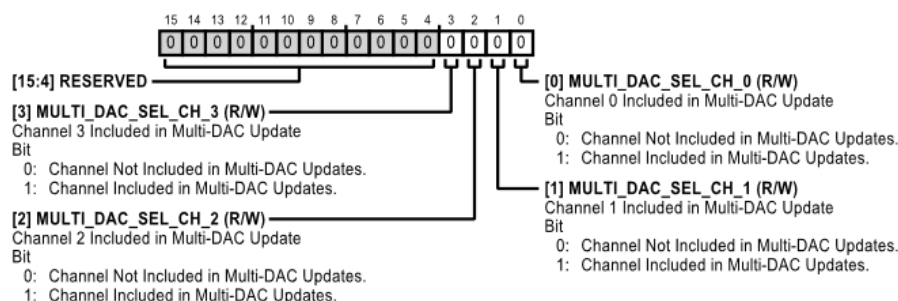


Figure 91. Address: 0x14, Reset: 0x0000, Name: MULTI\_DAC\_CH\_SEL

Table 43. Bit Descriptions for MULTI\_DAC\_CH\_SEL

Bits	Bit Name	Description	Reset	Access
[15:4]	RESERVED	Reserved.	0x0	R
3	MULTI_DAC_SEL_CH_3	Channel 3 Included in Multi-DAC Update Bit. 0: Channel not included in multi-DAC updates.	0x0	R/W

## REGISTERS

Table 43. Bit Descriptions for MULTI\_DAC\_CH\_SEL (Continued)

Bits	Bit Name	Description	Reset	Access
2	MULTI_DAC_SEL_CH_2	1: Channel included in multi-DAC updates. 0: Channel not included in multi-DAC updates.	0x0	R/W
1	MULTI_DAC_SEL_CH_1	1: Channel included in multi-DAC updates. 0: Channel not included in multi-DAC updates.	0x0	R/W
0	MULTI_DAC_SEL_CH_0	1: Channel included in multi-DAC updates. 0: Channel not included in multi-DAC updates.	0x0	R/W

## Sync-Async Load Register

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

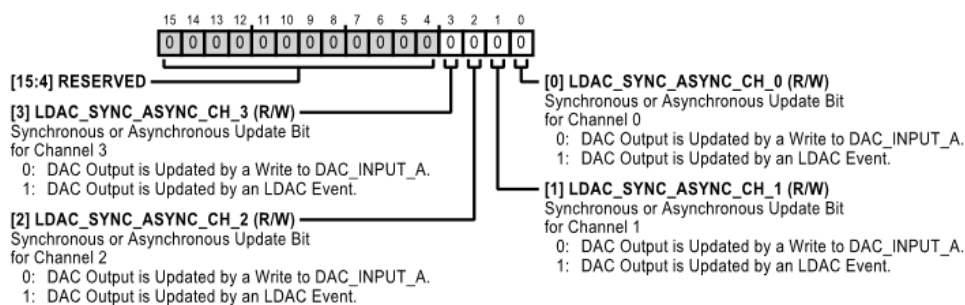


Figure 92. Address: 0x16, Reset: 0x0000, Name: LDAC\_SYNC\_ASYNC

Table 44. Bit Descriptions for LDAC\_SYNC\_ASYNC

Bits	Bit Name	Description	Reset	Access
[15:4]	RESERVED	Reserved.	0x0	R
3	LDAC_SYNC_ASYNC_CH_3	Synchronous or Asynchronous Update Bit for Channel 3. 0: DAC output is updated by a write to DAC_INPUT_A. 1: DAC output is updated by an LDAC event.	0x0	R/W
2	LDAC_SYNC_ASYNC_CH_2	Synchronous or Asynchronous Update Bit for Channel 2. 0: DAC output is updated by a write to DAC_INPUT_A. 1: DAC output is updated by an LDAC event.	0x0	R/W
1	LDAC_SYNC_ASYNC_CH_1	Synchronous or Asynchronous Update Bit for Channel 1. 0: DAC output is updated by a write to DAC_INPUT_A. 1: DAC output is updated by an LDAC event.	0x0	R/W
0	LDAC_SYNC_ASYNC_CH_0	Synchronous or Asynchronous Update Bit for Channel 0. 0: DAC output is updated by a write to DAC_INPUT_A. 1: DAC output is updated by an LDAC event.	0x0	R/W

## Hardware-Software Trigger Register

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

# REGISTERS

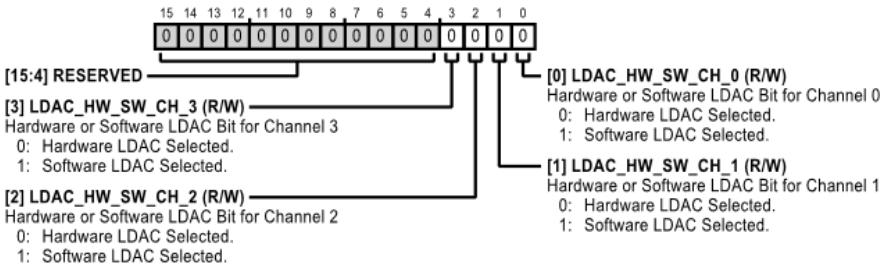


Figure 93. Address: 0x18, Reset: 0x0000, Name: LDAC\_HW\_SW

Table 45. Bit Descriptions for LDAC\_HW\_SW

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

## LDAC Edge Select Register

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

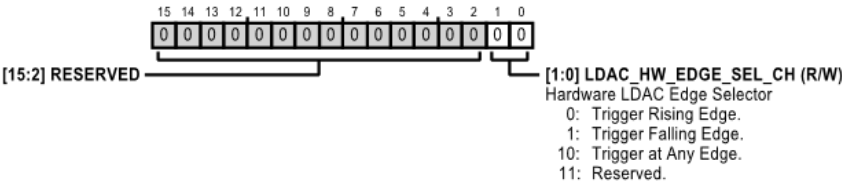


Figure 94. Address: 0x1A to 0x20 (Increments of 2), Reset: 0x0000, Name: LDAC\_EDGE\_SEL\_CHn

Table 46. Bit Descriptions for LDAC\_EDGE\_SEL\_CHn

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

## Output Operating Mode Register

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

## REGISTERS

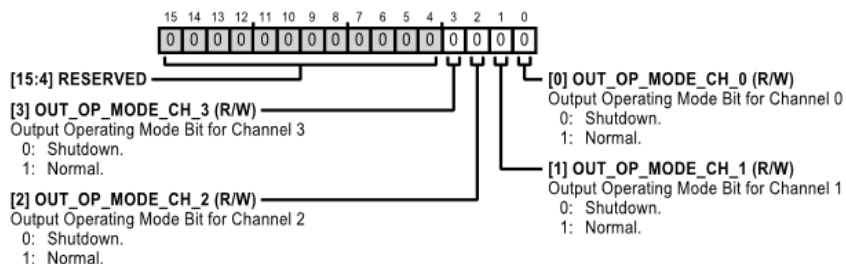


Figure 95. Address: 0x22, Reset: 0x0000, Name: OUT\_OPERATING\_MODE

Table 47. Bit Descriptions for OUT\_OPERATING\_MODE

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

## Output Switch Enable Register

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

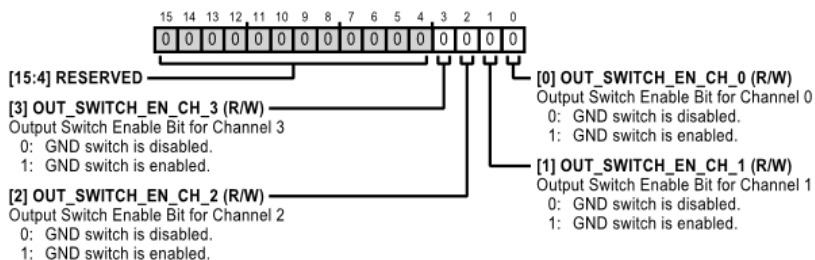


Figure 96. Address: 0x24, Reset: 0x0000, Name: OUT\_SWITCH\_EN

Table 48. Bit Descriptions for OUT\_SWITCH\_EN

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

## REGISTERS

Table 48. Bit Descriptions for OUT\_SWITCH\_EN (Continued)

Bits	Bit Name	Description	Reset	Access
1	OUT_SWITCH_EN_CH_1	Output Switch Enable Bit for Channel 1. 0: GND switch is disabled. 1: GND switch is enabled.	0x0	R/W
0	OUT_SWITCH_EN_CH_0	Output Switch Enable Bit for Channel 0. 0: GND switch is disabled. 1: GND switch is enabled.	0x0	R/W

## OUT\_EN Pin Enable Register

This register is used to enable the shutdown of the output stage with the OUT\_EN pin

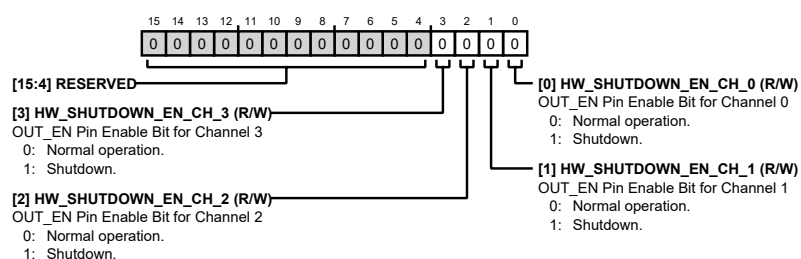


Figure 97. Address: 0x26, Reset: 0x0000, Name: OUT\_EN

Table 49. Bit Descriptions for OUT\_EN

Bits	Bit Name	Description	Reset	Access
[15:4]	RESERVED	Reserved.	0x0	R
3	HW_SHUTDOWN_EN_CH_3	OUT_EN Pin Enable Bit for Channel 3. 0: Normal operation. 1: IOUT3 output disable.	0x0	R/W
2	HW_SHUTDOWN_EN_CH_2	OUT_EN Pin Enable Bit for Channel 2. 0: Normal operation. 1: IOUT2 output disable.	0x0	R/W
1	HW_SHUTDOWN_EN_CH_1	OUT_EN Pin Enable Bit for Channel 1. 0: Normal operation. 1: IOUT1 output disable.	0x0	R/W
0	HW_SHUTDOWN_EN_CH_0	OUT_EN Pin Enable Bit for Channel 0. 0: Normal operation. 1: IOUT0 output disable.	0x0	R/W

## Output Range Register

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

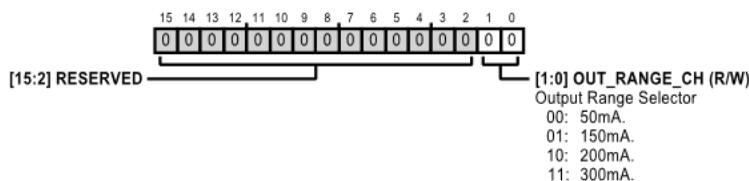


Figure 98. Address: 0x28 to 0x2E (Increments of 2), Reset: 0x0000, Name: OUT\_RANGE\_CHn

## REGISTERS

Table 50. Bit Descriptions for OUT\_RANGE\_CHn

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

## Function Enable Register

This register is used to enable the other DAC functions

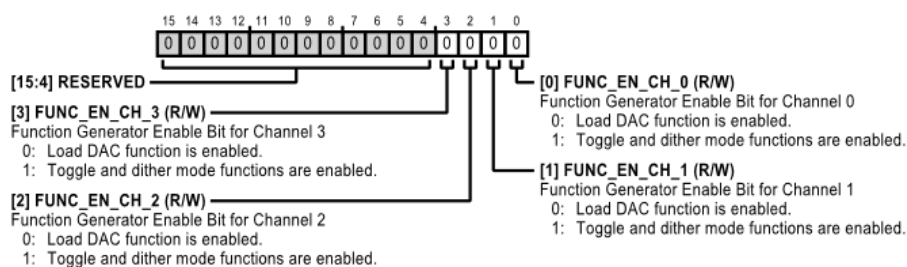


Figure 99. Address: 0x30, Reset: 0x0000, Name: FUNC\_EN

Table 51. Bit Descriptions for FUNC\_EN

Bits	Bit Name	Description	Reset	Access
[15:4]	RESERVED	Reserved.	0x0	R
3	FUNC_EN_CH_3	Function Generator Enable Bit for Channel 3. 0: Load DAC function is enabled. 1: Toggle and dither mode functions are enabled.	0x0	R/W
2	FUNC_EN_CH_2	Function Generator Enable Bit for Channel 2. 0: Load DAC function is enabled. 1: Toggle and dither mode functions are enabled.	0x0	R/W
1	FUNC_EN_CH_1	Function Generator Enable Bit for Channel 1. 0: Load DAC function is enabled. 1: Toggle and dither mode functions are enabled.	0x0	R/W
0	FUNC_EN_CH_0	Function Generator Enable Bit for Channel 0. 0: Load DAC function is enabled. 1: Toggle and dither mode functions are enabled.	0x0	R/W

## Function Mode Selector Register

This register is used to select the active DAC function

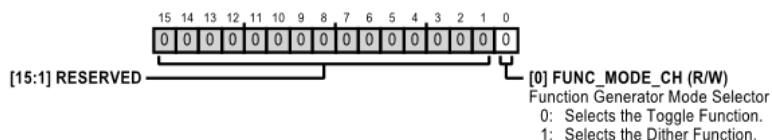


Figure 100. Address: 0x32 to 0x38 (Increments of 2), Reset: 0x0000, Name: FUNC\_MODE\_SEL\_CHn

## REGISTERS

Table 52. Bit Descriptions for FUNC\_MODE\_SEL\_CHn

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

## DAC Input B Register

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

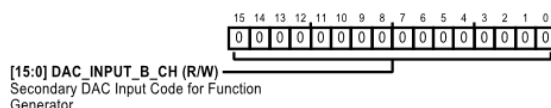


Figure 101. Address: 0x3A to 0x40 (Increments of 2), Reset: 0x0000, Name: FUNC\_DAC\_INPUT\_B\_CHn

Table 53. Bit Descriptions for FUNC\_DAC\_INPUT\_B\_CHn

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

## Dither Settings Register

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

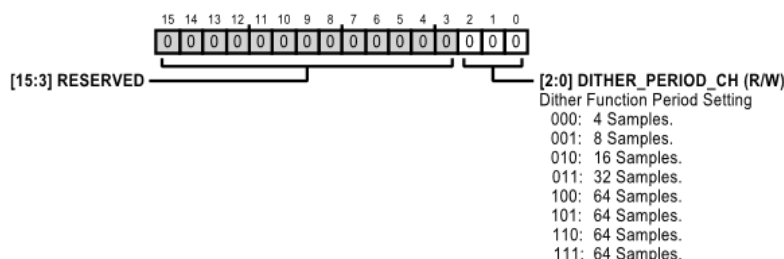


Figure 102. Address: 0x42 to 0x48 (Increments of 2), Reset: 0x0000, Name: FUNC\_DITHER\_PERIOD\_CHn

Table 54. Bit Descriptions for FUNC\_DITHER\_PERIOD\_CHn

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

## Dither Settings Register

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

# REGISTERS

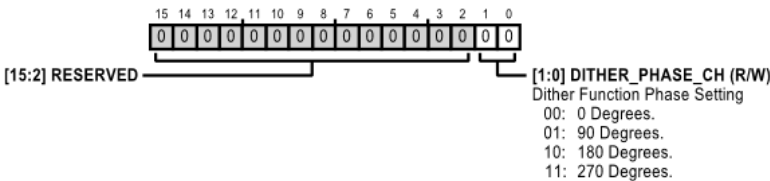


Figure 103. Address: 0x4A to 0x50 (Increments of 2), Reset: 0x0000, Name: FUNC\_DITHER\_PHASE\_CHn

Table 55. Bit Descriptions for FUNC\_DITHER\_PHASE\_CHn

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

## MUX Out Select Register

This register is used to control the MUX\_OUT pin

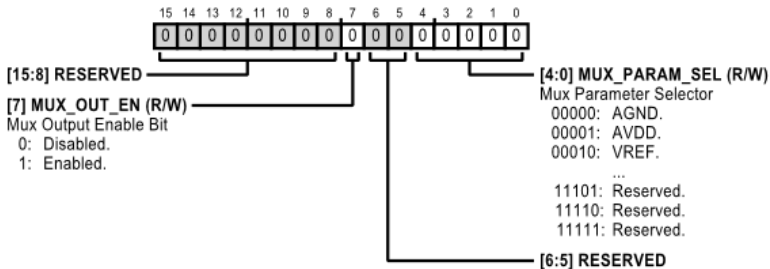


Figure 104. Address: 0x54, Reset: 0x0000, Name: MUX\_OUT\_SEL

Table 56. Bit Descriptions for MUX\_OUT\_SEL

Bits	Bit Name	Description	Reset	Access
[15:8]	RESERVED	Reserved.	0x0	R
7	MUX_OUT_EN	Mux Output Enable Bit. 0: Disabled. 1: Enabled.	0x0	R/W
[6:5]	RESERVED	Reserved.	0x0	R
[4:0]	MUX_PARAM_SEL	Mux Parameter Selector. 00000: AGND. 00001: AVDD. 00010: VREF. 00011: AGND. 00100: DAC VOUT CH0. 00101: DAC VOUT CH1. 00110: DAC VOUT CH2. 00111: DAC VOUT CH3. 01000: DAC IOUT CH0. 01001: DAC IOUT CH1. 01010: DAC IOUT CH2.	0x0	R/W

# REGISTERS

Table 56. Bit Descriptions for MUX\_OUT\_SEL (Continued)

Bits	Bit Name	Description	Reset	Access
		01011: DAC IOUT CH3.		
		01100: DAC PVDD CH0.		
		01101: DAC PVDD CH1.		
		01110: DAC PVDD CH2.		
		01111: DAC PVDD CH3.		
		10000: DAC TDIODE CH0.		
		10001: DAC TDIODE CH1.		
		10010: DAC TDIODE CH2.		
		10011: DAC TDIODE CH3.		
		10100: MUX_IN.		
		10101: MUX_IN.		
		10110: MUX_IN.		
		10111: MUX_IN.		
		11000: Reserved.		
		11001: Reserved.		
		11010: Reserved.		
		11011: Reserved.		
		11100: Reserved.		
		11101: Reserved.		
		11110: Reserved.		
		11111: Reserved.		

## MUX Out Control Register

This register is used to control the buffer on the MUX\_OUT pin

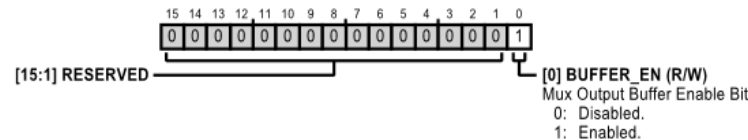


Figure 105. Address: 0x56, Reset: 0x0001, Name: MUX\_OUT\_CONTROL

Table 57. Bit Descriptions for MUX\_OUT\_CONTROL

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

## Temperature Warning Interrupt Enabled Register

This register is used to enable the status bits in the TEMP\_WARN\_STAT register

# REGISTERS

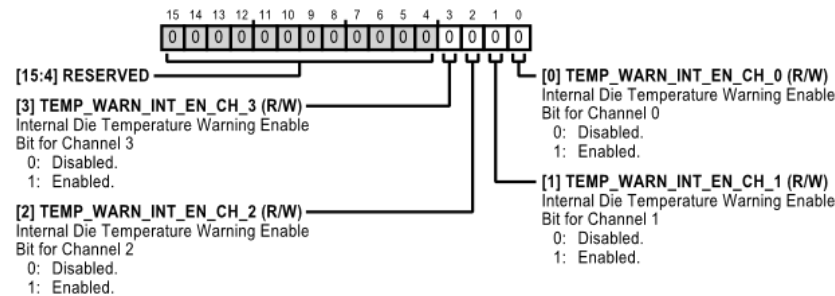


Figure 106. Address: 0x58, Reset: 0x0000, Name: TEMP\_WARN\_INT\_EN

Table 58. Bit Descriptions for TEMP\_WARN\_INT\_EN

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

## Software LDAC for Multiple DAC Input Register

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

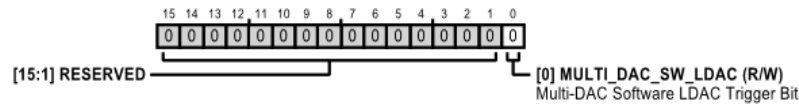


Figure 107. Address: 0x5A, Reset: 0x0000, Name: MULTI\_DAC\_SW\_LDAC

Table 59. Bit Descriptions for MULTI\_DAC\_SW\_LDAC

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

## Multiple DAC Input Register

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

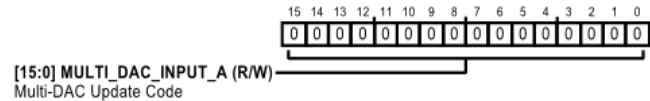


Figure 108. Address: 0x5C, Reset: 0x0000, Name: MULTI\_DAC\_INPUT\_A

## REGISTERS

Table 60. Bit Descriptions for MULTI\_DAC\_INPUT\_A

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

## Software LDAC for DAC Input Register

This register is used to trigger the software LDAC mode

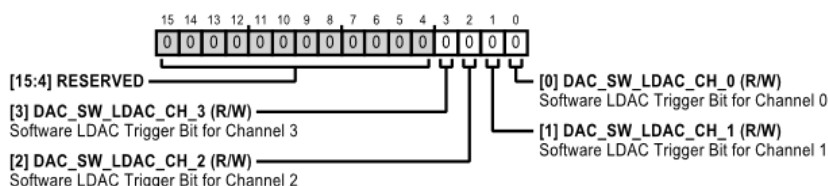


Figure 109. Address: 0x5E, Reset: 0x0000, Name: DAC\_SW\_LDAC

Table 61. Bit Descriptions for DAC\_SW\_LDAC

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

## DAC Input Register

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

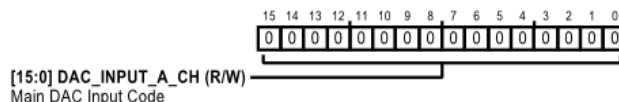


Figure 110. Address: 0x60 to 0x66 (Increments of 2), Reset: 0x0000, Name: DAC\_INPUT\_A\_CHn

Table 62. Bit Descriptions for DAC\_INPUT\_A\_CHn

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

## DAC Data Readback Register

This register contains the current DAC code in the DAC core

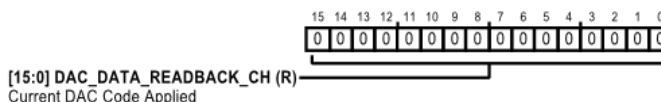


Figure 111. Address: 0x68 to 0x6E (Increments of 2), Reset: 0x0000, Name: DAC\_DATA\_READBACK\_CHn

Table 63. Bit Descriptions for DAC\_DATA\_READBACK\_CHn

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

## REGISTERS

## Temperature Warning Flags Status Register

This register contains the status of the internal die temperature

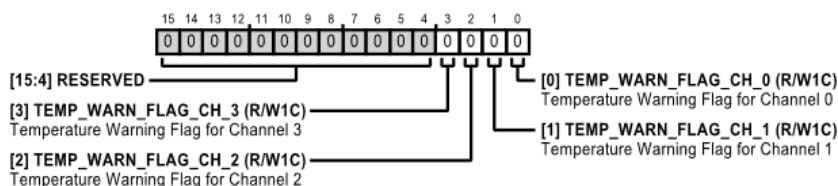


Figure 112. Address: 0x70, Reset: 0x0000, Name: TEMP\_WARN\_STAT

Table 64. Bit Descriptions for TEMP\_WARN\_STAT

Bits	Bit Name	Description	Reset	Access
[15:4]	RESERVED	Reserved.	0x0	R
3	TEMP_WARN_FLAG_CH_3	Temperature Warning Flag for Channel 3. This bit is set to 1'b1 when the internal die temperature of channel 3 reaches approximately 125°C. This bit can be cleared by writing a 1'b1, as long as the temperature is below 125°C.	0x0	R/W1C
2	TEMP_WARN_FLAG_CH_2	Temperature Warning Flag for Channel 2. This bit is set to 1'b1 when the internal die temperature of channel 2 reaches approximately 125°C. This bit can be cleared by writing a 1'b1, as long as the temperature is below 125°C.	0x0	R/W1C
1	TEMP_WARN_FLAG_CH_1	Temperature Warning Flag for Channel 1. This bit is set to 1'b1 when the internal die temperature of channel 1 reaches approximately 125°C. This bit can be cleared by writing a 1'b1, as long as the temperature is below 125°C.	0x0	R/W1C
0	TEMP_WARN_FLAG_CH_0	Temperature Warning Flag for Channel 0. This bit is set to 1'b1 when the internal die temperature of channel 0 reaches approximately 125°C. This bit can be cleared by writing a 1'b1, as long as the temperature is below 125°C.	0x0	R/W1C

## Digital Status Register

This register contains the status signals of the digital core

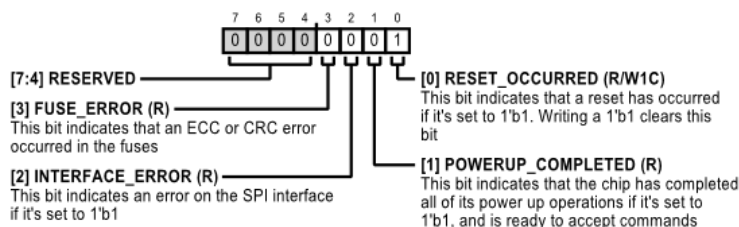


Figure 113. Address: 0x72, Reset: 0x01, Name: DIGITAL\_STATUS

Table 65. Bit Descriptions for DIGITAL\_STATUS

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

# REGISTERS

## Bandgap Control Register

This register contains the control for the internal bandgap

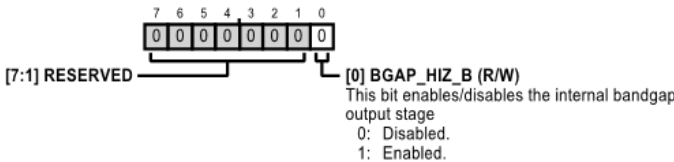


Figure 114. Address: 0x73, Reset: 0x00, Name: BANDGAP\_CONTROL

Table 66. Bit Descriptions for BANDGAP\_CONTROL

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

## User Spare 0 Register

Spare register

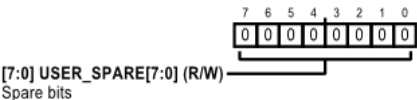


Figure 115. Address: 0x74, Reset: 0x00, Name: USER\_SPARE\_0

Table 67. Bit Descriptions for USER\_SPARE\_0

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

## User Spare 1 Register

Spare register

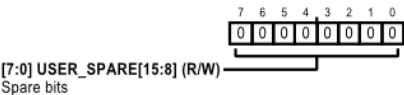


Figure 116. Address: 0x75, Reset: 0x00, Name: USER\_SPARE\_1

Table 68. Bit Descriptions for USER\_SPARE\_1

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

## User Spare 2 Register

Spare register

REGISTERS

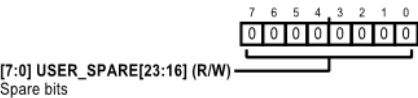


Figure 117. Address: 0x76, Reset: 0x00, Name: USER\_SPARE\_2

Table 69. Bit Descriptions for USER\_SPARE\_2

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

User Spare 3 Register

Spare register

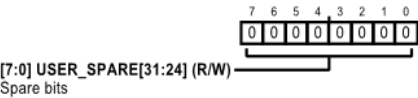
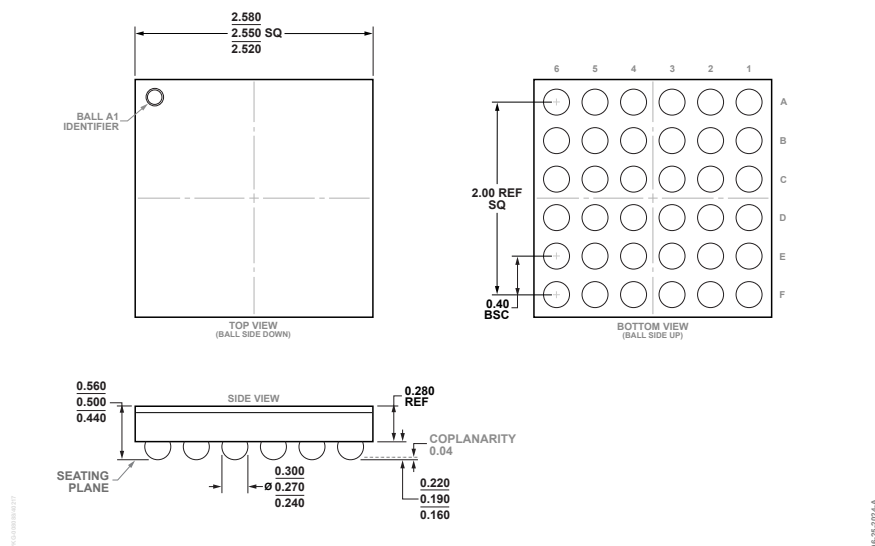


Figure 118. Address: 0x77, Reset: 0x00, Name: USER\_SPARE\_3

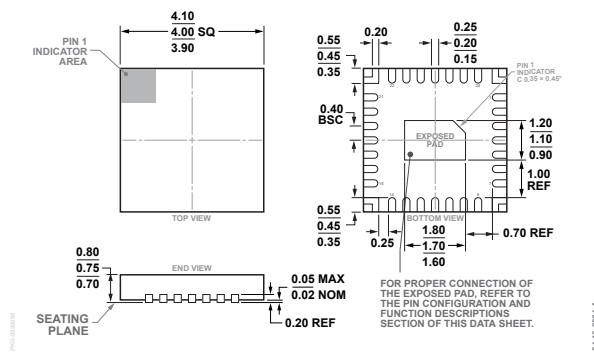
Table 70. Bit Descriptions for USER\_SPARE\_3

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

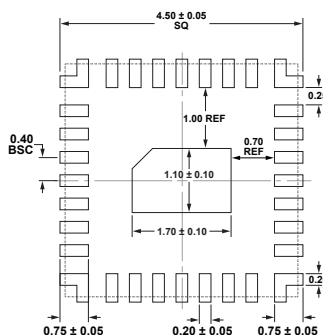
## OUTLINE DIMENSIONS



**Figure 119. 36-Ball Wafer Level Chip Scale Package (WLCSP)  
(CB-36-12)**  
Dimensions shown in millimeters



**RECOMMENDED SOLDER PAD LAYOUT  
(TOP VIEW)**



**Figure 120. 28-Lead Lead Frame Chip Scale Package [LFCSP]  
4mm × 4mm Body and 0.75 mm Package Height  
(CP-28-17)**  
Dimensions shown in millimeters

## OUTLINE DIMENSIONS

## ORDERING GUIDE

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

<sup>1</sup> Z = RoHS Compliant Part.

**Legal Terms and Conditions**

Information furnished by Analog Devices is believed to be accurate and reliable "as is". However, no responsibility is assumed by Analog Devices for its use, nor for any infringements of patents or other rights of third parties that may result from its use. Specifications subject to change without notice. No license is granted by implication or otherwise under any patent or patent rights of Analog Devices. Trademarks and registered trademarks are the property of their respective owners. All Analog Devices products contained herein are subject to release and availability.