

# 16-Bit, 625 MSPS, 2x Interpolating, Dual-Channel Digital-to-Analog Converter (DAC)

Check for Samples: [DAC3282](#)

## FEATURES

- Dual, 16-Bit, 625 MSPS DACs
- 8-Bit Input LVDS Data Bus
  - Byte-Wide Interleaved Data Load
  - 8 Sample Input FIFO
  - Optional Data Pattern Checker
- Multi-DAC Synchronization
- Optional 2x Interpolation Filter
- Zero-IF Sinc Correction Filter
- $F_s/2$  and  $\pm F_s/4$  Coarse Mixer
- Digital Offset Adjustment for LO Correction
- Temperature Sensor
- 3- or 4-Wire Serial Control Interface
- On Chip 1.2-V reference
- Differential Scalable Output: 2 to 20 mA
- Low Power: 950 mW at 625 MSPS, 845 mW at 500 MHz, Full Operating Conditions
- Space Saving Package: 48-pin 7x7mm QFN

## APPLICATIONS

- Cellular Base Stations
- Diversity Transmit
- Wideband Communications
- Digital Synthesis

## DESCRIPTION

The DAC3282 is a dual-channel 16-bit 625 MSPS digital-to-analog converter (DAC) with an 8-bit LVDS input data bus with on-chip termination, optional 2x interpolation filter, and internal voltage reference. The DAC3282 offers superior linearity, noise and crosstalk performance.

Input data can be interpolated by 2x through an on-chip interpolating FIR filter with over 85 dB of stop-band attenuation. Multiple DAC3282 devices can be fully synchronized.

The DAC3282 allows either a complex or real output. An optional coarse mixer in complex mode provides frequency upconversion and the dual DAC output produces a complex Hilbert Transform pair. The digital offset correction feature allows optimization of LO feed-through of an external quadrature modulator performing the final single sideband RF up-conversion.

The DAC3282 is characterized for operation over the entire industrial temperature range of  $-40^{\circ}\text{C}$  to  $85^{\circ}\text{C}$  and is available in a 48-pin 7x7mm QFN package.

## ORDERING INFORMATION

$T_A$	ORDER CODE	PACKAGE DRAWING/TYPE <sup>(1)</sup> (2)	TRANSPORT MEDIA	QUANTITY
$-40^{\circ}\text{C}$ to $85^{\circ}\text{C}$	DAC3282IRGZT	RGZ / 48QFN Quad Flatpack No-Lead	Tape and Reel	250
	DAC3282IRGZR			2500

(1) Thermal Pad Size: 5,6 mm x 5,6 mm

(2) MSL Peak Temperature: Level-3-260C-168 HR



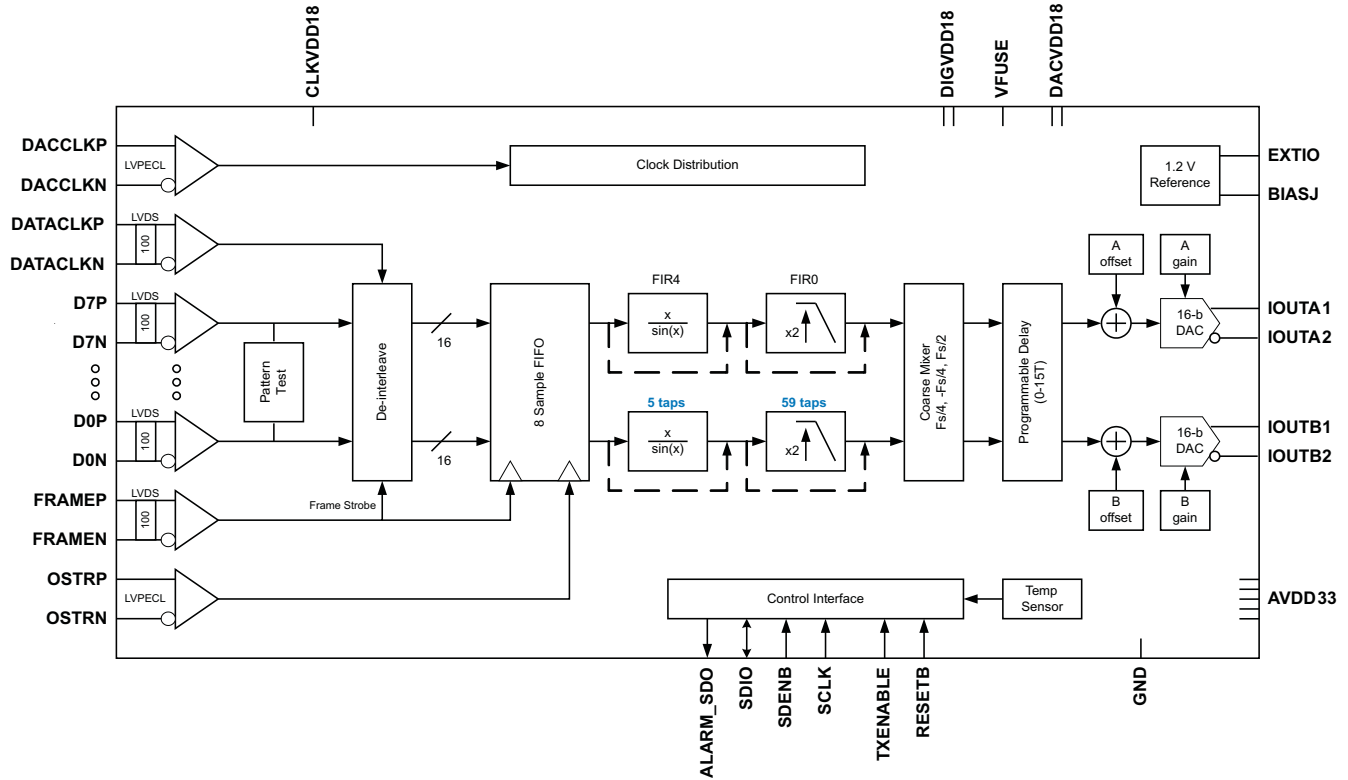
Please be aware that an important notice concerning availability, standard warranty, and use in critical applications of Texas Instruments semiconductor products and disclaimers thereto appears at the end of this data sheet.



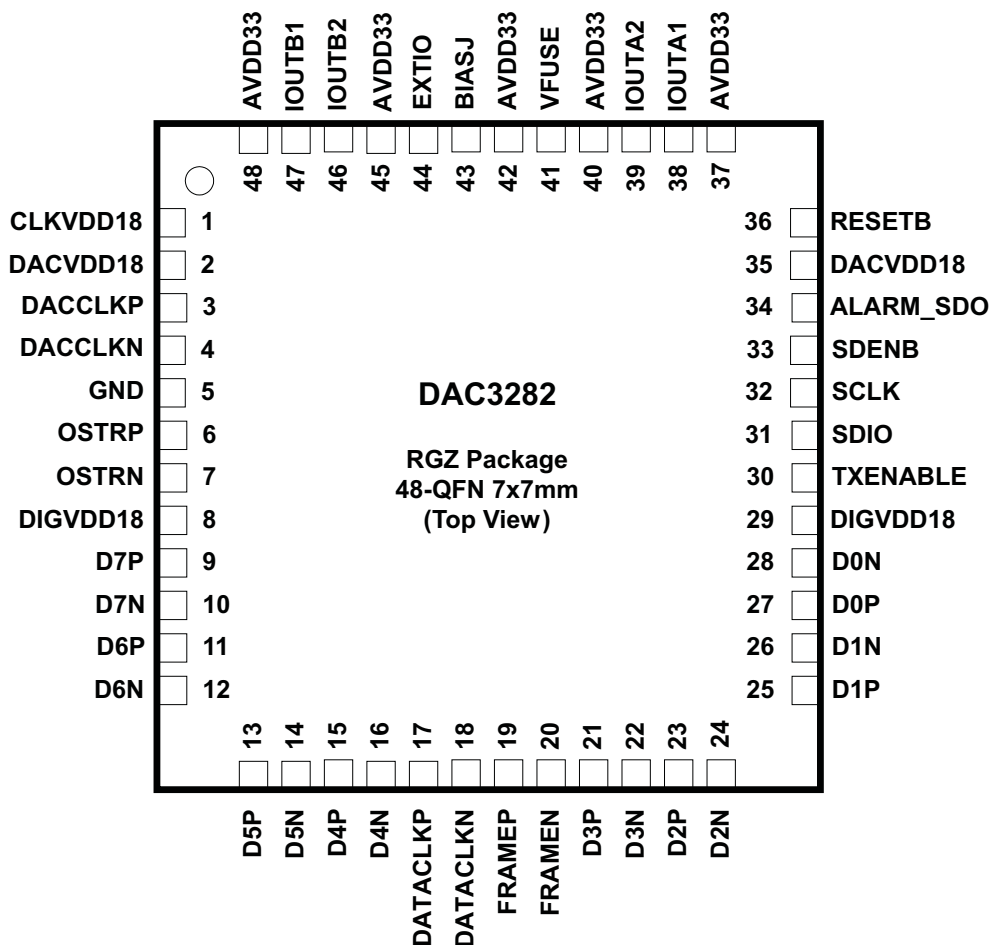
This integrated circuit can be damaged by ESD. Texas Instruments recommends that all integrated circuits be handled with appropriate precautions. Failure to observe proper handling and installation procedures can cause damage.

ESD damage can range from subtle performance degradation to complete device failure. Precision integrated circuits may be more susceptible to damage because very small parametric changes could cause the device not to meet its published specifications.

### FUNCTIONAL BLOCK DIAGRAM



PINOUT



PIN FUNCTIONS

PIN		I/O	DESCRIPTION
NAME	NO.		
AVDD33	37, 40, 42, 45, 48	I	Analog supply voltage. (3.3 V)
ALARM_SDO	34	O	1.8V CMOS output for ALARM condition. The ALARM output functionality is defined through the CONFIG6 register. Default polarity is active low, but can be changed to active high via CONFIG0 <b>alarm_pol</b> control bit. Optionally, it can be used as the uni-directional data output in 4-pin serial interface mode (CONFIG 23 <b>sif4_ena</b> = '1').
BIASJ	43	O	Full-scale output current bias. For 20mA full-scale output current, connect a 960 Ω resistor to GND.
CLKVDD18	1	I	Internal clock buffer supply voltage. (1.8 V) It is recommended to isolate this supply from DACVDD18 and DIGVDD18.
D[7..0]P	9, 11, 13, 15, 21, 23, 25, 27	I	LVDS positive input data bits 0 through 7. Each positive/negative LVDS pair has an internal 100 Ω termination resistor. Data format relative to DATACLKP/N clock is Double Data Rate (DDR) with two data transfers per DATAKP/N clock cycle. Dual channel 16-bit data is transferred byte-wide on this single 8-bit data bus using FRAMEP/N as a frame strobe indicator. D7P is most significant data bit (MSB) – pin 9 D0P is least significant data bit (LSB) – pin 27 The order of the bus can be reversed via CONFIG19 <b>rev</b> bit.
D[7..0]N	10, 12, 14, 16, 22, 24, 26, 28	I	LVDS negative input data bits 0 through 15. (See D[7:0]P description above) D7N is most significant data bit (MSB) – pin 10 D0N is least significant data bit (LSB) – pin 28
DACCLKP	3	I	Positive external LVPECL clock input for DAC core with a self-bias of approximately CLKVDD18/2.

**PIN FUNCTIONS (continued)**

PIN		I/O	DESCRIPTION
NAME	NO.		
DACCLKN	4	I	Complementary external LVPECL clock input for DAC core. (see the DACCLKP description)
DACVDD18	2, 35	I	DAC core supply voltage. (1.8 V) It is recommended to isolate this supply from CLKVDD18 and DIGVDD18.
DATACLKP	17	I	LVDS positive input data clock. This positive/negative pair has an internal 100 Ω termination resistor. Input data D[7:0]P/N is latched on both edges of DATACLKP/N (Double Data Rate) with two data transfers input per DATACLKP/N clock cycle.
DATACLKN	18	I	LVDS negative input data clock. (See DATACLKP description)
DIGVDD18	8, 29	I	Digital supply voltage. (1.8V) It is recommended to isolate this supply from CLKVDD18 and DACVDD18.
EXTIO	44	I/O	Used as external reference input when internal reference is disabled through CONFIG25 <b>extref_ena</b> = '1'. Used as internal reference output when CONFIG25 <b>extref_ena</b> = '0' (default). Requires a 0.1 μF decoupling capacitor to AGND when used as reference output.
FRAMEP	19	I	LVDS frame indicator positive input. This positive/negative pair has an internal 100 Ω termination resistor. This signal is captured with the rising edge of DATACLKP/N and used to indicate the beginning of the frame. It is also used as a reset signal by the FIFO. The FRAMEP/N signal should be edge-aligned with D[7:0]P/N.
FRAMEN	20	I	LVDS frame indicator negative input. (See the FRAMEN description)
GND	5, Thermal Pad	I	Pin 5 and the Thermal Pad located on the bottom of the QFN package is ground for all supplies.
IOUTA1	38	O	A-Channel DAC current output. An offset binary data pattern of 0x0000 at the DAC input results in a full scale current sink and the least positive voltage on the IOUTA1 pin. Similarly, a 0xFFFF data input results in a 0 mA current sink and the most positive voltage on the IOUTA1 pin.
IOUTA2	39	O	A-Channel DAC complementary current output. The IOUTA2 has the opposite behavior of the IOUTA1 described above. An input data value of 0x0000 results in a 0 mA sink and the most positive voltage on the IOUTA2 pin.
IOUTB1	47	O	B-Channel DAC current output. Refer to IOUTA1 description above.
IOUTB2	46	O	B-Channel DAC complementary current output. Refer to IOUTA2 description above.
OSTRP	6	I	LVPECL output strobe positive input. This positive/negative pair is captured with the rising edge of DACCLKP/N. It is used to reset the clock dividers and for multiple DAC synchronization. If unused it can be left floating.
OSTRN	7	I	LVPECL output strobe negative input. (See the OSTRP description)
RESETB	36	I	1.8V CMOS active low input for chip RESET. Internal pull-up.
SCLK	32	I	1.8V CMOS serial interface clock. Internal pull-down.
SDENB	33	I	1.8V CMOS active low serial data enable, always an input to the DAC3282. Internal pull-up.
SDIO	31	I/O	1.8V CMOS serial interface data. Bi-directional in 3-pin mode (default) and 4-pin mode. Internal pull-down.
TXENABLE	30	I	1.8V CMOS active high input. TXENABLE must be high for the DATA to the DAC to be enabled. When TXENABLE is low, the digital logic section is forced to all 0, and any input data is ignored. Internal pull-down.
VFUSE	41	I	Digital supply voltage. (1.8V) This supply pin is also used for factory fuse programming. <b>Connect to DACVDD18 pins for normal operation.</b>

## ABSOLUTE MAXIMUM RATINGS

 over operating free-air temperature range (unless otherwise noted)<sup>(1)</sup>

		VALUE	UNIT
Supply voltage range	DACVDD18 <sup>(2)</sup>	–0.5 to 2.3	V
	DIGVDD18 <sup>(2)</sup>	–0.5 to 2.3	V
	CLKVDD18 <sup>(2)</sup>	–0.5 to 2.3	V
	VFUSE <sup>(2)</sup>	–0.5 to 2.3	V
	AVDD33 <sup>(2)</sup>	–0.5 to 4	V
Terminal voltage range	CLKVDD18 to DIGVDD18	–0.5 to 0.5	V
	DACVDD18 to DIGVDD18	–0.5 to 0.5	V
	D[7..0]P ,D[7..0]N, DATACLKP,DATACLKN, FRAMEP, FRAMEN <sup>(2)</sup>	–0.5 to DIGVDD18 + 0.5	V
	DACCLKP, DACCLKN, OSTRP, OSTRN <sup>(2)</sup>	–0.5 to CLKVDD18 + 0.5	V
	ALARM_SDO, SDIO, SCLK, SDENB, RESETB, TXENABLE <sup>(2)</sup>	–0.5 to DIGVDD18 + 0.5	V
	IOUTA1/B1, IOUTA2/B2 <sup>(2)</sup>	–1.0 to AVDD33 + 0.5	V
	EXTIO, BIASJ <sup>(2)</sup>	–0.5 to AVDD33 + 0.5	V
Peak input current (any input)		20	mA
Peak total input current (all inputs)		–30	mA
Operating free-air temperature range, T <sub>A</sub> : DAC3282		–40 to 85	°C
Storage temperature range		–65 to 150	°C
Lead temperature 1,6 mm (1/16 inch) from the case for 10 seconds		260	°C

(1) Stresses beyond those listed under *absolute maximum ratings* may cause permanent damage to the device. These are stress ratings only and functional operation of these or any other conditions beyond those indicated under *recommended operating conditions* is not implied. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.

(2) Measured with respect to GND.

## THERMAL CHARACTERISTICS

over operating free-air temperature range (unless otherwise noted)

THERMAL CONDUCTIVITY		48ld QFN	UNIT
T <sub>J</sub>	Maximum Junction Temperature <sup>(1) (2)</sup>	125	°C
θ <sub>JA</sub>	Theta junction-to-ambient (still air)	30	°C/W
	Theta junction-to-ambient (150 lfm)	24	°C/W
θ <sub>JB</sub>	Theta junction-to-board	8	°C/W
θ <sub>JP</sub>	Theta junction-to-pad	1.3	°C/W

(1) Air flow or heat sinking reduces θ<sub>JA</sub> and may be required for sustained operation at 85° and maximum operating conditions.

(2) It is strongly recommended to solder the device thermal pad to the board ground plane.

## ELECTRICAL CHARACTERISTICS – DC SPECIFICATIONS<sup>(1)</sup>

over recommended operating free-air temperature range, nominal supplies,  $I_{OUT_{FS}} = 20 \text{ mA}$  (unless otherwise noted)

PARAMETER		TEST CONDITIONS	MIN	TYP	MAX	UNIT
Resolution			16			Bits
<b>DC ACCURACY</b>						
DNL	Differential nonlinearity	1 LSB = $I_{OUT_{FS}}/2^{16}$		±2		LSB
INL	Integral nonlinearity			±4		LSB
<b>ANALOG OUTPUT</b>						
Coarse gain linearity				±0.04		LSB
Offset error		Mid code offset		0.01		%FSR
Gain error		With external reference		±2		%FSR
		With internal reference		±2		%FSR
Gain mismatch		With internal reference	-2		2	%FSR
Minimum full scale output current		Nominal full-scale current, $I_{OUT_{FS}} = 16 \times I_{BIAS}$ current.		2		mA
Maximum full scale output current				20		
Output compliance range <sup>(2)</sup>		$I_{OUT_{FS}} = 20 \text{ mA}$	AVDD -0.5V		AVDD +0.5V	V
Output resistance				300		kΩ
Output capacitance				5		pF
<b>REFERENCE OUTPUT</b>						
$V_{REF}$	Reference output voltage		1.14	1.2	1.26	V
	Reference output current <sup>(3)</sup>			100		nA
<b>REFERENCE INPUT</b>						
$V_{EXTIO}$	Input voltage range	External Reference Mode	0.1	1.2	1.25	V
	Input resistance				1	
	Small signal bandwidth			472		kHz
	Input capacitance			100		pF
<b>TEMPERATURE COEFFICIENTS</b>						
Offset drift				±1		ppm of FSR/°C
Gain drift		With external reference		±15		ppm of FSR/°C
		With internal reference		±30		
Reference voltage drift				±8		ppm/°C

(1) Measured differential across IOUTA1 and IOUTA2 or IOUTB1 and IOUTB2 with 25 Ω each to AVDD.

(2) The lower limit of the output compliance is determined by the CMOS process. Exceeding this limit may result in transistor breakdown, resulting in reduced reliability of the DAC3282 device. The upper limit of the output compliance is determined by the load resistors and full-scale output current. Exceeding the upper limit adversely affects distortion performance and integral nonlinearity.

(3) Use an external buffer amplifier with high impedance input to drive any external load.

**ELECTRICAL CHARACTERISTICS – DC SPECIFICATIONS<sup>(1)</sup> (continued)**

 over recommended operating free-air temperature range, nominal supplies, IOUT<sub>FS</sub> = 20 mA (unless otherwise noted)

PARAMETER		TEST CONDITIONS	MIN	TYP	MAX	UNIT
<b>POWER SUPPLY</b>						
	AVDD33		3.0	3.3	3.6	V
	DACVDD18, DIGVDD18, CLKVDD18		1.7	1.8	1.9	V
I <sub>(AVDD33)</sub>	Analog supply current	Mode 1 (below)		96		mA
I <sub>(DIGVDD18)</sub>	Digital supply current			268		mA
I <sub>(DACVDD18)</sub>	DAC supply current			74		mA
I <sub>(CLKVDD18)</sub>	Clock supply current			10		mA
I <sub>(AVDD33)</sub>	Power down mode analog supply current	Mode 4 (below)		2		mA
I <sub>(DIGVDD18)</sub>	Power down mode digital supply current			3		mA
I <sub>(DACVDD18)</sub>	Power down mode DAC supply current			0.5		mA
I <sub>(CLKVDD18)</sub>	Power down mode clock supply current			1		mA
P	Power Dissipation	Mode 1: f <sub>DAC</sub> = 625MSPS, 2x interpolation, mixer on, Digital Offset Control on		950	1100	mW
		Mode 2: f <sub>DAC</sub> = 491.52MSPS, 2x interpolation, Zero-IF Correction Filter on, mixer off, Digital Offset Control on		845		mW
		Mode 3: Sleep Mode, f <sub>DAC</sub> = 625MSPS, 2X interpolation, mixer on, DAC in sleep mode: CONFIG24 <b>sleepa</b> , <b>sleepb</b> set to 1		575		mW
		Mode 4: Power-Down mode, No clock, static data pattern, DAC in power-down mode: CONFIG23 <b>clkpath_sleep_a</b> , <b>clkpath_sleepb</b> set to 1 CONFIG24 <b>clkrecv_sleep</b> , <b>sleepa</b> , <b>sleepb</b> set to 1		15		mW
PSRR	Power Supply Rejection Ratio	DC tested	-0.4		0.4	%/FSR/V
T	Operating Range		-40	25	85	°C

## ELECTRICAL CHARACTERISTICS – AC SPECIFICATIONS

over recommended operating free-air temperature range, nominal supplies, IOUTFS = 20 mA (unless otherwise noted)

PARAMETER		TEST CONDITIONS	MIN	TYP	MAX	UNIT
<b>ANALOG OUTPUT <sup>(1)</sup></b>						
f <sub>DAC</sub>	Maximum output update rate	1x Interpolation	312.5			MSPS
		2x Interpolation	625			
t <sub>s(DAC)</sub>	Output settling time to 0.1%	Transition: Code 0x0000 to 0xFFFF	10.4			ns
t <sub>pd</sub>	Output propagation delay	DAC outputs are updated on the falling edge of DAC clock. Does not include Digital Latency (see below).	2			ns
t <sub>r(IOUT)</sub>	Output rise time 10% to 90%		220			ps
t <sub>f(IOUT)</sub>	Output fall time 90% to 10%		220			PS
Power-up time	DAC Wake-up Time	IOUT current settling to 1% of IOUT <sub>FS</sub> . Measured from SDENB rising edge; Register CONFIG24, toggle <b>sleepa</b> from 1 to 0	90			μs
	DAC Sleep Time	IOUT current settling to less than 1% of IOUT <sub>FS</sub> . Measured from SDENB rising edge; Register CONFIG24, toggle <b>sleepa</b> from 0 to 1.	90			μs
Digital Latency	No interpolation, FIFO off, Offset off, Inverse sinc off		38			DAC clock cycles
	2x Interpolation		59			
	Zero-IF Sinc Correction Filter		16			
	FIFO		8			
	Offset		4			
<b>AC PERFORMANCE <sup>(2)</sup></b>						
SFDR	Spurious Free Dynamic Range SFDR (0 to f <sub>DAC</sub> /2) Tone at 0 dBFS	f <sub>DAC</sub> = 625 MSPS, f <sub>OUT</sub> = 10.1 MHz 2x Interp, DAC A+B on	83			dBc
		f <sub>DAC</sub> = 625 MSPS, f <sub>OUT</sub> = 20.1 MHz 2x Interp, DAC A+B on	78			
		f <sub>DAC</sub> = 625 MSPS, f <sub>OUT</sub> = 70.1 MHz 2x Interp, DAC A+B on	64			
IMD3	Third-order two-tone intermodulation distortion Each tone at -6 dBFS	f <sub>DAC</sub> = 625 MSPS, f <sub>OUT</sub> = 30 ± 0.5 MHz 2x Interp, DAC A+B on	82			dBc
		f <sub>DAC</sub> = 625 MSPS, f <sub>OUT</sub> = 50 ± 0.5 MHz 2x Interp, DAC A+B on	80			
		f <sub>DAC</sub> = 625 MSPS, f <sub>OUT</sub> = 150 ± 0.5 MHz 2x Interp, DAC A+B on,	69			
NSD	Noise Spectral Density Single Tone at 0 dBm	f <sub>DAC</sub> = 625 MSPS, f <sub>OUT</sub> = 10.1 MHz 2x Interp, DAC A+B on	161			dBc/Hz
		f <sub>DAC</sub> = 625 MSPS, f <sub>OUT</sub> = 150.1 MHz 2x Interp, DAC A+B on	150			
WCDMA <sup>(3)</sup>	Adjacent Channel Leakage Ratio, Single Carrier	f <sub>DAC</sub> = 491.52 MSPS, f <sub>OUT</sub> = 30.72 MHz 2x Interp, DAC A+B on	81			dBc
		f <sub>DAC</sub> = 491.52 MSPS, f <sub>OUT</sub> = 153.6 MHz 2x Interp, DAC A+B on	76			
	Alternate Channel Leakage Ratio, Single Carrier	f <sub>DAC</sub> = 491.52 MSPS, f <sub>OUT</sub> = 30.72 MHz 2x Interp, DAC A+B on	84			dBc
		f <sub>DAC</sub> = 491.52 MSPS, f <sub>OUT</sub> = 153.6 MHz 2x Interp, DAC A+B on	77			
	Channel Isolation	f <sub>DAC</sub> = 625 MSPS, f <sub>OUT</sub> = 10 MHz	84			dBc

(1) Measured single ended into 50 Ω load.

(2) 4:1 transformer output termination, 50 Ω doubly terminated load.

(3) Single carrier, W-CDMA with 3.84 MHz BW, 5-MHz spacing, centered at IF, PAR = 12dB. TESTMODEL 1, 10 ms



## ELECTRICAL CHARACTERISTICS – DIGITAL SPECIFICATIONS

 over recommended operating free-air temperature range, nominal supplies,  $I_{OUT_{FS}} = 20 \text{ mA}$  (unless otherwise noted)

PARAMETER		TEST CONDITIONS	MIN	TYP	MAX	UNIT
<b>LVDS INTERFACE: D[7:0]P/N, DATACLKP/N, FRAMEP/N<sup>(1)</sup></b>						
$f_{DATA}$	Input data rate	Byte-wide DDR format DATACLK frequency = 625 MHz			312.5	MSPS
$f_{BUS}$	Byte-wide LVDS data transfer rate	1x Interpolation			1250	MSPS
		2x Interpolation			1250	
$V_{A,B+}$	Logic high differential input voltage threshold		175	400		mV
$V_{A,B-}$	Logic low differential input voltage threshold		-175	-400		mV
$V_{COM}$	Input Common Mode		1.0	1.2	2.0	V
$Z_T$	Internal termination		85	110	135	$\Omega$
$C_L$	LVDS Input capacitance			2		pF
<b>TIMING LVDS INPUTS: DATACLKP/N, double edge latching – See Figure 36</b>						
$t_{s(DATA)}$	Setup time, D[7:0]P/N and FRAMEP/N, valid to either edge of DATACLKP/N	FRAMEP/N latched on rising edge of DATACLKP/N only	0			ps
$t_{h(DATA)}$	Hold time, D[7:0]P/N and FRAMEP/N, valid after either edge of DATACLKP/N	FRAMEP/N latched on rising edge of DATACLKP/N only	400			ps
$t_{(FRAME)}$	FRAMEP/N pulse width	$f_{DATACLK}$ is DATACLK frequency in MHz	$1/2f_{DATACLK}$			ns
$t_{align}$	Maximum offset between DATACLKP/N and DACCLKP/N rising edges	FIFO Bypass Mode only $f_{DACCLK}$ is DACCLK frequency in MHz			$1/2f_{DACCLK}$ -0.55	ns
<b>CLOCK INPUT (DACCLKP/N)</b>						
	Duty cycle		40%		60%	
	Differential voltage <sup>(2)</sup>		0.4	1.0		V
	DACCLKP/N Input Frequency				625	MHz
<b>OUTPUT STROBE (OSTRP/N)</b>						
$f_{OSTR}$	Frequency	$f_{OSTR} = f_{DACCLK} / (n \times 8 \times \text{Interp})$ where n is any positive integer $f_{DACCLK}$ is DACCLK frequency in MHz			$f_{DACCLK} / (8 \times \text{interp})$	MHz
	Duty cycle		40%		60%	
	Differential voltage		0.4	1.0		V
<b>TIMING OSTRP/N Input: DACCLKP/N rising edge latching</b>						
$t_{s(OSTR)}$	Setup time, OSTRP/N valid to rising edge of DACCLKP/N			200		ps
$t_{h(OSTR)}$	Hold time, OSTRP/N valid after rising edge of DACCLKP/N			200		ps
<b>CMOS INTERFACE: ALARM, SDO, SDIO, SCLK, SDENB, RESETB, TXENABLE</b>						
$V_{IH}$	High-level input voltage		1.25			V
$V_{IL}$	Low-level input voltage				0.54	V
$I_{IH}$	High-level input current		-40		40	$\mu\text{A}$
$I_{IL}$	Low-level input current		-40		40	$\mu\text{A}$
$C_I$	CMOS Input capacitance			2		pF
$V_{OH}$	SDO, SDIO	Iload = -100 $\mu\text{A}$			DIGVDD18 -0.2	V
	SDO, SDIO	Iload = -2 mA			0.8 x DIGVDD18	V
$V_{OL}$	SDO, SDIO	Iload = 100 $\mu\text{A}$			0.2	V
	SDO, SDIO	Iload = 2 mA			0.5	V

(1) See LVDS INPUTS section for terminology.

(2) Driving the clock input with a differential voltage lower than 1 V will result in degraded performance.

## ELECTRICAL CHARACTERISTICS – DIGITAL SPECIFICATIONS (continued)

over recommended operating free-air temperature range, nominal supplies,  $I_{OUT_{FS}} = 20 \text{ mA}$  (unless otherwise noted)

PARAMETER		TEST CONDITIONS	MIN	TYP	MAX	UNIT
<b>SERIAL PORT TIMING – See Figure 32 and Figure 33</b>						
$t_{s(SDENB)}$	Setup time, SDENB to rising edge of SCLK		20			ns
$t_{s(SDIO)}$	Setup time, SDIO valid to rising edge of SCLK		10			ns
$t_{h(SDIO)}$	Hold time, SDIO valid to rising edge of SCLK		5			ns
$t_{(SCLK)}$	Period of SCLK	Register CONFIG5 read (temperature sensor read)	1			$\mu\text{s}$
		All other registers	100			ns
$t_{(SCLKH)}$	High time of SCLK	Register CONFIG5 read (temperature sensor read)	0.4			$\mu\text{s}$
		All other registers	40			ns
$t_{(SCLKL)}$	Low time of SCLK	Register CONFIG5 read (temperature sensor read)	0.4			$\mu\text{s}$
		All other registers	40			ns
$t_{d(Data)}$	Data output delay after falling edge of SCLK			10		ns
$t_{RESET}$	Minimum RESETB pulsewidth			25		ns

TYPICAL CHARACTERISTICS

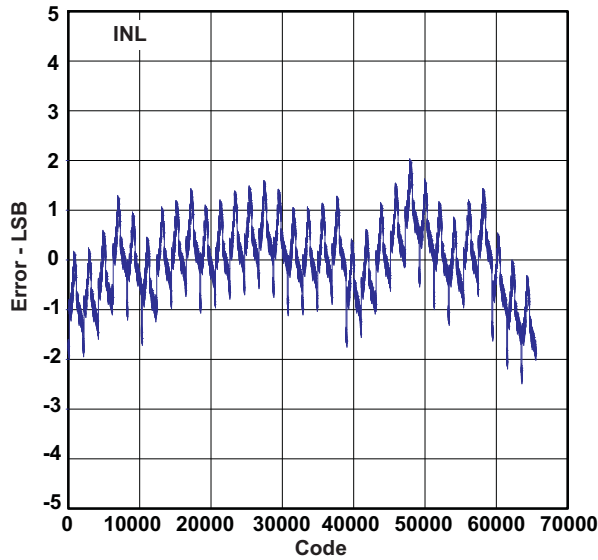


Figure 1. Integral Non-Linearity

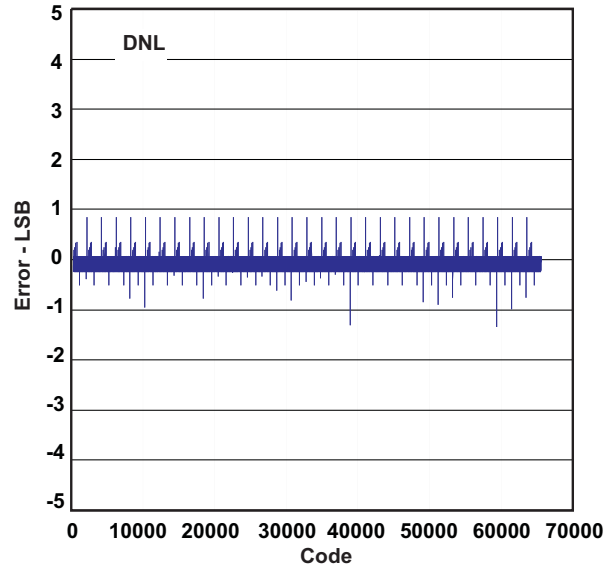


Figure 2. Differential Non-Linearity

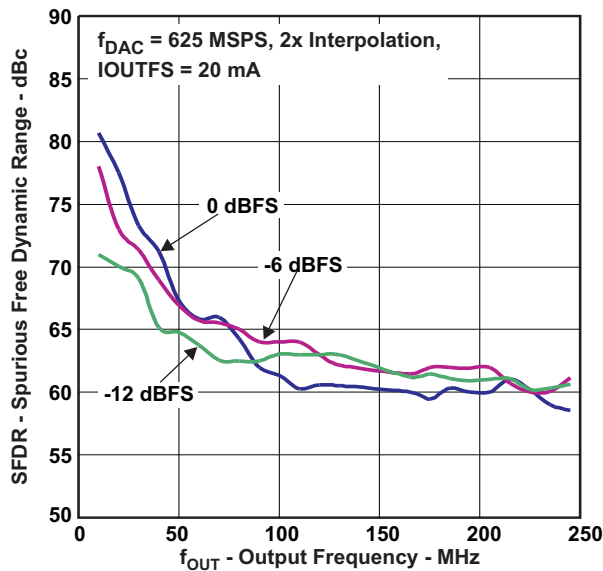


Figure 3. SFDR vs Input Scale

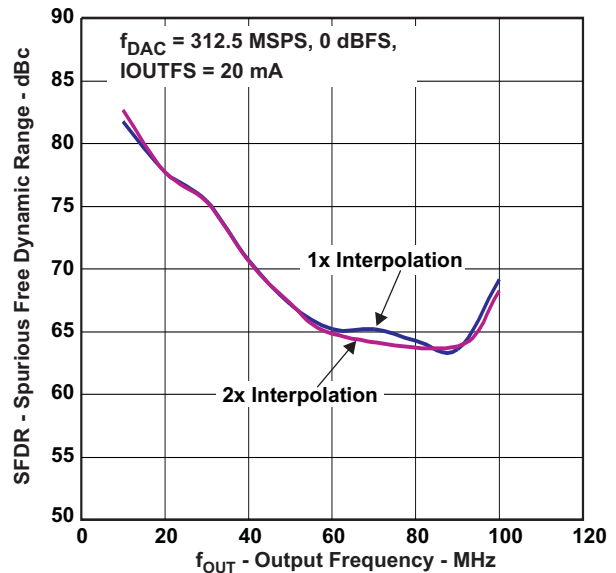


Figure 4. SFDR vs Interpolation

TYPICAL CHARACTERISTICS (continued)

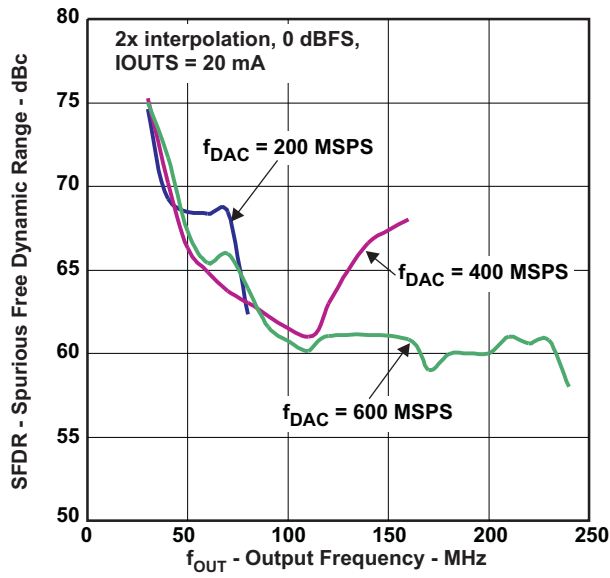


Figure 5. SFDR vs  $f_{DAC}$

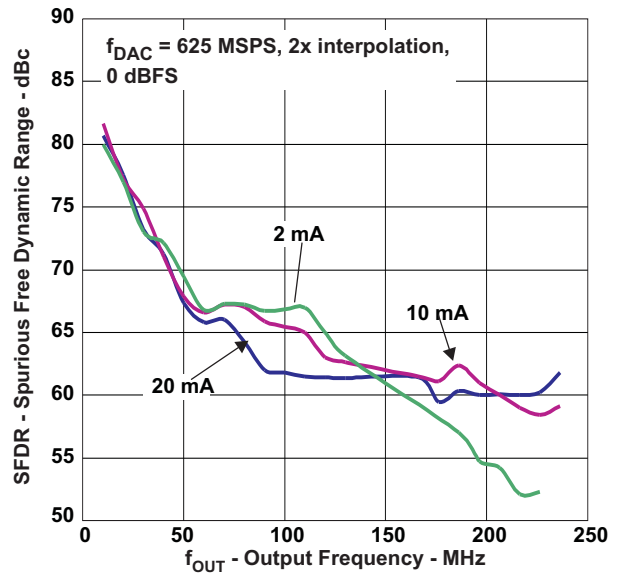


Figure 6. SFDR vs  $I_{OUTS}$

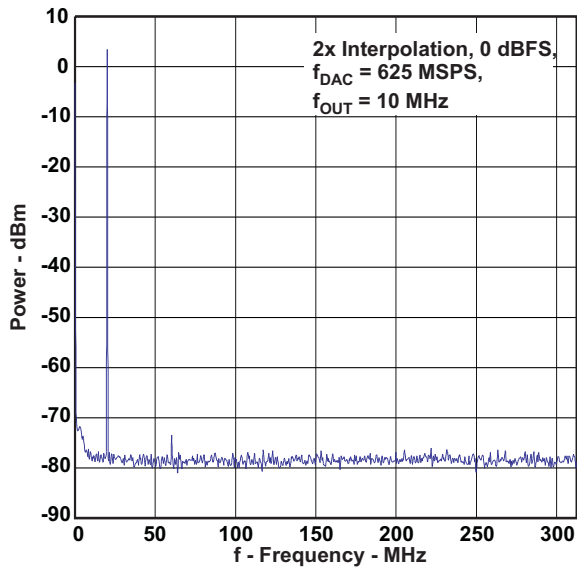


Figure 7. Single Tone Spectral Plot

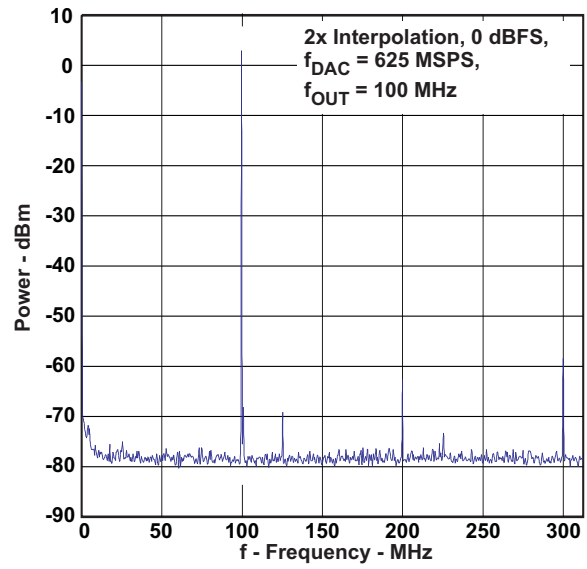


Figure 8. Single Tone Spectral Plot

TYPICAL CHARACTERISTICS (continued)

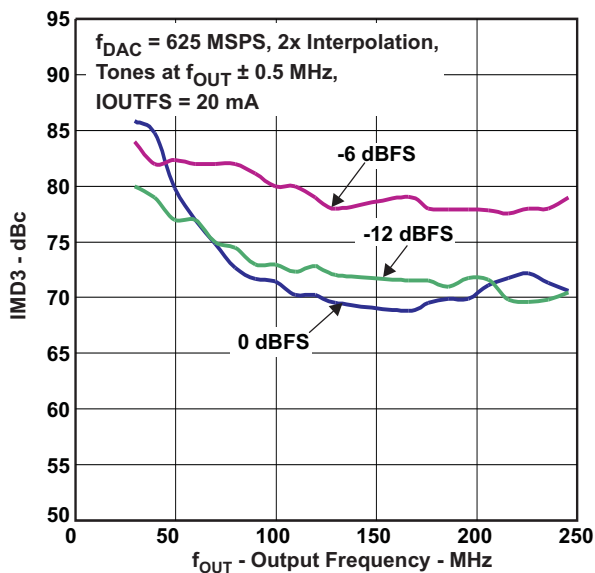


Figure 9. IMD3 vs Input Scale

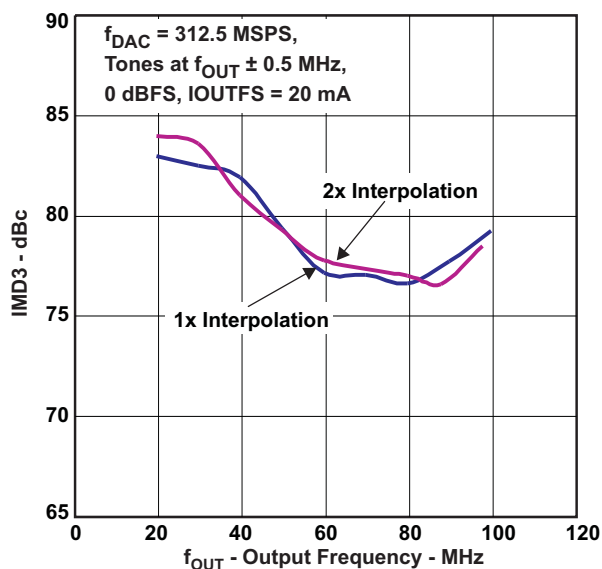


Figure 10. IMD3 vs Interpolation

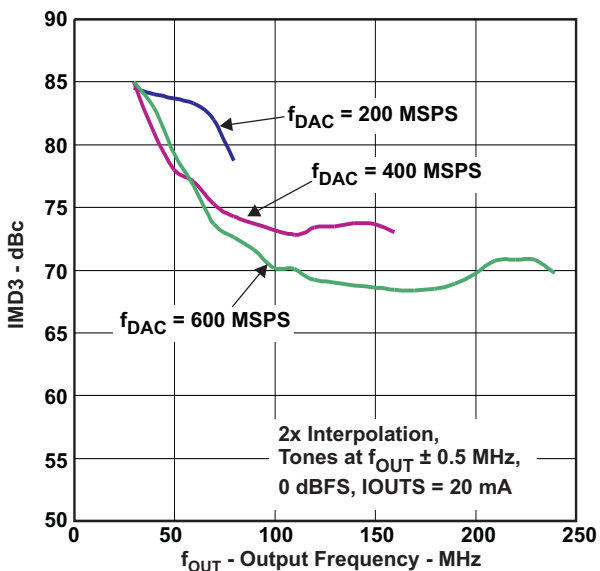


Figure 11. IMD3 vs  $f_{DAC}$

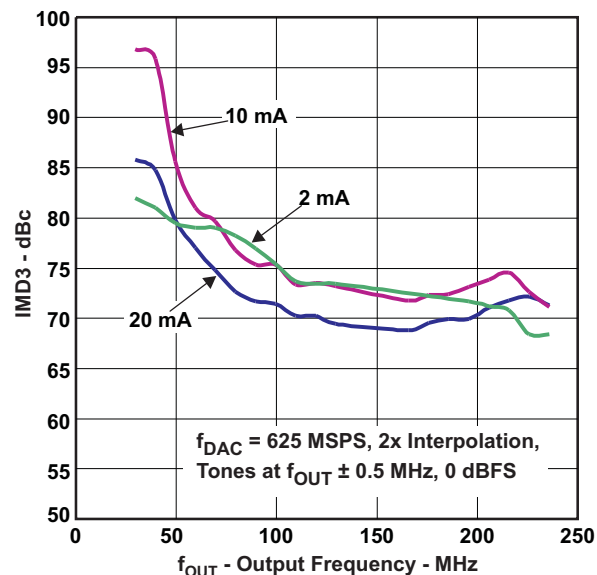


Figure 12. IMD3 vs  $I_{OUTFS}$

TYPICAL CHARACTERISTICS (continued)

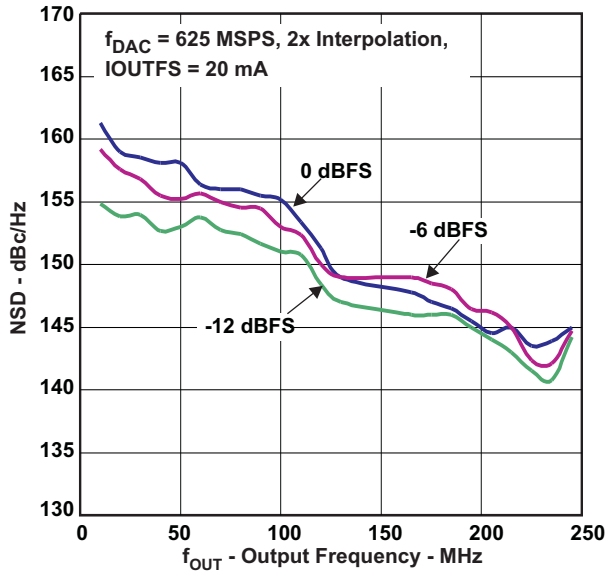


Figure 13. NSD vs Input Scale

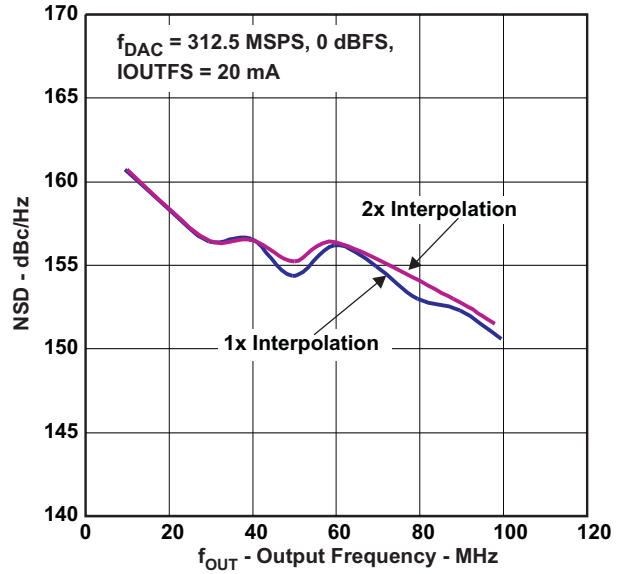


Figure 14. NSD vs Interpolation

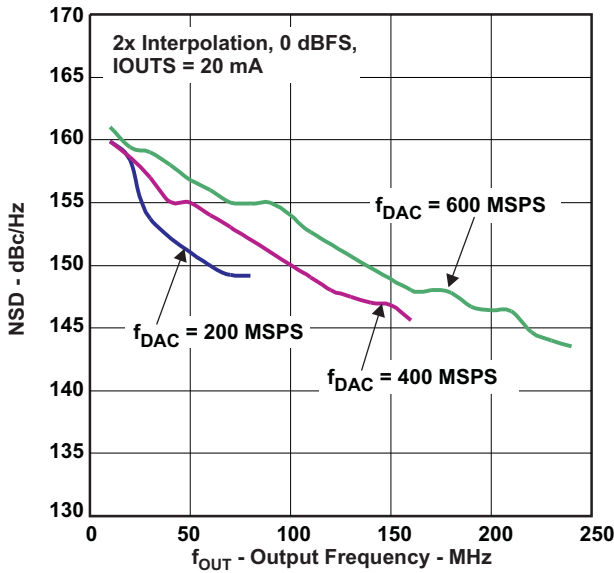


Figure 15. NSD vs  $f_{DAC}$

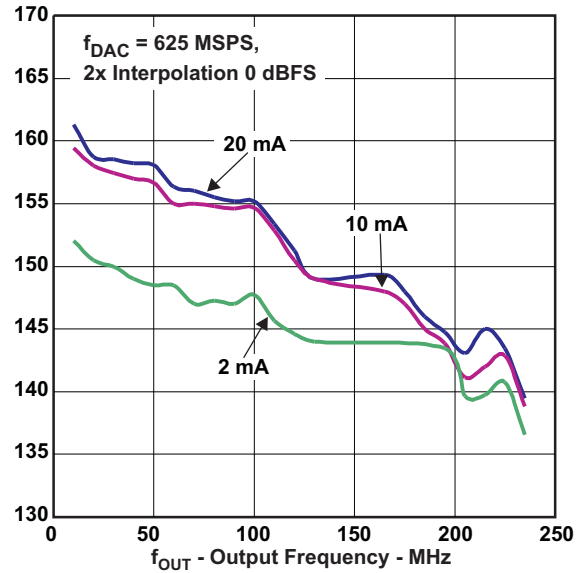


Figure 16. NSD vs  $I_{OUTFS}$

TYPICAL CHARACTERISTICS (continued)

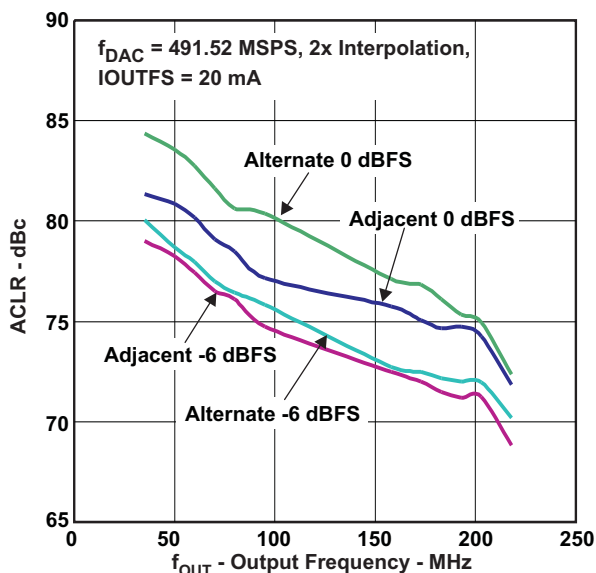


Figure 17. Single Carrier WCDMA ACLR vs Input Scale

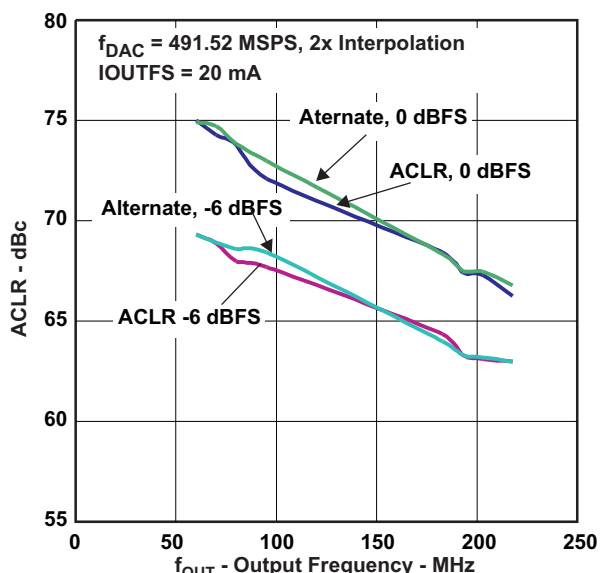


Figure 18. Four Carrier WCDMA ACLR vs Input Scale

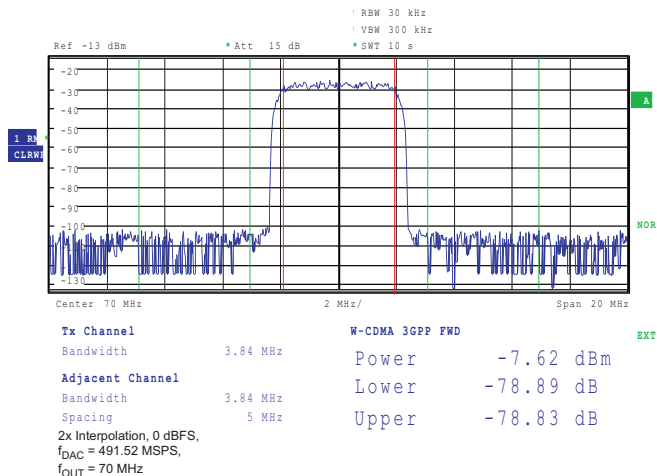


Figure 19. Single Carrier W-CDMA Test Model 1,  $f_{OUT} = 70$  MHz

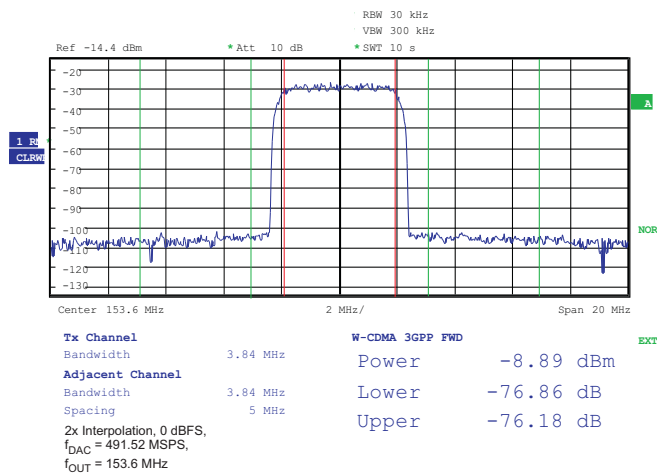


Figure 20. Single Carrier W-CDMA Test Model 1,  $f_{OUT} = 153.6$  MHz

TYPICAL CHARACTERISTICS (continued)

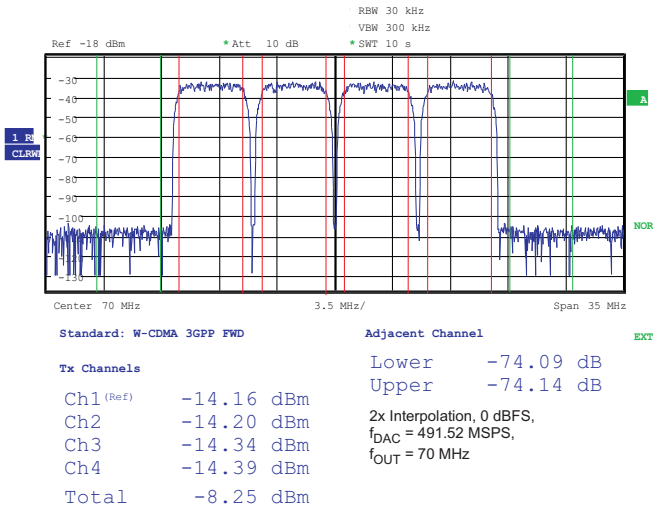


Figure 21. Four Carrier W-CDMA Test Model 1, f<sub>OUT</sub> = 70 MHz

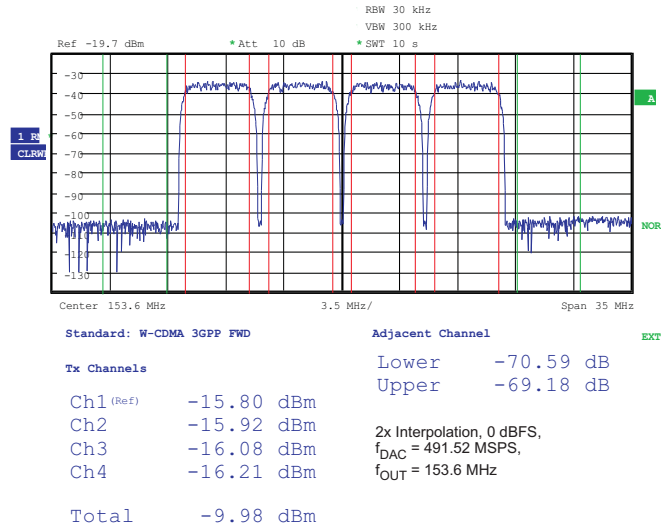


Figure 22. Four Carrier W-CDMA Test Model 1, f<sub>OUT</sub> = 153.6 MHz

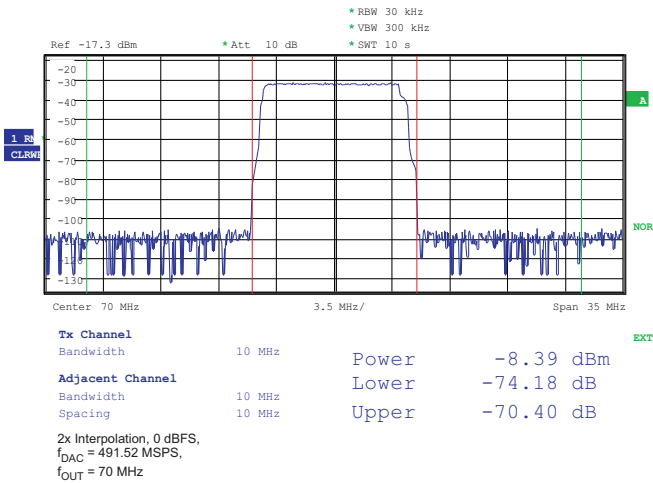


Figure 23. 10MHz Single Carrier LTE, f<sub>OUT</sub> = 70 MHz

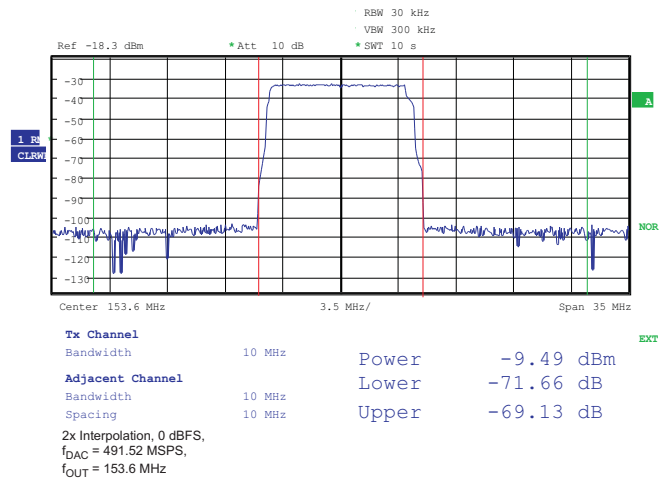


Figure 24. 10MHz Single Carrier LTE, f<sub>OUT</sub> = 153.6 MHz



TYPICAL CHARACTERISTICS (continued)

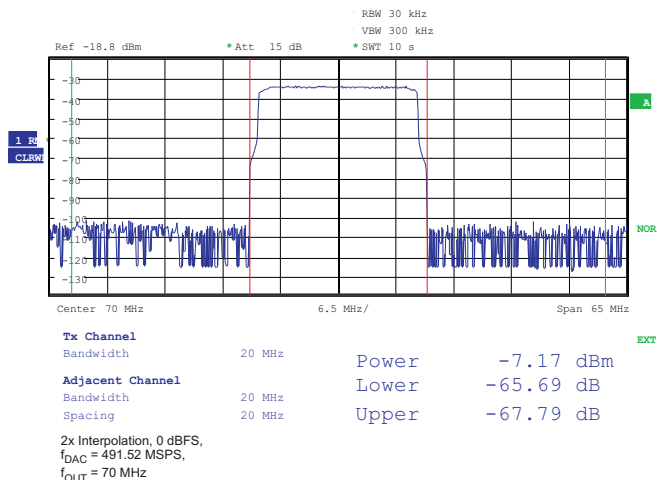


Figure 25. 20MHz Single Carrier LTE,  $f_{OUT} = 70$  MHz

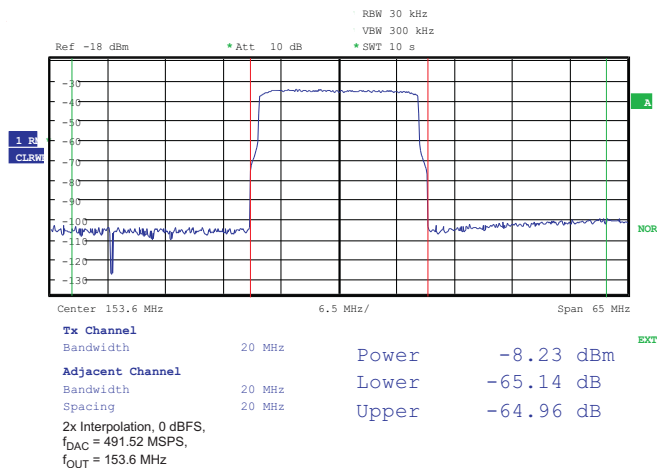


Figure 26. 20MHz Single Carrier LTE,  $f_{OUT} = 153.6$  MHz

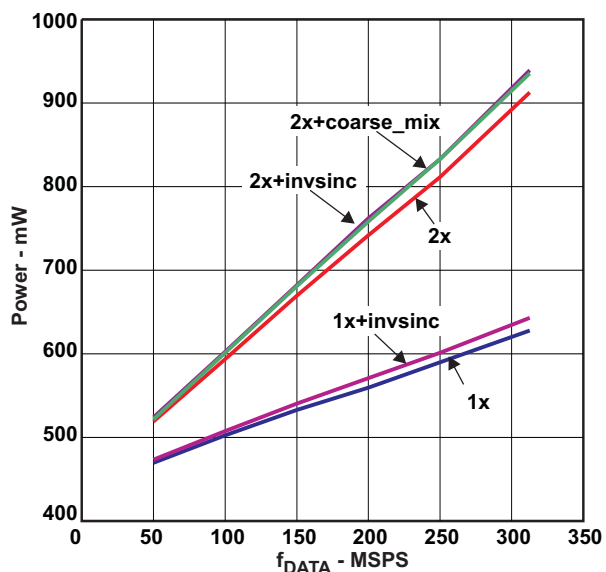


Figure 27. Power vs  $f_{DATA}$

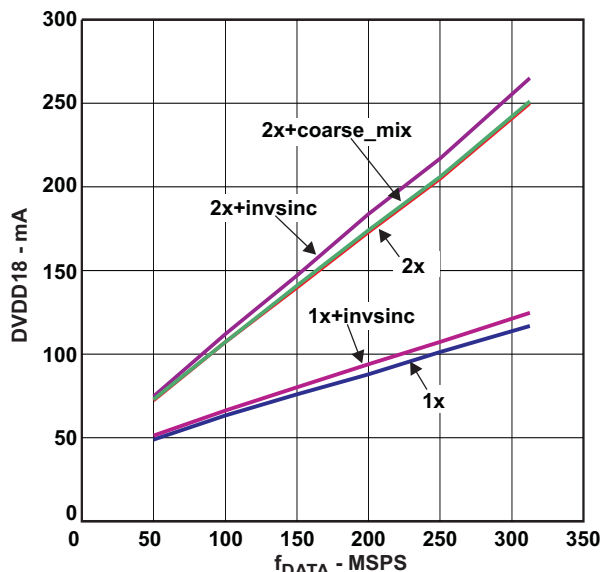


Figure 28. DVDD18 vs  $f_{DATA}$

TYPICAL CHARACTERISTICS (continued)

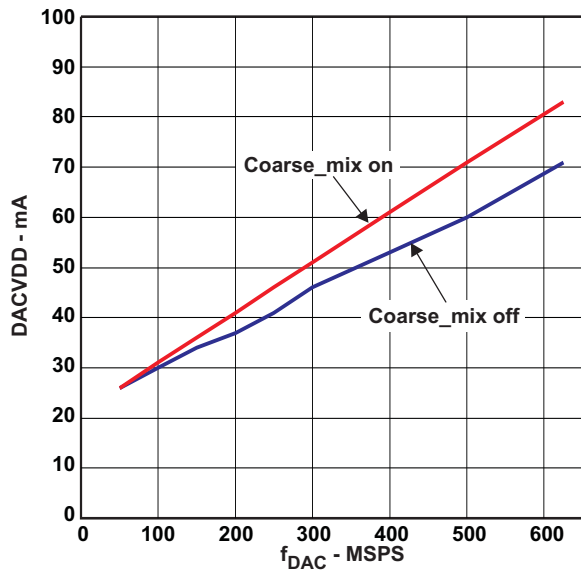


Figure 29. DACVDD18 vs f<sub>DAC</sub>

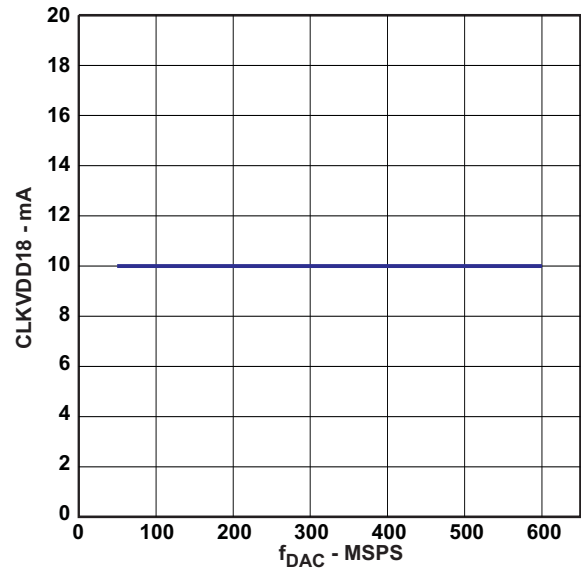


Figure 30. CLKVDD18 vs f<sub>DAC</sub>

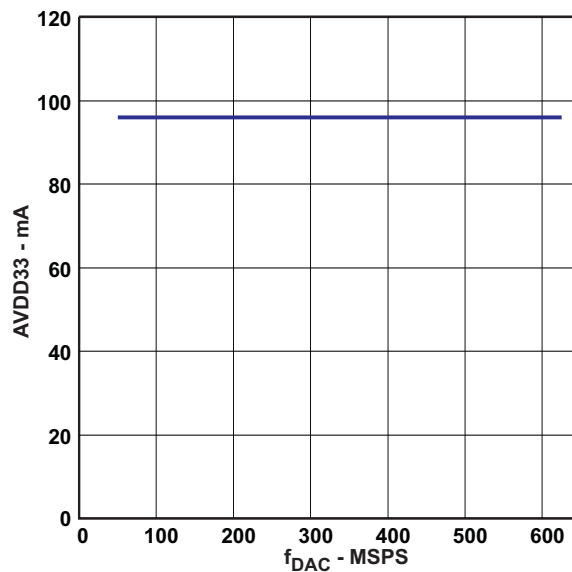


Figure 31. AVDD33 vs f<sub>DAC</sub>

## DEFINITION OF SPECIFICATIONS

**Adjacent Carrier Leakage Ratio (ACLR):** Defined for a 3.84Mcps 3GPP W-CDMA input signal measured in a 3.84MHz bandwidth at a 5MHz offset from the carrier with a 12dB peak-to-average ratio.

**Analog and Digital Power Supply Rejection Ratio (APSSR, DPSSR):** Defined as the percentage error in the ratio of the delta IOUT and delta supply voltage normalized with respect to the ideal IOUT current.

**Differential Nonlinearity (DNL):** Defined as the variation in analog output associated with an ideal 1 LSB change in the digital input code.

**Gain Drift:** Defined as the maximum change in gain, in terms of ppm of full-scale range (FSR) per °C, from the value at ambient (25°C) to values over the full operating temperature range.

**Gain Error:** Defined as the percentage error (in FSR%) for the ratio between the measured full-scale output current and the ideal full-scale output current.

**Integral Nonlinearity (INL):** Defined as the maximum deviation of the actual analog output from the ideal output, determined by a straight line drawn from zero scale to full scale.

**Intermodulation Distortion (IMD3):** The two-tone IMD3 is defined as the ratio (in dBc) of the 3rd-order intermodulation distortion product to either fundamental output tone.

**Offset Drift:** Defined as the maximum change in DC offset, in terms of ppm of full-scale range (FSR) per °C, from the value at ambient (25°C) to values over the full operating temperature range.

**Offset Error:** Defined as the percentage error (in FSR%) for the ratio between the measured mid-scale output current and the ideal mid-scale output current.

**Output Compliance Range:** Defined as the minimum and maximum allowable voltage at the output of the current-output DAC. Exceeding this limit may result reduced reliability of the device or adversely affecting distortion performance.

**Reference Voltage Drift:** Defined as the maximum change of the reference voltage in ppm per °C from value at ambient (25°C) to values over the full operating temperature range.

**Spurious Free Dynamic Range (SFDR):** Defined as the difference (in dBc) between the peak amplitude of the output signal and the peak spurious signal.

**Noise Spectral Density (NSD):** Defined as the difference of power (in dBc) between the output tone signal power and the noise floor of 1Hz bandwidth within the first Nyquist zone.

## APPLICATION INFORMATION

### SERIAL INTERFACE

The serial port of the DAC3282 is a flexible serial interface which communicates with industry standard microprocessors and microcontrollers. The interface provides read/write access to all registers used to define the operating modes of DAC3282. It is compatible with most synchronous transfer formats and can be configured as a 3 or 4 pin interface by **sif4\_ena** in register **CONFIG23**. In both configurations, SCLK is the serial interface input clock and SDENB is serial interface enable. For 3 pin configuration, SDIO is a bidirectional pin for both data in and data out. For 4 pin configuration, SDIO is bidirectional and ALARM\_SDO is data out only. Data is input into the device with the rising edge of SCLK. Data is output from the device on the falling edge of SCLK.

Each read/write operation is framed by signal SDENB (Serial Data Enable Bar) asserted low for 2 to 5 bytes, depending on the data length to be transferred (1–4 bytes). The first frame byte is the instruction cycle which identifies the following data transfer cycle as read or write, how many bytes to transfer, and what address to transfer the data. [Table 1](#) indicates the function of each bit in the instruction cycle and is followed by a detailed description of each bit. Frame bytes 2 to 5 comprise the data transfer cycle.

**Table 1. Instruction Byte of the Serial Interface**

	MSB							LSB
Bit	7	6	5	4	3	2	1	0
Description	R/W	N1	N0	A4	A3	A2	A1	A0

**R/W** Identifies the following data transfer cycle as a read or write operation. A high indicates a read operation from DAC3282 and a low indicates a write operation to DAC3282.

**[N1 : N0]** Identifies the number of data bytes to be transferred per [Table 2](#). Data is transferred MSB first.

**Table 2. Number of Transferred Bytes Within One Communication Frame**

N1	N0	Description
0	0	Transfer 1 Byte
0	1	Transfer 2 Bytes
1	0	Transfer 3 Bytes
1	1	Transfer 4 Bytes

**[A4 : A0]** Identifies the address of the register to be accessed during the read or write operation. For multi-byte transfers, this address is the starting address. Note that the address is written to the DAC3282 MSB first and counts down for each byte.

[Figure 32](#) shows the serial interface timing diagram for a DAC3282 write operation. SCLK is the serial interface clock input to DAC3282. Serial data enable SDENB is an active low input to DAC3282. SDIO is serial data in. Input data to DAC3282 is clocked on the rising edges of SCLK.

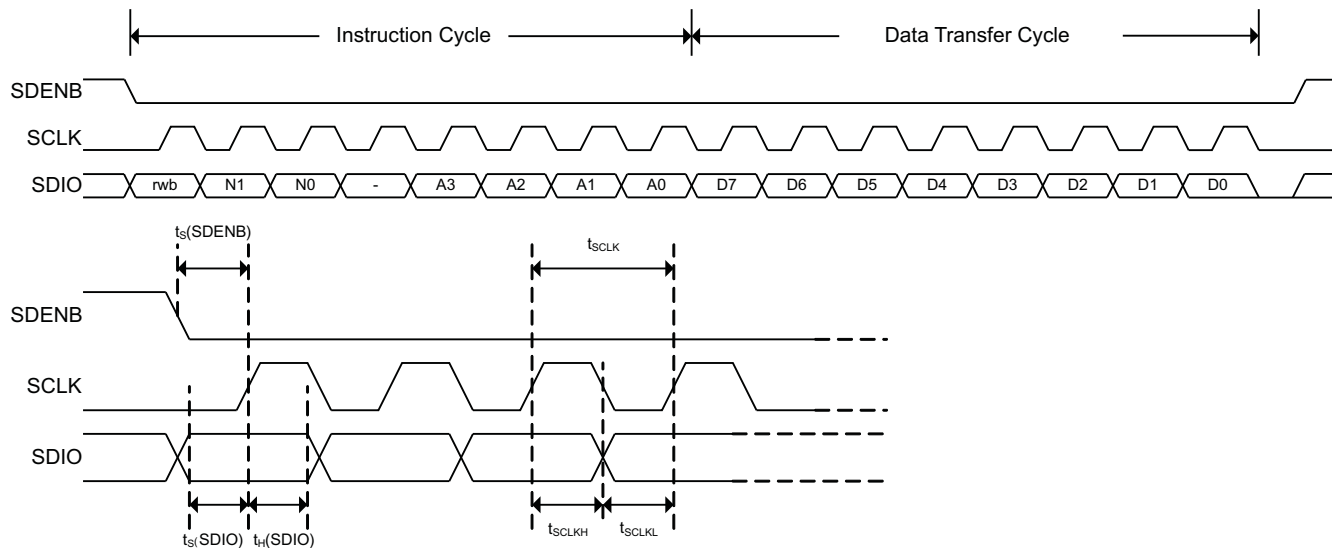


Figure 32. Serial Interface Write Timing Diagram

Figure 33 shows the serial interface timing diagram for a DAC3282 read operation. SCLK is the serial interface clock input to DAC3282. Serial data enable SDENB is an active low input to DAC3282. SDIO is serial data in during the instruction cycle. In 3 pin configuration, SDIO is data out from DAC3282 during the data transfer cycle(s), while ALARM\_SDO is in a high-impedance state. In 4 pin configuration, both ALARM\_SDO and SDIO are data out from DAC3282. At the end of the data transfer, ALARM\_SDO will output low on the final falling edge of SCLK until the rising edge of SDENB when it will 3-state.

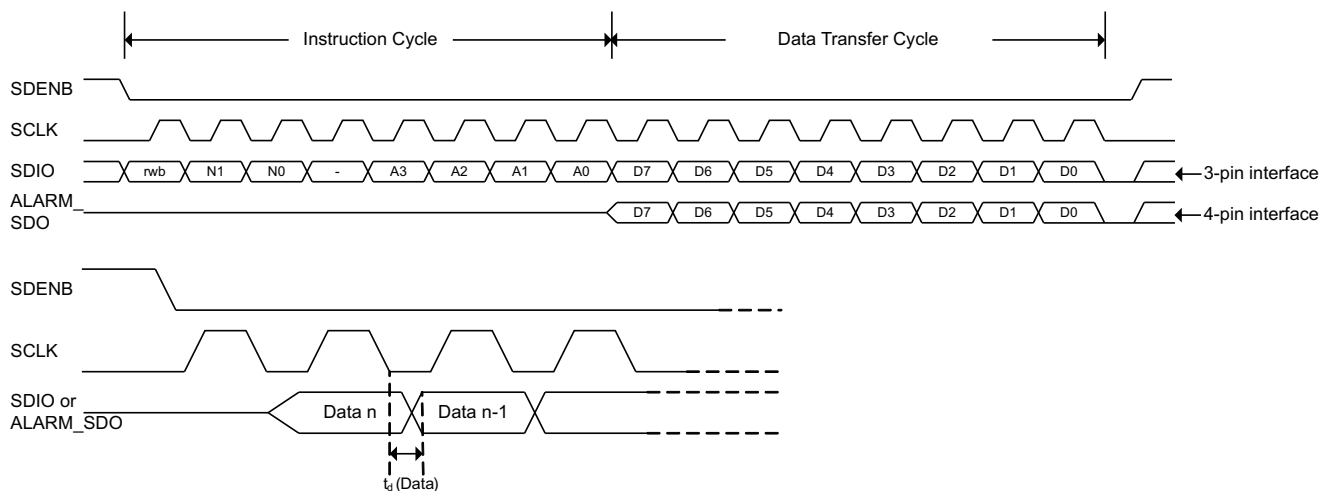


Figure 33. Serial Interface Read Timing Diagram

## REGISTER DESCRIPTIONS

### Register Map

Table 3. Register Map

Name	Address	Default	(MSB) Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	(LSB) Bit 0
CONFIG0	0x00	0x70	qmc_offset_ena	fifo_ena	fifo_reset_ena	multi_sync_ena	alarm_out_ena	alarm_pol	mixer_func(1:0)	
CONFIG1	0x01	0x11	unused	unused	unused	fir_ena	fir4_ena	iotest_ena	unused	twos
CONFIG2	0x02	0x00	unused	unused	unused	unused	output_delay(3:0)			
CONFIG3	0x03	0x10	64cnt_ena	unused	unused	fifo_offset(2:0)			alarm_2away_ena	alarm_1away_ena
CONFIG4	0x04	0xFF	coarse_daca(3:0)				coarse_dacb(3:0)			
CONFIG5	0x05	N/A	tempdata(7:0)							
CONFIG6	0x06	0x00	unused	alarm_mask(6:0)						
CONFIG7	0x07	0x00	unused	alarm_from_zerock	alarm_fifo_collision	reserved	alarm_from_iotest	unused	alarm_fifo_2away	alarm_fifo_1away
CONFIG8	0x08	0x00	iotest_results(7:0)							
CONFIG9	0x09	0x7A	iotest_pattern0(7:0)							
CONFIG10	0x0A	0xB6	iotest_pattern1(7:0)							
CONFIG11	0x0B	0xEA	iotest_pattern2(7:0)							
CONFIG12	0x0C	0x45	iotest_pattern3(7:0)							
CONFIG13	0x0D	0x1A	iotest_pattern4(7:0)							
CONFIG14	0x0E	0x16	iotest_pattern5(7:0)							
CONFIG15	0x0F	0xAA	iotest_pattern6(7:0)							
CONFIG16	0x10	0xC6	iotest_pattern7(7:0)							
CONFIG17	0x11	0x00	reserved	reserved	reserved	reserved				
CONFIG18	0x12	0x02	reserved			reserved	daca_complement	dacb_complement	clkdiv_sync_ena	unused
CONFIG19	0x13	0x00	bequalsa	aequalsb	reserved	unused	unused	unused	multi_sync_sel	rev
CONFIG20	0x14	0x00	qmc_offseta(7:0)							
CONFIG21	0x15	0x00	qmc_offsetb(7:0)							
CONFIG22	0x16	0x00	qmc_offseta(12:8)					unused	unused	unused
CONFIG23	0x17	0x00	qmc_offsetb(12:8)					sif4_ena	clkpath_sleep_a	clkpath_sleep_b
CONFIG24	0x18	0x83	tsense_ena	clkrecv_sleep	unused	reserved	sleepb	sleepa	reserved	reserved
CONFIG25	0x19	0x00	reserved					extref_ena	reserved	reserved
CONFIG26	0x1A	0x00	unused	unused	unused	unused	unused	reserved		
CONFIG27	0x1B	0x00	reserved							
CONFIG28	0x1C	0x00	reserved							
CONFIG29	0x1D	0x00	reserved							
CONFIG30	0x1E	0x00	reserved							
VERSION31	0x1F	0x43	deviceid(1:0)			version(5:0)				

**Register name: CONFIG0 – Address: 0x00, Default: 0x70**

Register Name	Address	Bit	Name	Function	Default Value
CONFIG0	0x00	7	qmc_offset_ena	When asserted the DAC offset correction is enabled.	0
		6	fifoin_ena	When asserted the FIFO is enabled. <b>When the FIFO is bypassed DACCLKP/N and DATACLKP/N must be aligned to within t<sub>align</sub>.</b>	1
		5	fifo_reset_ena	Allows the FRAME input to reset the FIFO write pointer when asserted.	1
		4	multi_sync_ena	Allows the FRAME or OSTR signal to reset the FIFO read pointer when asserted. This selection is determined by multi_sync_sel in register CONFIG19.	1
		3	alarm_out_ena	When asserted the ALARM_SDO pin becomes an output. The functionality of this pin is controlled by the CONFIG6 alarm_mask setting.	0
		2	alarm_pol	This bit changes the polarity of the ALARM signal. (0=negative logic, 1=positive)	0
		1:0	mixer_func(1:0)	Controls the function of the mixer block.	00
		<b>Mode</b>	<b>mixer_func(1:0)</b>		
		Normal	00		
		High Pass(Fs/2)	01		
		Fs/4	10		
		-Fs/4	11		

**Register name: CONFIG1 – Address: 0x01, Default: 0x11**

Register Name	Address	Bit	Name	Function	Default Value
CONFIG1	0x01	7	Unused	Reserved for factory use.	0
		6	Unused	Reserved for factory use.	0
		5	Unused	Reserved for factory use.	0
		4	fir_ena	When asserted the chip does 2X interpolation of the data.	1
		3	fir4_ena	When asserted, the zero-IF sinc correction filter is enabled. This filter cannot be used unless fir_ena is asserted.	0
		2	iotest_ena	When asserted enables the data pattern checker operation.	0
		1	Unused	Reserved for factory use.	0
		0	twos	When asserted the inputs are expected to be in 2's complement format. When de-asserted the input format is expected to be offset-binary.	1

**Register name: CONFIG2 – Address: 0x02, Default: 0x00**

Register Name	Address	Bit	Name	Function	Default Value
CONFIG2	0x02	7	Unused	Reserved for factory use.	0
		6	Unused	Reserved for factory use.	0
		5	Unused	Reserved for factory use.	0
		4	Unused	Reserved for factory use.	0
		3:0	output_delay(3:0)	Delays the output to the DACs from 0 to 15 DAC clock cycles.	0000

**Register name: CONFIG3 – Address: 0x03, Default: 0x10**

Register Name	Address	Bit	Name	Function	Default Value
CONFIG3	0x03	7	64cnt_ena	This enables resetting the alarms after 64 good samples with the goal of removing unnecessary errors. For instance, when checking setup/hold through the pattern checker test, there may initially be errors. Setting this bit removes the need for a SIF write to clear the alarm register.	0
		6	Unused	Reserved for factory use.	0
		5	Unused	Reserved for factory use.	0
		4:2	fifo_offset(2:0)	This is the default FIFO read pointer position after the FIFO read pointer has been synced. With this value the initial difference between write and read pointers can be controlled. This may be helpful in controlling the delay through the device.	100
		1	alarm_2away_ena	When asserted alarms from the FIFO that represent the write and read pointers being 2 away are enabled.	0
		0	alarm_1away_ena	When asserted alarms from the FIFO that represent the write and read pointers being 1 away are enabled.	0

**Register name: CONFIG4 – Address: 0x04, Default: 0xFF**

Register Name	Address	Bit	Name	Function	Default Value
CONFIG4	0x04	7:4	coarse_daca(3:0)	Scales the output current in 16 equal steps. $\frac{V_{EXTIO}}{R_{bias}} \times (coarse\_daca/b+1)$	1111
		3:0	coarse_dacb(3:0)	Scales the output current in 16 equal steps.	1111

**Register name: CONFIG5 – Address: 0x05, READ ONLY**

Register Name	Address	Bit	Name	Function	Default Value
CONFIG5	0x05	7:0	tempdata(7:0)	This is the output from the chip temperature sensor. The value of this register in two's complement format represents the temperature in degrees Celsius. <b>This register must be read with a minimum SCLK period of 1µs. (Read Only)</b>	N/A

**Register name: CONFIG6 – Address: 0x06, Default: 0x00**

Register Name	Address	Bit	Name	Function	Default Value																
CONFIG6	0x06	7	Unused	Reserved for factory use.	0																
		6:0	alarm_mask(6:0)	These bits control the masking of the alarm outputs. This means that the ALARM_SDO pin will not be asserted if the appropriate bit is set. The alarm will still show up in the CONFIG7 bits. (0=not masked, 1=masked). <table border="1" style="width: 100%; margin-top: 5px;"> <thead> <tr> <th>alarm_mask</th> <th>Masked Alarm</th> </tr> </thead> <tbody> <tr> <td>6</td> <td>alarm_from_zerochk</td> </tr> <tr> <td>5</td> <td>alarm_fifo_collision</td> </tr> <tr> <td>4</td> <td>reserved</td> </tr> <tr> <td>3</td> <td>alarm_from_iotest</td> </tr> <tr> <td>2</td> <td>not used (expansion)</td> </tr> <tr> <td>1</td> <td>alarm_fifo_2away</td> </tr> <tr> <td>0</td> <td>alarm_fifo_1away</td> </tr> </tbody> </table>	alarm_mask	Masked Alarm	6	alarm_from_zerochk	5	alarm_fifo_collision	4	reserved	3	alarm_from_iotest	2	not used (expansion)	1	alarm_fifo_2away	0	alarm_fifo_1away	0000000
alarm_mask	Masked Alarm																				
6	alarm_from_zerochk																				
5	alarm_fifo_collision																				
4	reserved																				
3	alarm_from_iotest																				
2	not used (expansion)																				
1	alarm_fifo_2away																				
0	alarm_fifo_1away																				



**Register name: CONFIG7 – Address: 0x07, Default: 0x00 (WRITE TO CLEAR)**

Register Name	Address	Bit	Name	Function	Default Value
CONFIG7	0x07	7	Unused	Reserved for factory use.	0
		6	alarm_from_zerochk	This alarm indicates the 8-bit FIFO write pointer address has an all zeros patterns. Due to pointer address being a shift register, this is not a valid address and will cause the write pointer to be stuck until the next sync. This error is typically caused by timing error or improper power start-up sequence. If this alarm is asserted, resynchronization of FIFO is necessary. Refer to the Power-Up Sequence section for more detail.	0
		5	alarm_fifo_collision	Alarm occurs when the FIFO pointers over/under run each other.	0
		4	Reserved	Reserved for factory use.	0
		3	alarm_from_iotest	This is asserted when the input data pattern does not match the pattern in the iotest_pattern registers.	0
		2	Unused	Reserved for factory use.	0
		1	alarm_fifo_2away	Alarm occurs with the read and write pointers of the FIFO are within 2 addresses of each other.	0
		0	alarm_fifo_1away	Alarm occurs with the read and write pointers of the FIFO are within 1 address of each other.	0

**Register name: CONFIG8 – Address: 0x08, Default: 0x00 (WRITE TO CLEAR)**

Register Name	Address	Bit	Name	Function	Default Value
CONFIG8	0x08	7:0	iotest_results(7:0)	The values of these bits tell which bit in the word failed during the pattern checker test.	0x00

**Register name: CONFIG9 – Address: 0x09, Default: 0x7A**

Register Name	Address	Bit	Name	Function	Default Value
CONFIG9	0x09	7:0	iotest_pattern0(7:0)	This is dataword0 in the IO test pattern. It is used with the seven other words to test the input data.	0x7A

**Register name: CONFIG10 – Address: 0x0A, Default: 0xB6**

Register Name	Address	Bit	Name	Function	Default Value
CONFIG10	0x0A	7:0	iotest_pattern1(7:0)	This is dataword1 in the IO test pattern. It is used with the seven other words to test the input data.	0xB6

**Register name: CONFIG11 – Address: 0x0B, Default: 0xEA**

Register Name	Address	Bit	Name	Function	Default Value
CONFIG11	0x0B	7:0	iotest_pattern2(7:0)	This is dataword2 in the IO test pattern. It is used with the seven other words to test the input data.	0xEA

**Register name: CONFIG12 – Address: 0x0C, Default: 0x45**

Register Name	Address	Bit	Name	Function	Default Value
CONFIG12	0x0C	7:0	iotest_pattern3(7:0)	This is dataword3 in the IO test pattern. It is used with the seven other words to test the input data.	0x45

**Register name: CONFIG13 – Address: 0x0D, Default: 0x1A**

Register Name	Address	Bit	Name	Function	Default Value
CONFIG13	0x0D	7:0	iotest_pattern4(7:0)	This is dataword4 in the IO test pattern. It is used with the seven other words to test the input data.	0x1A

**Register name: CONFIG14 – Address: 0x0E, Default: 0x16**

Register Name	Address	Bit	Name	Function	Default Value
CONFIG14	0x0E	7:0	iotest_pattern5(7:0)	This is dataword5 in the IO test pattern. It is used with the seven other words to test the input data.	0x16

**Register name: CONFIG15 – Address: 0x0F, Default: 0xAA**

Register Name	Address	Bit	Name	Function	Default Value
CONFIG15	0x0F	7:0	iotest_pattern6(7:0)	This is dataword6 in the IO test pattern. It is used with the seven other words to test the input data.	0xAA

**Register name: CONFIG16 – Address: 0x10, Default: 0xC6**

Register Name	Address	Bit	Name	Function	Default Value
CONFIG16	0x10	7:0	iotest_pattern7(7:0)	This is dataword7 in the IO test pattern. It is used with the seven other words to test the input data.	0xC6

**Register name: CONFIG17 – Address: 0x11, Default: 0x00**

Register Name	Address	Bit	Name	Function	Default Value
CONFIG17	0x11	7:6	Reserved	Reserved for factory use.	00
		5	Reserved	Reserved for factory use.	0
		4	Reserved	Reserved for factory use.	0
		3:0	Reserved	Reserved for factory use.	0000

**Register name: CONFIG18 – Address: 0x12, Default: 0x02**

Register Name	Address	Bit	Name	Function	Default Value
CONFIG18	0x12	7:5	Reserved	Reserved for factory use.	000
		4	Reserved	Reserved for factory use.	0
		3	daca_complement	When asserted the output to the DACA is complemented. This allows to effectively change the + and – designations of the LVDS data lines.	0
		2	dacb_complement	When asserted the output to the DACB is complemented. This allows to effectively change the + and – designations of the LVDS data lines.	0
		1	clkdiv_sync_ena	Enables the syncing of the clock divider using the OSTR signal or the FRAME signal passed through the FIFO. This selection is determined by multi_sync_sel in register CONFIG19. The internal divided-down clocks are phase-aligned after synchronizing. See Power-Up Sequence section for more detail.	1
		0	Unused	Reserved for factory use.	0

**Register name: CONFIG19 – Address: 0x13, Default: 0x00**

Register Name	Address	Bit	Name	Function	Default Value					
CONFIG19	0x13	7	bequalsa	When asserted the DACA data is driven onto DACB.	0					
		6	aequalsb	When asserted the DACB data is driven onto DACA.	0					
		5	Reserved	Reserved for factory use.	0					
		4	Unused	Reserved for factory use.	0					
		3	Unused	Reserved for factory use.	0					
		2	Unused	Reserved for factory use.	0					
		1	multi_sync_sel	Selects the signal source for multiple device and clock divider synchronization.	0					
			<table border="1"> <thead> <tr> <th>multit_sync_sel</th> <th>Sync Source</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>OSTR</td> </tr> <tr> <td>1</td> <td>FRAME through FIFO handoff</td> </tr> </tbody> </table>	multit_sync_sel	Sync Source	0	OSTR	1	FRAME through FIFO handoff	
multit_sync_sel	Sync Source									
0	OSTR									
1	FRAME through FIFO handoff									
		0	rev	Reverse the input bits for the data word. MSB becomes LSB.	0					

**Register name: CONFIG20 – Address: 0x14, Default: 0x00 (CAUSES AUTOSYNC)**

Register Name	Address	Bit	Name	Function	Default Value
CONFIG20	0x14	7:0	qmc_offseta(7:0)	Lower 8 bits of the DAC A offset correction. The offset is measured in DAC LSBs. <b>Writing this register causes an autosync to be generated. This loads the values of all four qmc_offset registers (CONFIG20-CONFIG23) into the offset block at the same time. When updating the offset values CONFIG20 should be written last. Programming any of the other three registers will not affect the offset setting.</b>	0x00

**Register name: CONFIG21 – Address: 0x15, Default: 0x00**

Register Name	Address	Bit	Name	Function	Default Value
CONFIG21	0x15	7:0	qmc_offsetb(7:0)	Lower 8 bits of the DAC B offset correction. The offset is measured in DAC LSBs.	0x00

**Register name: CONFIG22 – Address: 0x16, Default: 0x00**

Register Name	Address	Bit	Name	Function	Default Value
CONFIG22	0x16	7:3	qmc_offseta(12:8)	Upper 5 bits of the DAC A offset correction.	00000
		2	Unused	Reserved for factory use.	0
		1	Unused	Reserved for factory use.	0
		0	Unused	Reserved for factory use.	0

**Register name: CONFIG23 – Address: 0x27, Default: 0x00**

Register Name	Address	Bit	Name	Function	Default Value
CONFIG23	0x17	7:3	qmc_offsetb(12:8)	Upper 5 bits of the DAC B offset correction.	00000
		2	sif4_ena	When asserted the SIF interface becomes a 4 pin interface. The ALARM_SDO pin is turned into a dedicated output for the reading of data.	0
		1	clkpath_sleep_a	When asserted puts the clock path through DAC A to sleep. This is useful for sleeping individual DACs. Even if the DAC is asleep the clock needs to pass through it for the logic to work. However, if the chip is being put into a power down mode, then all parts of the DAC can be turned off.	0
		0	clkpath_sleep_b	When asserted puts the clock path through DAC B to sleep.	0

**Register name: CONFIG24 – Address: 0x18, Default: 0x83**

Register Name	Address	Bit	Name	Function	Default Value
CONFIG24	0x18	7	tsense_ena	Turns on the temperature sensor when asserted.	1
		6	clkrecv_sleep	When asserted the clock input receiver gets put into sleep mode. This also affects the OSTR receiver.	0
		5	Unused	Reserved for factory use.	0
		4	Reserved	Reserved for factory use.	0
		3	sleepb	When asserted DACB is put into sleep mode.	0
		2	sleepa	When asserted DACA is put into sleep mode.	0
		1	Reserved	Reserved for factory use.	1
		0	Reserved	Reserved for factory use.	1

**Register name: CONFIG25 – Address: 0x19, Default: 0x00**

Register Name	Address	Bit	Name	Function	Default Value
CONFIG25	0x19	7:3	Reserved	Reserved for factory use.	00000
		2	extref_ena	Allows the device to use an external reference or the internal reference. (0=internal, 1=external)	0
		1	Reserved	Reserved for factory use.	0
		0	Reserved	Reserved for factory use.	0

**Register name: CONFIG26 – Address: 0x1A, Default: 0x00**

Register Name	Address	Bit	Name	Function	Default Value
CONFIG26	0x1A	7	Unused	Reserved for factory use.	0
		6	Unused	Reserved for factory use.	0
		5	Unused	Reserved for factory use.	0
		4	Unused	Reserved for factory use.	0
		3	Unused	Reserved for factory use.	0
		2:0	Reserved	Reserved for factory use.	000

**Register name: CONFIG27 – Address: 0x1B, Default: 0x00**

Register Name	Address	Bit	Name	Function	Default Value
CONFIG27	0x1B	7:0	Reserved	Reserved for factory use.	0x00

**Register name: CONFIG28 – Address: 0x1C, Default: 0x00**

Register Name	Address	Bit	Name	Function	Default Value
CONFIG28	0x1C	7:0	Reserved	Reserved for factory use.	0x00

**Register name: CONFIG29 – Address: 0x1D, Default: 0x00**

Register Name	Address	Bit	Name	Function	Default Value
CONFIG29	0x1D	7:0	Reserved	Reserved for factory use.	0x00

**Register name: CONFIG30 – Address: 0x1E, Default: 0x00**

Register Name	Address	Bit	Name	Function	Default Value
CONFIG30	0x1E	7:0	Reserved	Reserved for factory use.	0x00

**Register name: VERSION31 – Address: 0x1F, Default: 0x43 (READ ONLY)**

Register Name	Address	Bit	Name	Function	Default Value
VERSION31	0x1F	7:0	deviceid(1:0)	Returns '01' for DAC3282. (Read Only)	01
		5:0	version(5:0)	A hardwired register that contains the version of the chip. (Read Only)	000011

## FUNCTIONAL DESCRIPTION

### DATA INTERFACE

The DAC3282 has a single 8-bit LVDS bus that accepts dual, 16-bit data input in byte-wide format. Data into the DAC3282 is formatted according to the diagram shown in Figure 34 where index 0 is the data LSB and index 15 is the data MSB. The data is sampled by DATACLK, a double data rate (DDR) clock.

The FRAME signal is required to indicate the beginning of a frame. The frame signal can be either a pulse or a periodic signal where the frame period corresponds to 8 samples. The pulse-width ( $t_{(FRAME)}$ ) needs to be at least equal to 1/2 the DATACLK period. FRAME is sampled by a rising edge in DATACLK.

The setup and hold requirements listed in the specifications tables must be met to ensure proper sampling.

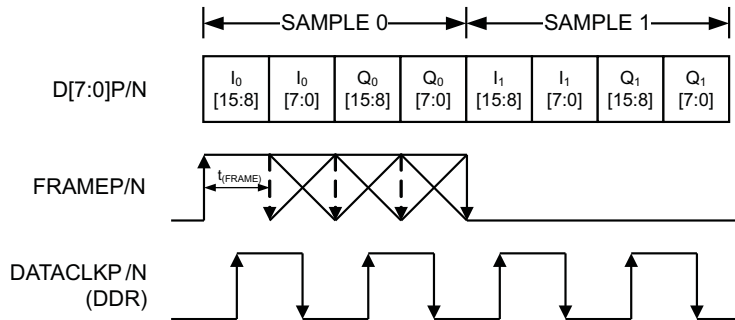


Figure 34. Byte-Wide Data Transmission Format

### INPUT FIFO

The DAC3282 includes a 2-channel, 16-bits wide and 8-samples deep input FIFO which acts as an elastic buffer. The purpose of the FIFO is to absorb any timing variations between the input data and the internal DAC data rate clock such as the ones resulting from clock-to-data variations from the data source.

Figure 35 shows the block diagram of the FIFO.

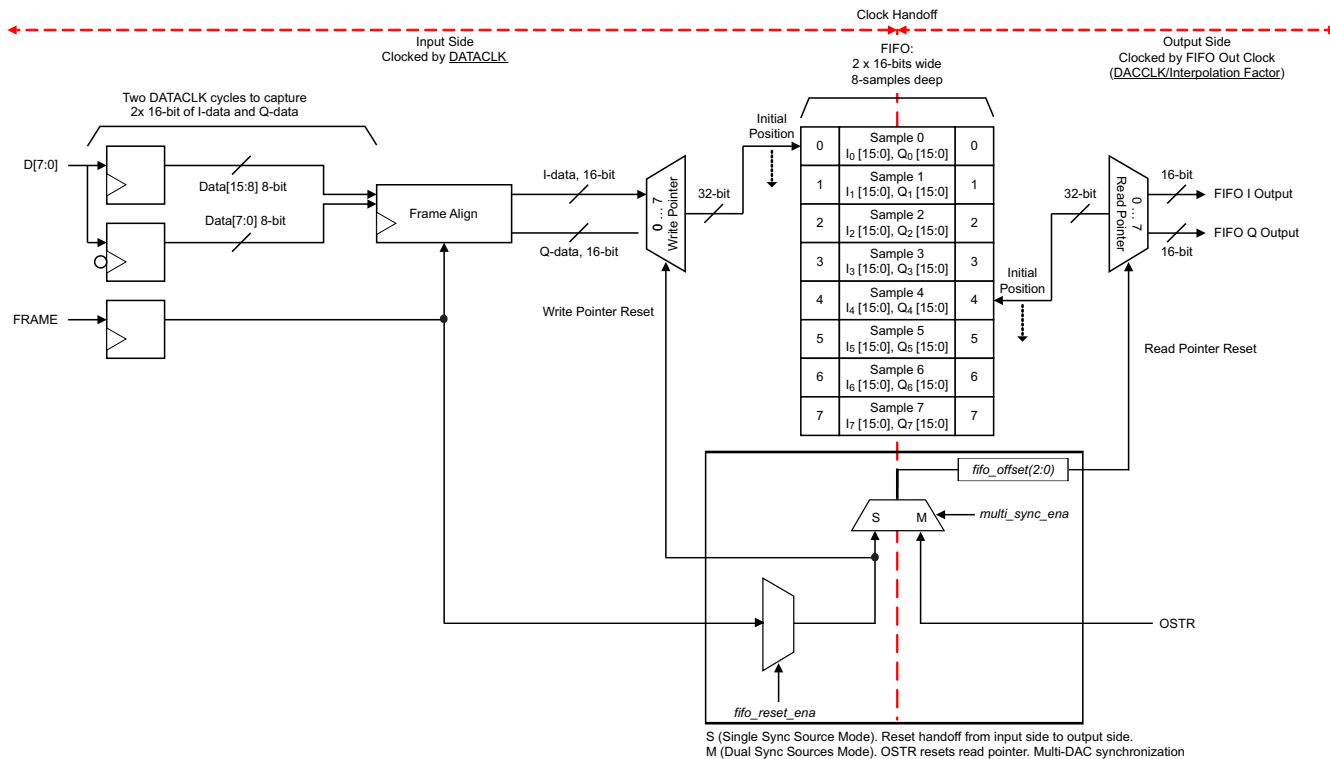


Figure 35. DAC3282 FIFO Block Diagram

Data is written to the device 8-bits at a time on the rising and falling edges of DATACLK. In order to form a complete 32-bit wide sample (16-bit I-data and 16-bit Q-data) two DATACLK periods are required as shown in Figure 36. Each 32-bit wide sample is written into the FIFO at the address indicated by the write pointer. Similarly, data from the FIFO is read by the FIFO Out Clock 32-bits at a time from the address indicated by the read pointer. The FIFO Out Clock is generated internally from the DACCLK signal and its rate is equal to DACCLK/Interpolation. Each time a FIFO write or FIFO read is done the corresponding pointer moves to the next address.

The reset position for the FIFO read and write pointers is set by default to addresses 0 and 4 as shown in Figure 35. This offset gives optimal margin within the FIFO. The default read pointer location can be set to another value using fifo\_offset(2:0) in register CONFIG3. Under normal conditions data is written-to and read-from the FIFO at the same rate and consequently the write and read pointer gap remains constant. If the FIFO write and read rates are different, the corresponding pointers will be cycling at different speeds which could result in pointer collision. Under this condition the FIFO attempts to read and write data from the same address at the same time which will result in errors and thus must be avoided.

The FRAME signal besides acting as a frame indicator can also be used to reset the FIFO pointers to their initial location. Unlike Data, the FRAME signal is latched only on the rising edges of DATACLK. When a rising edge occurs on FRAME, the pointers will return to their original position. The write pointer is always set back to position 0 upon reset. The read pointer reset position is determined by fifo\_offset (address 4 by default).

Similarly, the read pointer sync source is selected by multi\_sync\_sel (CONFIG19). Either the FRAME or OSTR signal can be set to reset the read pointer. If FRAME is used to reset the read pointer, the FIFO Out Clock will recapture the FRAME signal to reset the read pointer. This clock domain transfer (DATACLK to FIFO Out Clock) results in phase ambiguity of the reset signal. This limits the precise control of the output timing and makes full synchronization of multiple devices difficult.

To alleviate this, the device offers the alternative of resetting the FIFO read pointer independently of the write pointer by using the OSTR signal. The OSTR signal is sampled by DACCLK and must satisfy the timing requirements in the specification table. In order to minimize the skew it is recommended to use the same clock distribution device such as Texas Instruments CDCE62005 to provide the DACCLK and OSTR signals to all the DAC3282 devices in the system. Swapping the polarity of the DACCLK output with respect to the OSTR output establishes proper phase relationship.

The FIFO pointers reset procedure can be done periodically or only once during initialization as the pointers automatically return to the initial position when the FIFO has been filled. To reset the FIFO periodically, it is necessary to have FRAME and OSTR signals to repeat at multiple of 8 FIFO samples. To disable FIFO reset, set `fifo_reset_ena` and `multi_sync_ena` (CONFIG0) to 0.

The frequency limitation for the FRAME signal is the following

$$f_{\text{SYNC}} = f_{\text{DATACLK}} / (n \times 16) \text{ where } n = 1, 2, \dots$$

The frequency limitation for the OSTR signal is the following:

$$f_{\text{OSTR}} = f_{\text{DAC}} / (n \times \text{interpolation} \times 8) \text{ where } n = 1, 2, \dots$$

The frequencies above are at maximum when  $n = 1$ . This is when FRAME and OSTR have a rising edge transition every 8 FIFO samples. The occurrence can be made less frequent by setting  $n > 1$ , for example, every  $n \times 8$  FIFO samples.

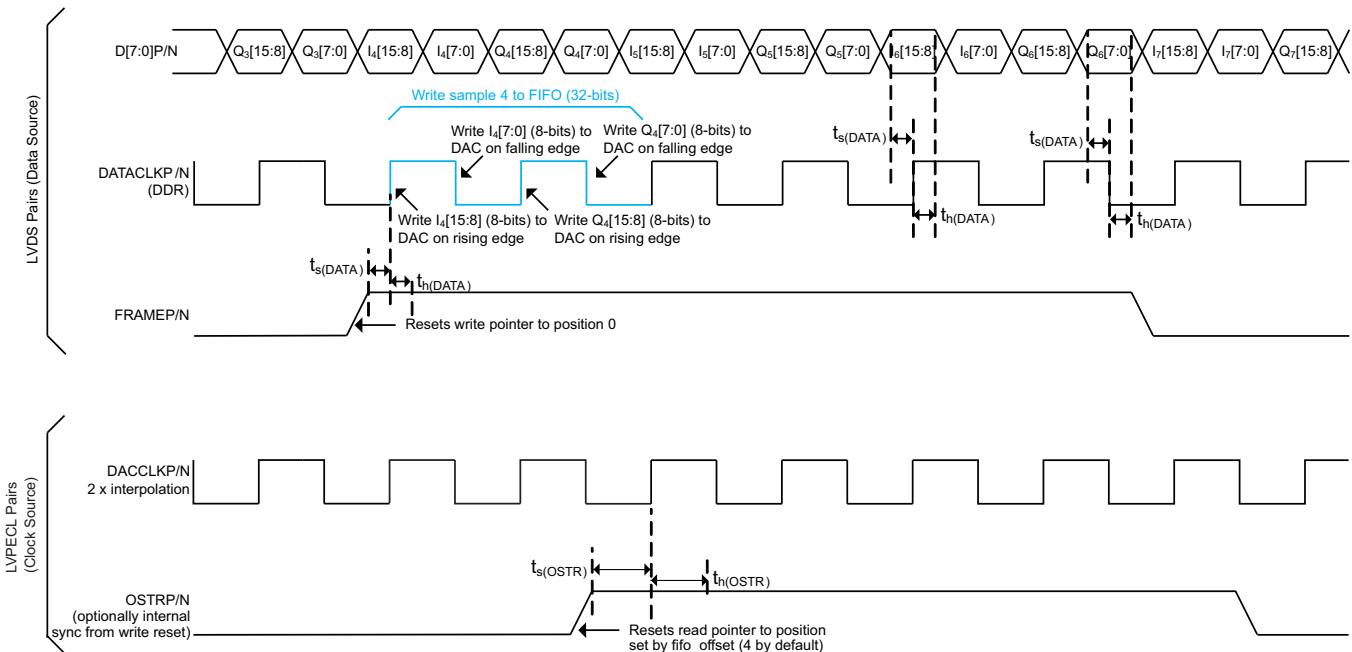


Figure 36. FIFO Write Description

## FIFO ALARMS

The FIFO only operates correctly when the write and read pointers are positioned properly. If either pointer over or under runs the other, samples will be duplicated or skipped. To prevent this, register CONFIG7 can be used to track three FIFO related alarms:

- `alarm_fifo_2away`. Occurs when the pointers are within two addresses of each other.
- `alarm_fifo_1away`. Occurs when the pointers are within one address of each other.
- `alarm_fifo_collision`. Occurs when the pointers are equal to each other.

These three alarm events are generated asynchronously with respect to the clocks and can be accessed either through CONFIG7 or through the ALARM\_SDO pin.



## FIFO MODES OF OPERATION

The DAC3282 input FIFO can be completely bypassed through registers *config0* and *config19*. The register configuration for each mode is described in [Table 4](#).

Register	Control Bits
CONFIG0	fifo_ena, fifo_reset_ena, multi_sync_ena
CONFIG19	multi_sync_sel

**Table 4. FIFO Operation Modes**

FIFO Mode	config0 FIFO Bits			Config19
	fifo_ena	fifo_reset_ena	multi_sync_ena	multi_sync_sel
Dual Sync Sources	1	1	1	0
Single Sync Source	1	1	1	1
Bypass	0	X	X	X

### Dual Sync Sources Mode

This is the recommended mode of operation for those applications that require precise control of the output timing. In Dual Sync Sources mode, the FIFO write and read pointers are reset independently. The FIFO write pointer is reset using the LVDS FRAME signal, and the FIFO read pointer is reset using the LVPECL OSTR signal. This allows LVPECL OSTR signal to control the phase of the output for either a single chip or multiple chips. Multiple devices can be fully synchronized in this mode.

### Single Sync Source Mode

In Single Sync Source mode, the FIFO write and read pointers are reset from the same LVDS FRAME signal. This mode has a possibility of up to 2 DAC clocks offset between the outputs of multiple devices (the DAC outputs of the same device maintain the phase phase). Applications requiring exact output timing control will need Dual Sync Sources mode instead of Single Sync Source Mode. A rising edge for FIFO and clock divider sync is recommended. Periodic sync signal is not recommended due to non-deterministic latency of the sync signal through the clock domain transfer.

### Bypass Mode

In FIFO bypass mode, the FIFO block is not used. As a result the input data is handed off from the DATACLK to the DACCLK domain without any compensation. In this mode the relationship between DATACLK and DACCLK  $t_{(align)}$  is critical and used as a synchronizing mechanism for the internal logic. Due to the  $t_{(align)}$  constraint it is highly recommended that a clock synchronizer such as Texas Instruments' CDCM7005 or CDCE62005 is used to provide both clock inputs. In bypass mode the pointers have no effect on the data path or handoff.

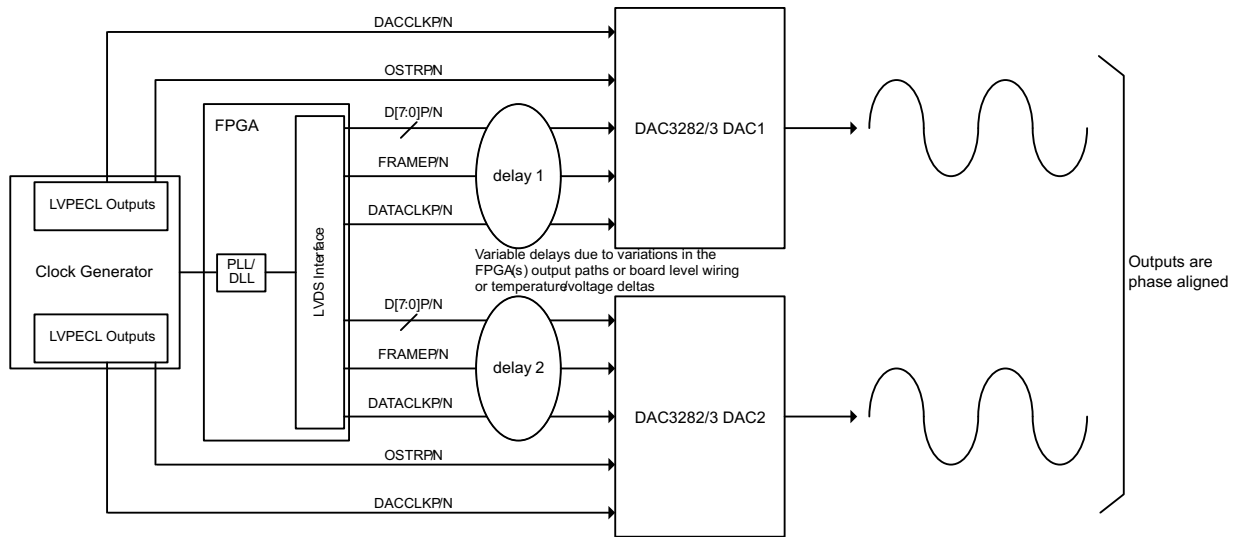
## MULTI-DEVICE SYNCHRONIZATION

In various applications, such as multi antenna systems where the various transmit channels information is correlated, it is required that multiple DAC devices are completely synchronized such that their outputs are phase aligned. The DAC3282 architecture supports this mode of operation.

### MULTI-DEVICE SYNCHRONIZATION: DUAL SYNC SOURCES MODE

For single or multi-device synchronization it is important that delay differences in the data are absorbed by the device so that latency through the device remains the same. Furthermore, to guarantee that the outputs from each DAC are phase aligned it is necessary that data is read from the FIFO of each device simultaneously. In the DAC3282 this is accomplished by operating the multiple devices in Dual Sync Sources mode. In this mode the additional OSTR signal is required by each DAC3282 to be synchronized.

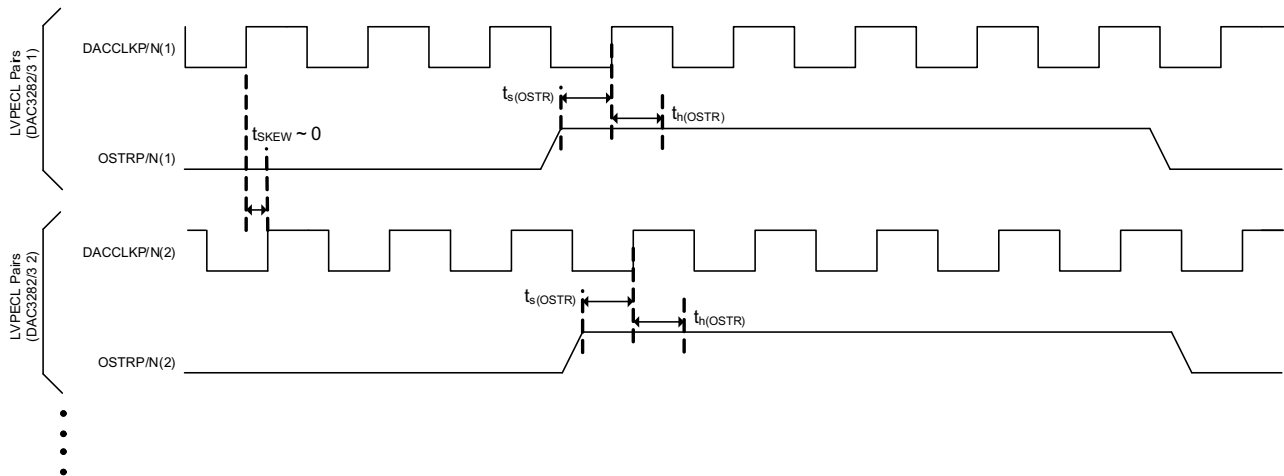
Data into the device is input as LVDS signals from one or multiple baseband ASICs or FPGAs. Data into the multiple DAC devices can experience different delays due to variations in the digital source output paths or board level wiring. These different delays can be effectively absorbed by the DAC3282 FIFO so that all outputs are phase aligned correctly.



**Figure 37. Synchronization System in Dual Sync Sources Mode with PLL Bypassed**

For correct operation both OSTR and DACCLK must be generated from the same clock domain. The OSTR signal is sampled by DACCLK and must satisfy the timing requirements in the specifications table. If the clock generator does not have the ability to delay the DACCLK to meet the OSTR timing requirement, the polarity of the DACCLK outputs can be swapped with respect to the OSTR ones to create 180 degree phase delay of the DACCLK. This may help establish proper setup and hold time requirement of the OSTR signal.

Careful board layout planning must be done to ensure that the DACCLK and OSTR signals are distributed from device to device with the lowest skew possible as this will affect the synchronization process. In order to minimize the skew across devices it is recommended to use the same clock distribution device to provide the DACCLK and OSTR signals to all the DAC devices in the system.



**Figure 38. Timing Diagram for LVPECL Synchronization Signals**

The following steps are required to ensure the devices are fully synchronized. The procedure assumes all the DAC3282 devices have a DACCLK and OSTR signal and must be carried out on each device.

1. Start-up the device as described in the power-up sequence. Set the DAC3282 in Dual Sync Sources mode and select OSTR as the FIFO output pointer sync source and clock divider sync source (*multi\_sync\_sel* in register *config19*).
2. Sync the clock divider and FIFO pointers.
3. Verify there are no FIFO alarms either through register *config7* or through the ALARM\_SDO pin.

After these steps all the DAC3282 outputs will be synchronized.

### MULTI-DEVICE OPERATION: SINGLE SYNC SOURCE MODE

In Single Sync Source mode, the FIFO write and read pointers are reset from the same FRAME source. Although the FIFO in this mode can still absorb the data delay differences due to variations in the digital source output paths or board level wiring, it is impossible to guarantee data will be read from the FIFO of different devices simultaneously thus preventing exact phase alignment.

The FIFO read pointer reset is handoff between the two clock domains (DATACLK and FIFO OUT CLOCK) by simply re-sampling the write pointer reset. Since the two clocks are asynchronous there is a small but distinct possibility of a meta-stability during the pointer handoff. This meta-stability can cause the outputs of the multiple devices to slip by up to 2 DAC clock cycles.

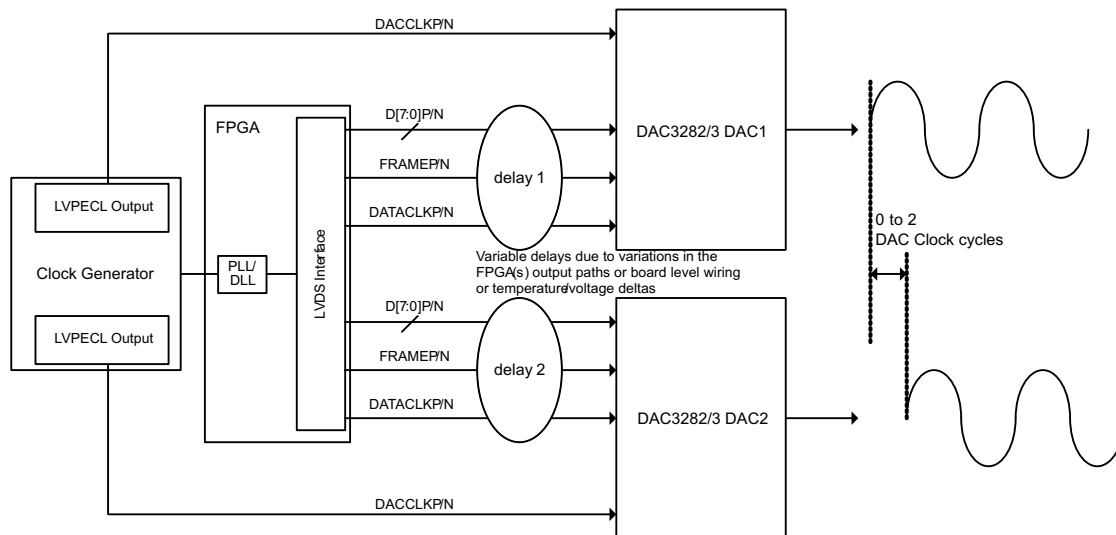


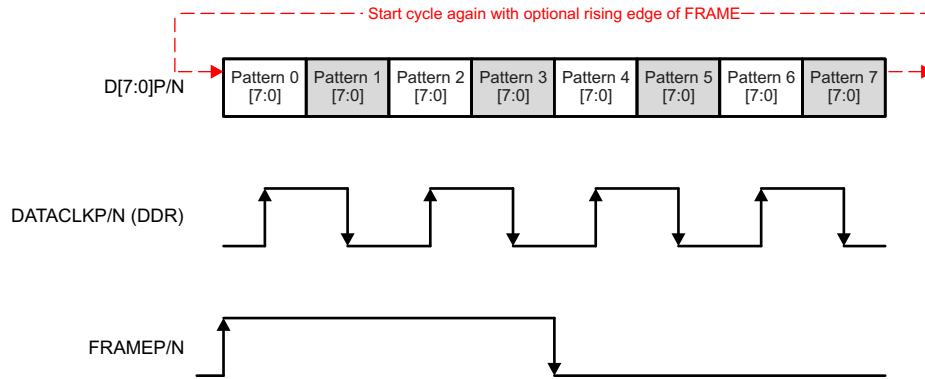
Figure 39. Multi-Device Operation in Single Sync Source Mode

### DATA PATTERN CHECKER

The DAC3282 incorporates a simple pattern checker test in order to determine errors in the data interface. The main cause of failure is setup/hold timing issues. The test mode is enabled by asserting *iotest\_ena* in register *config1*. In test mode the analog outputs are deactivated regardless of the state of TXENABLE.

The data pattern key used for the test is 8 words long and is specified by the contents of *iotest\_pattern[0:7]* in registers *config9* through *config16*. The data pattern key can be modified by changing the contents of these registers.

The first word in the test frame is determined by a rising edge transition in FRAME. At this transition, the *pattern0* word should be input to the data pins. Patterns 1 through 7 should follow sequentially on each edge of DATACLK (rising and falling). The sequence should be repeated until the pattern checker test is disabled by setting *iotest\_ena* back to "0". It is not necessary to have a rising FRAME edge aligned with every *pattern0* word, just the first one to mark the beginning of the series.



**Figure 40. IO Pattern Checker Data Transmission Format**

The test mode determines if the 8-bit LVDS data D[7:0]P/N of all the patterns were received correctly by comparing the received data against the data pattern key. If any of the 8-bit data D[7:0]P/N were received incorrectly, the corresponding bits in *iotest\_results(7:0)* in register *config8* will be set to “1” to indicate bit error location. Furthermore, the error condition will trigger the *alarm\_from\_iotest* bit in register *config7* to indicate a general error in the data interface. When data pattern checker mode is enabled, this alarm in register *config7*, bit 3 is the only valid alarm. Other alarms in register *config7* are not valid and can be disregarded.

For instance, *pattern0* is programmed to the default of 0x7A. If the received Pattern 0 is 0x7B, then bit 0 in *iotest\_results(7:0)* will be set to “1” to indicate an error in bit 0 location. The *alarm\_from\_iotest* will also be set to “1” to report the data transfer error. The user can then narrow down the error from the bit location information and implement the fix accordingly.

The alarms can be cleared by writing 0x00 to *iotest\_results(7:0)* and “0” to *alarm\_from\_iotest* through the serial interface. The serial interface will read back 0s if there are no errors or if the errors are cleared. The corresponding alarm bit will remain a “1” if the errors remain.

It is recommended to enable the pattern checker and then run the pattern sequence for 100 or more complete cycles before clearing the *iotest\_results(7:0)* and *alarm\_from\_iotest*. This will eliminate the possibility of false alarms generated during the setup sequence.

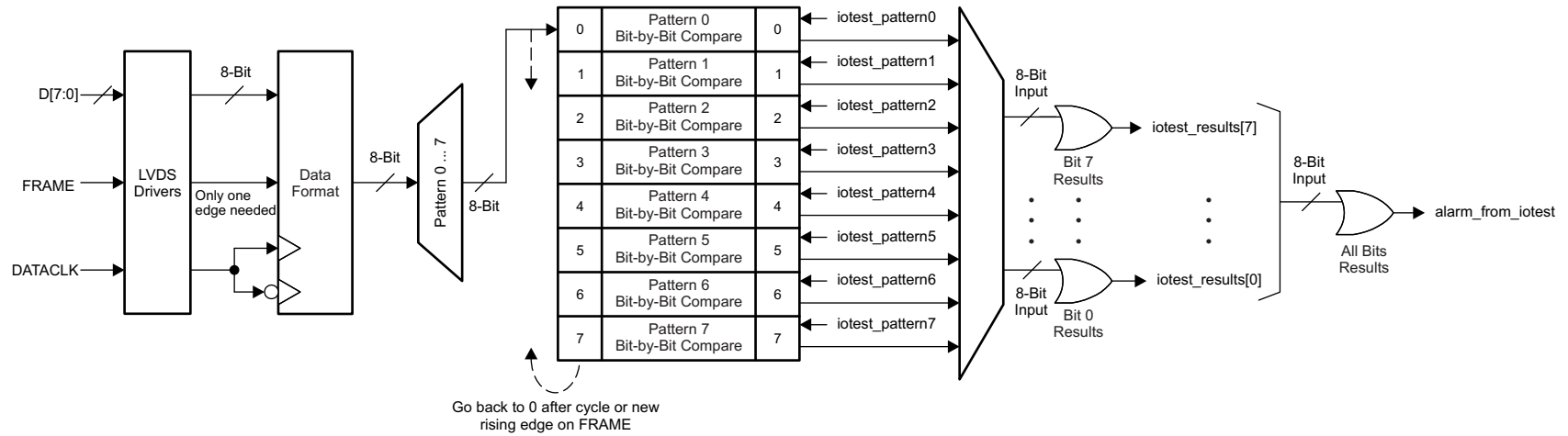


Figure 41. DAC3282 Pattern Check Block Diagram

## FIR FILTERS

The DAC3282 has two FIR filters, a 2x interpolation FIR (FIR0) and a non-interpolating FIR (FIR4) that compensates for the sinc droop of the DAC on zero-IF applications. The correction filter is placed before the interpolating filter and can only be used with both FIRs enabled.

Figure 42 shows the magnitude spectrum response for FIR0, a 59-tap interpolating half-band filter. The transition band is from  $0.4$  to  $0.6 \times f_{IN}$  (the input data rate for the FIR filter) with  $< 0.002\text{dB}$  of pass-band ripple and  $> 85\text{ dB}$  stop-band attenuation. Figure 43 shows the transition band region from  $0.36$  to  $0.46 \times f_{IN}$ . Up to  $0.45 \times f_{IN}$  there is less than  $0.5\text{ dB}$  of attenuation.

The DAC sample and hold operation results in the well known  $\sin(x)/x$  or  $\text{sinc}(x)$  frequency response shown in Figure 44 (red line). The DAC3282 has a 5-tap inverse sinc filter (FIR4) placed before the 2x interpolation filter to compensate for this effect up to  $0.2 \times f_{DAC}$ . The inverse sinc filter runs at the input data rate and is operational only if the 2x interpolation filter is enabled as well, correspondingly the rate of this filter is always half of the DAC update rate. As a result, the filter cannot completely flatten the frequency response of the sample and hold output as shown in Figure 44.

Figure 45 shows the magnitude spectrum for FIR4 over the correction range. The inverse sinc filter response (Figure 45, black line) has approximately the opposite frequency response to  $\sin(x)/x$  between  $0$  to  $0.2 \times f_{DAC}$ , resulting in the corrected response in Figure 45 (blue line). Between  $0$  to  $0.2 \times f_{DAC}$ , the inverse sinc filter compensates for the sample and hold roll-off with less than  $0.04\text{-dB}$  error.

The zero-IF sinc filter has a gain  $> 1$  at all frequencies. Therefore, the input data must be reduced from full scale to prevent saturation in the filter. The amount of back-off required depends on the signal frequency, and is set such that at the signal frequencies the combination of the input signal and filter response is less than  $1$  ( $0\text{ dB}$ ). For example, if the signal input to FIR4 is at  $0.1 \times f_{DAC}$ , the response of FIR4 is  $0.1\text{ dB}$ , and the signal must be backed off from full scale by  $0.1\text{ dB}$  to avoid saturation.

The filter taps for all digital filters are listed in Table 5.

Note that the loss of signal amplitude may result in lower SNR due to decrease in signal amplitude.

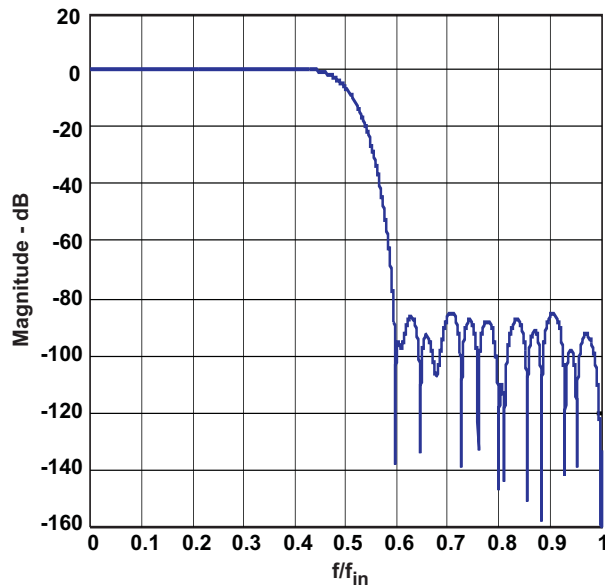


Figure 42. Magnitude Spectrum for FIR0

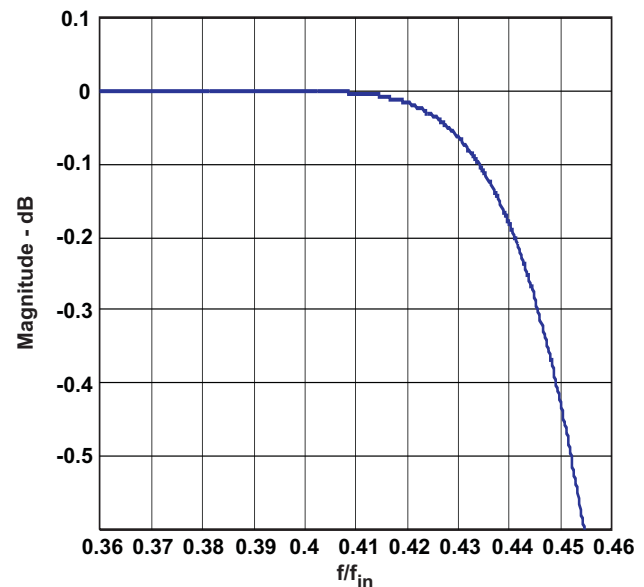


Figure 43. FIR0 Transition Band

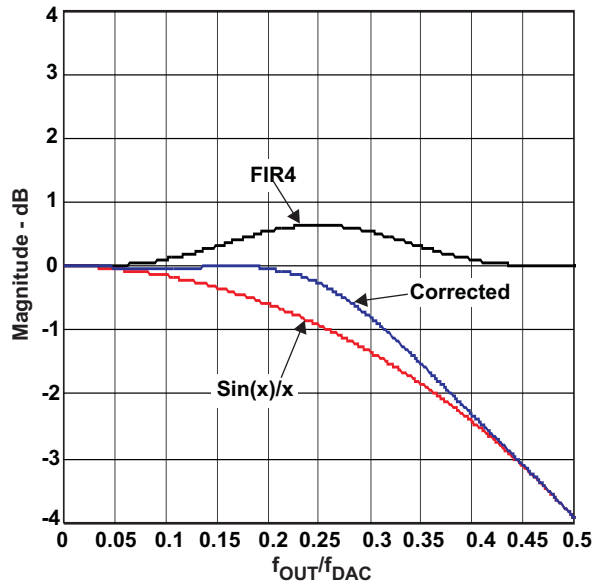


Figure 44. Magnitude Spectrum for Zero-IF Sinc Correction Filter up to  $0.5 \times f_{DAC}$

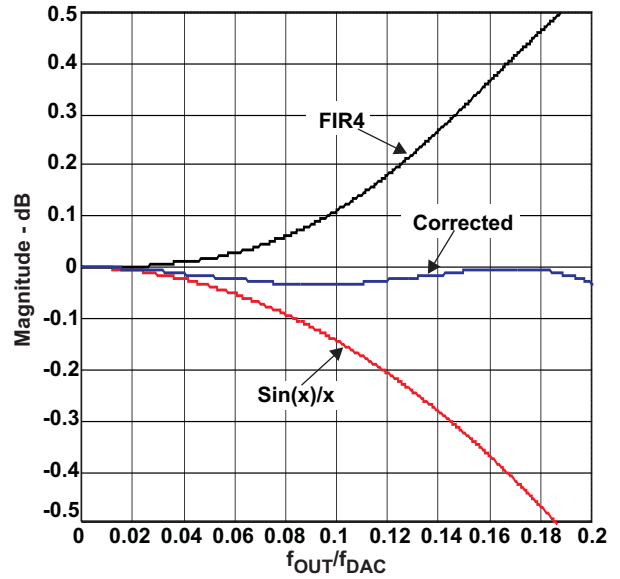


Figure 45. Correction Range of Zero-IF Sinc Correction Filter 0 to  $0.2 \times f_{DAC}$

**Table 5. FIR Filter Coefficients**

FIR0 2x Interpolating Half-Band Filter		FIR4 Non-Interpolating Zero-IF Sinc Correction Filter
59 Taps		5 Taps
4	4	1
0	0	-5
-12	-12	<b>264</b> <sup>(1)</sup>
0	0	-5
28	28	1
0	0	
-58	-58	
0	0	
108	108	
0	0	
-188	-188	
0	0	
308	308	
0	0	
-483	-483	
0	0	
734	734	
0	0	
-1091	-1091	
0	0	
1607	1607	
0	0	
-2392	-2392	
0	0	
3732	3732	
0	0	
-6681	-6681	
0	0	
20768	20768	
<b>32768</b> <sup>(1)</sup>		

(1) Center taps are highlighted in BOLD.



## COARSE MIXER

The DAC3282 has a coarse mixer block capable of shifting the input signal spectrum by the fixed mixing frequencies  $f_s/2$  or  $\pm f_s/4$ . The coarse mixing function is built into the interpolation filter and thus FIR0 must be enabled to use it.

Treating channels A and B as a complex vector of the form  $I(t) + j Q(t)$ , where  $I(t) = A(t)$  and  $Q(t) = B(t)$ , the outputs of the coarse mixer,  $A_{OUT}(t)$  and  $B_{OUT}(t)$  are equivalent to:

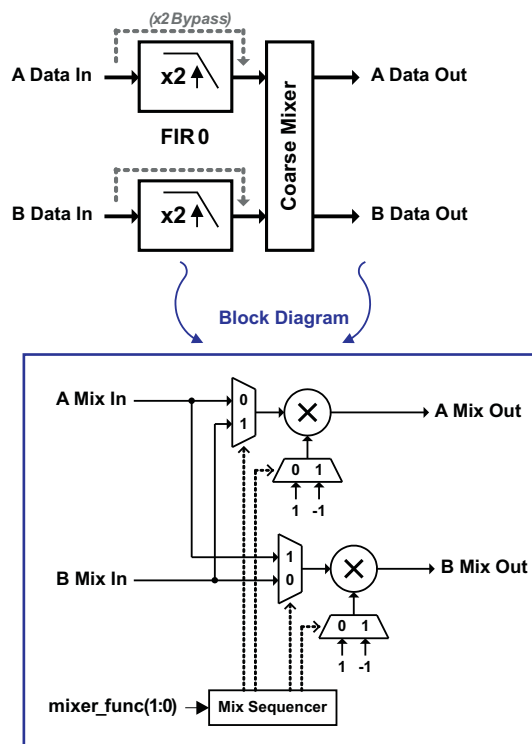
$$A_{OUT}(t) = A(t)\cos(2\pi f_{CMIX}t) - B(t)\sin(2\pi f_{CMIX}t) \tag{1}$$

$$B_{OUT}(t) = A(t)\sin(2\pi f_{CMIX}t) + B(t)\cos(2\pi f_{CMIX}t) \tag{2}$$

where  $f_{CMIX}$  is the fixed mixing frequency selected by `mixer_func(1:0)`. For  $f_s/2$ ,  $+f_s/4$  and  $-f_s/4$  the above operations result in the simple mixing sequences shown in [Table 6](#).

**Table 6. Coarse Mixer Sequences**

Mode	mixer_func(1:0)	Mixing Sequence
Normal (Low Pass, No Mixing)	00	$A_{OUT} = \{ +A, +A, +A, +A \}$ $B_{OUT} = \{ +B, +B, +B, +B \}$
$f_s/2$	01	$A_{OUT} = \{ +A, -A, +A, -A \}$ $B_{OUT} = \{ +B, -B, +B, -B \}$
$+f_s/4$	10	$A_{OUT} = \{ +A, -B, -A, +B \}$ $B_{OUT} = \{ +B, +A, -B, -A \}$
$-f_s/4$	11	$A_{OUT} = \{ +A, +B, -A, -B \}$ $B_{OUT} = \{ +B, -A, -B, +A \}$



**Figure 46. Coarse Mixers Block Diagram**

The coarse mixer in the DAC3282 treats the A and B inputs as complex input data and for most mixing frequencies produces a complex output. Only when the mixing frequency is set to  $f_s/2$  the A and B channels can be maintained isolated as shown in [Table 6](#). In this case the two channels are upconverted as independent signals. By setting the mixer to  $f_s/2$  the FIR0 outputs are inverted thus behaving as a high-pass filter.

**Table 7. Dual-Channel Real Upconversion Options**

FIR Mode	Input Frequency <sup>(1)</sup>	Output Frequency <sup>(1)</sup>	Signal Bandwidth <sup>(1)</sup>	Spectrum Inverted?
Low pass	0.0 to 0.4 x f <sub>DATA</sub>	0.0 to 0.4 x f <sub>DATA</sub>	0.4 x f <sub>DATA</sub>	No
High pass	0.0 to 0.4 x f <sub>DATA</sub>	0.6 to 1.0 x f <sub>DATA</sub>	0.4 x f <sub>DATA</sub>	Yes

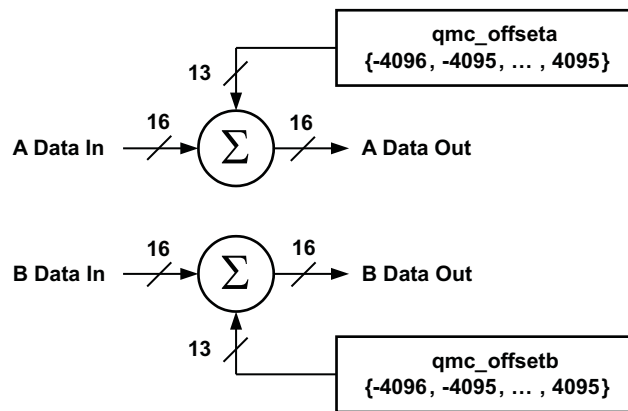
(1) f<sub>DATA</sub> is the input data rate of each channel after de-interleaving.

**DIGITAL OFFSET CONTROL**

The qmc\_offseta(12:0) and qmc\_offsetb(12:0) values in registers CONFIG20 through CONFIG23 can be used to independently adjust the A and B path DC offsets. Both offset values are in represented in 2s-complement format with a range from -4096 to 4095.

Note that a write to register CONFIG20 is required to load the values of all four qmc\_offset registers (CONFIG20-CONFIG23) into the offset block simultaneously. When updating the offset values CONFIG20 should be written last. Programming any of the other three registers will not affect the offset setting.

The offset value adds a digital offset to the digital data before digital-to-analog conversion. Since the offset is added directly to the data it may be necessary to back off the signal to prevent saturation. Both data and offset values are LSB aligned.



**Figure 47. Digital Offset Block Diagram**

**TEMPERATURE SENSOR**

The DAC3282 incorporates a temperature sensor block which monitors the temperature by measuring the voltage across 2 transistors. The voltage is converted to an 8-bit digital word using a successive-approximation (SAR) analog to digital conversion process. The result is scaled, limited and formatted as a twos complement value representing the temperature in degrees Celsius.

The sampling is controlled by the serial interface signals SDENB and SCLK. If the temperature sensor is enabled (tsense\_ena = 1 in register CONFIG24) a conversion takes place each time the serial port is written or read. The data is only read and sent out by the digital block when the temperature sensor is read in register CONFIG5. The conversion uses the first eight clocks of the serial clock as the capture and conversion clock, the data is valid on the falling eighth SCLK. The data is then clocked out of the chip on the rising edge of the ninth SCLK. No other clocks to the chip are necessary for the temperature sensor operation. As a result the temperature sensor is enabled even when the device is in sleep mode.

In order for the process described above to operate properly, the serial port read from CONFIG5 must be done with an SCLK period of at least 1 μs. If this is not satisfied the temperature sensor accuracy is greatly reduced.

## POWER-UP SEQUENCE

The following startup sequence is recommended to power-up the DAC3282:

1. Set TXENABLE low.
2. Supply all 1.8V voltages (DACVDD, DIGVDD, CLKVDD and VFUSE) and all 3.3V voltages (AVDD). The 1.8V and 3.3V supplies can be powered up simultaneously or in any order. There are no specific requirements on the ramp rate for the supplies.
3. Provide all LVPECL inputs: DACCLKP/N and the optional OSTRP/N. These inputs can also be provided after the SIF register programming.
4. Toggle the RESETB pin for a minimum of 25ns active low pulse width.
5. Program all the SIF registers.
6. FIFO configuration needed for synchronization:
  - (a) Program *fifo\_reset\_ena* (config0, bit<5>) to enable FRAMEP/N as the FIFO input pointer sync source.
  - (b) Program *multi\_sync\_ena* (config0, bit<4>) to enable syncing of the FIFO output pointer.
  - (c) Program *multi\_sync\_sel* (config19, bit<1>) to select the FIFO output pointer and clock divider sync source
7. Clock divider configuration needed for synchronization:
  - (a) Program *clkdiv\_sync\_ena* (config18, bit<1>) to “1” to enable clock divider sync.
8. Provide all LVDS inputs (D[7:0]P/N, DATACLKP/N, and FRAMEP/N) simultaneously. Synchronize the FIFO and clock divider by providing the pulse or periodic signals needed.
  - (a) For Single Sync Source Mode where FRAMEP/N is used to sync the FIFO, a single rising edge for FIFO, FIFO data formatter, and clock divider sync is recommended. Periodic sync signal is not recommended due to the non-deterministic latency of the sync signal through the clock domain transfer.
  - (b) For Dual Sync Sources Mode, both single pulse or periodic sync signals can be used.
9. FIFO and clock divider configurations after all the sync signals have provided the initial sync pulses needed for synchronization:
  - (a) For Single Sync Source Mode where the clock divider sync source is FRAMEP/N, clock divider syncing may be disabled after DAC3282 initialization and before the data transmission by setting *clkdiv\_sync\_ena* (config18, bit<1>) to “0”. This is to prevent accidental syncing of the clock divider when sending FRAMEP/N pulse to other digital blocks.
  - (b) For Dual Sync Sources Mode, where the clock divider sync source is from the OSTRP/N, the clock divider syncing may be enabled at all time.
  - (c) Optionally, to prevent accidental syncing of the FIFO, disable FIFO syncing by setting *fifo\_reset\_ena* and *multi\_sync\_ena* to “0” after the FIFO input and output pointers are initialized. If the FIFO sync remains enabled after initialization, the FRAMEP/N pulse must occur in ways to not disturb the FIFO operation. Refer to the [INPUT FIFO](#) section for detail.
10. Enable transmit of data by asserting the TXENABLE pin.
11. At all time, if any of the clocks (i.e. DATACLK or DACCLK) is lost or FIFO collision alarm is detected, a complete resynchronization of the DAC is necessary. Please set TXENABLE low and repeat step 6 through 10. Program the FIFO configuration and clock divider configuration per step 6 and 7 appropriately to accept the new sync pulse or pulses for the synchronization.

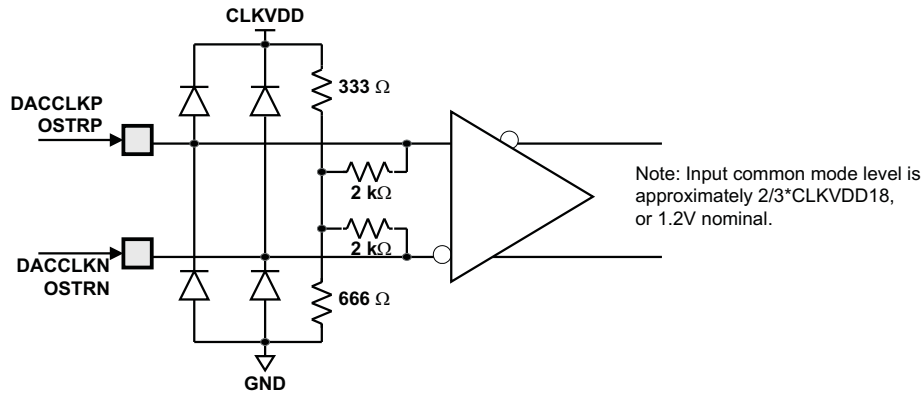
## SLEEP MODES

The DAC3282 features independent sleep control of each DAC (sleepa and sleepb), their corresponding clock path (clkpath\_sleep\_a and clkpath\_sleep\_b) as well as the clock input receiver of the device (clkrecv\_sleep). The sleep control of each of these components is done through the SIF interface and is enabled by setting a 1 to the corresponding sleep register.

Complete power down of the device is set by setting all of these components to sleep. Under this mode the supply power consumption is reduced to 15mW. Power-up time in this case will be in the milliseconds range. Alternatively for those applications where power-up and power-down times are critical it is recommended to only set the DACs to sleep through the sleepa and sleepb registers. In this case both the sleep and wake-up times are only 90µs.

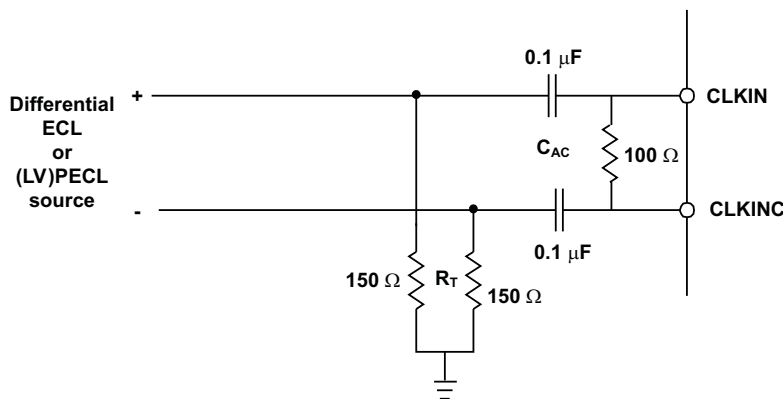
**LVPECL INPUTS**

Figure 48 shows an equivalent circuit for the DAC input clock (DACCLKP/N) and the FIFO output strobe clock (OSTRP/N).



**Figure 48. DACCLKP/N and OSTRP/N Equivalent Input Circuit**

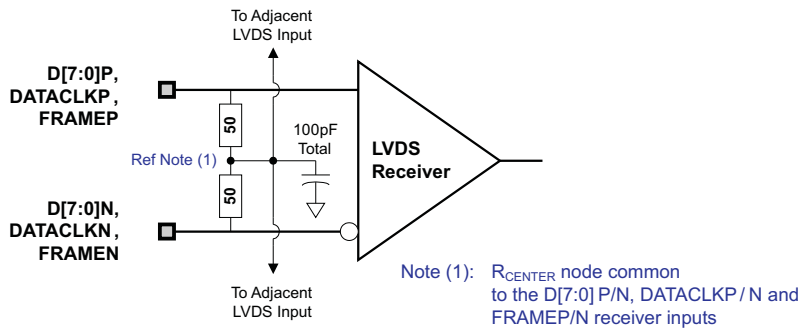
Figure 49 shows the preferred configuration for driving the CLKIN/CLKINC input clock with a differential ECL/PECL source.



**Figure 49. Preferred Clock Input Configuration With a Differential ECL/PECL Clock Source**

**LVDS INPUTS**

The D[7:0]P/N, DATACLKP/N and FRAMEP/N LVDS pairs have the input configuration shown in Figure 50. Figure 51 shows the typical input levels and common-mode voltage used to drive these inputs.



**Figure 50. D[7:0]P/N, DATACLKP/N and FRAMEP/N LVDS Input Configuration**

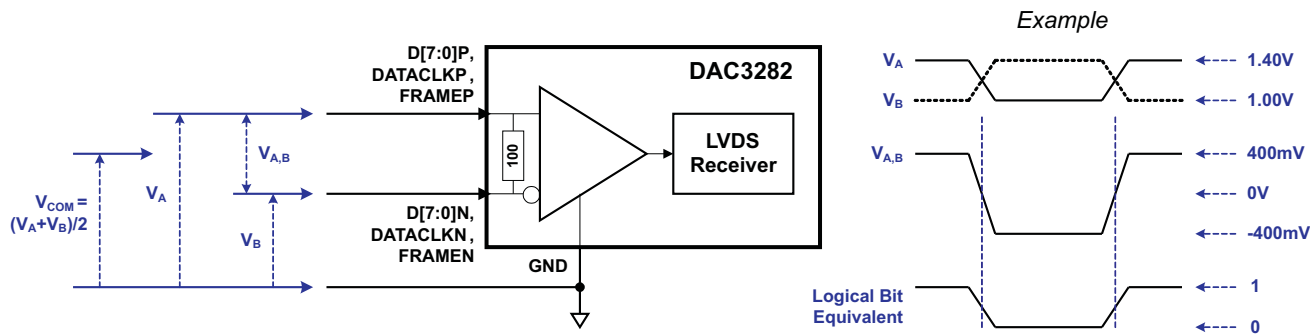


Figure 51. LVDS Data (D[7:0]P/N, DATACLKP/N, FRAMEP/N Pairs) Input Levels

Table 8. Example LVDS Data Input Levels

Applied Voltages		Resulting Differential Voltage	Resulting Common-Mode Voltage	Logical Bit Binary Equivalent
V <sub>A</sub>	V <sub>B</sub>	V <sub>A,B</sub>	V <sub>COM</sub>	
1.4 V	1.0 V	400 mV	1.2 V	1
1.0 V	1.4 V	-400 mV		0
1.2 V	0.8 V	400 mV	1.0 V	1
0.8 V	1.2 V	-400 mV		0

CMOS DIGITAL INPUTS

Figure 52 shows a schematic of the equivalent CMOS digital inputs of the DAC3282. SDIO, SCLK and TXENABLE have pull-down resistors while SDENB and RESETB have pull-up resistors internal to the DAC3282. See the specification table for logic thresholds. The pull-up and pull-down circuitry is approximately equivalent to 100kΩ.

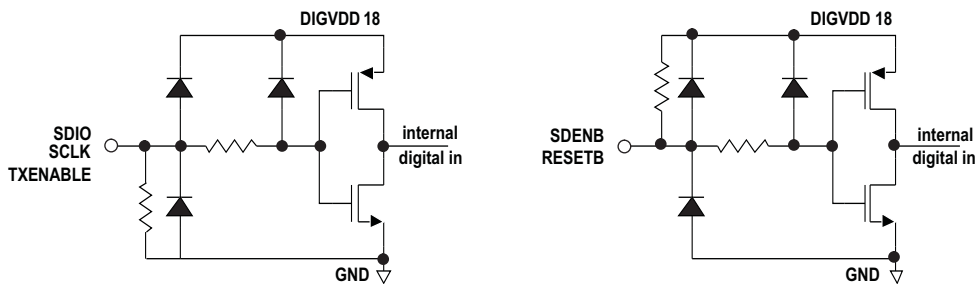


Figure 52. CMOS/TTL Digital Equivalent Input

REFERENCE OPERATION

The DAC3282 uses a bandgap reference and control amplifier for biasing the full-scale output current. The full-scale output current is set by applying an external resistor R<sub>BIAS</sub> to pin BIASJ. The bias current I<sub>BIAS</sub> through resistor R<sub>BIAS</sub> is defined by the on-chip bandgap reference voltage and control amplifier. The default full-scale output current equals 16 times this bias current and can thus be expressed as:

$$I_{OUT_{FS}} = 16 \times I_{BIAS} = 16 \times V_{EXTIO} / R_{BIAS}$$

Each DAC has a 4-bit independent coarse gain control via **coarse\_daca(3:0)** and **coarse\_dacb (3:0)** in the CONFIG4 register. Using gain control, the I<sub>OUT<sub>FS</sub></sub> can be expressed as:

$$I_{OUTA_{FS}} = (DACA\_gain + 1) \times I_{BIAS} = (DACA\_gain + 1) \times V_{EXTIO} / R_{BIAS}$$

$$I_{OUTB_{FS}} = (DACB\_gain + 1) \times I_{BIAS} = (DACB\_gain + 1) \times V_{EXTIO} / R_{BIAS}$$

Where  $V_{EXTIO}$  is the voltage at terminal EXTIO. The bandgap reference voltage delivers an accurate voltage of 1.2V. This reference is active when **extref\_ena** = '0' in CONFIG25. An external decoupling capacitor  $C_{EXT}$  of 0.1  $\mu$ F should be connected externally to terminal EXTIO for compensation. The bandgap reference can additionally be used for external reference operation. In that case, an external buffer with high impedance input should be applied in order to limit the bandgap load current to a maximum of 100 nA. The internal reference can be disabled and overridden by an external reference by setting the CONFIG25 **extref\_ena** control bit. Capacitor  $C_{EXT}$  may hence be omitted. Terminal EXTIO thus serves as either input or output node.

The full-scale output current can be adjusted from 20 mA down to 2 mA by varying resistor  $R_{BIAS}$  or changing the externally applied reference voltage. The internal control amplifier has a wide input range, supporting the full-scale output current range of 20 dB.

## DAC TRANSFER FUNCTION

The CMOS DAC's consist of a segmented array of NMOS current sinks, capable of sinking a full-scale output current up to 20 mA. Differential current switches direct the current to either one of the complementary output nodes IOUT1 or IOUT2. (DACA = IOUTA1 or IOUTA2 and DACB = IOUTB1 or IOUTB2.) Complementary output currents enable differential operation, thus canceling out common mode noise sources (digital feed-through, on-chip and PCB noise), dc offsets, even order distortion components, and increasing signal output power by a factor of two.

The full-scale output current is set using external resistor  $R_{BIAS}$  in combination with an on-chip bandgap voltage reference source (+1.2V) and control amplifier. Current  $I_{BIAS}$  through resistor  $R_{BIAS}$  is mirrored internally to provide a maximum full-scale output current equal to 16 times  $I_{BIAS}$ .

The relation between IOUT1 and IOUT2 can be expressed as:

$$IOUT1 = -IOUT_{FS} - IOUT2$$

We will denote current flowing into a node as – current and current flowing out of a node as + current. Since the output stage is a current sink the current can only flow from AVDD into the IOUT1 and IOUT2 pins. The output current flow in each pin driving a resistive load can be expressed as:

$$IOUT1 = IOUT_{FS} \times (65535 - CODE) / 65536$$

$$IOUT2 = IOUT_{FS} \times CODE / 65536$$

where CODE is the decimal representation of the DAC data input word.

For the case where IOUT1 and IOUT2 drive resistor loads  $R_L$  directly, this translates into single ended voltages at IOUT1 and IOUT2:

$$VOUT1 = AVDD - |IOUT1| \times R_L$$

$$VOUT2 = AVDD - |IOUT2| \times R_L$$

Assuming that the data is full scale (65536 in offset binary notation) and the  $R_L$  is 25  $\Omega$ , the differential voltage between pins IOUT1 and IOUT2 can be expressed as:

$$VOUT1 = AVDD - |-0mA| \times 25 \Omega = 3.3 \text{ V}$$

$$VOUT2 = AVDD - |-20mA| \times 25 \Omega = 2.8 \text{ V}$$

$$VDIFF = VOUT1 - VOUT2 = 0.5V$$

Note that care should be taken not to exceed the compliance voltages at node IOUT1 and IOUT2, which would lead to increased signal distortion.

## ANALOG CURRENT OUTPUTS

Figure 53 shows a simplified schematic of the current source array output with corresponding switches. Differential switches direct the current of each individual NMOS current source to either the positive output node IOUT1 or its complementary negative output node IOUT2. The output impedance is determined by the stack of the current sources and differential switches, and is typically >300 kΩ in parallel with an output capacitance of 5 pF.

The external output resistors are referred to an external ground. The minimum output compliance at nodes IOUT1 and IOUT2 is limited to  $AVDD - 0.5\text{ V}$ , determined by the CMOS process. Beyond this value, transistor breakdown may occur resulting in reduced reliability of the DAC3282 device. The maximum output compliance voltage at nodes IOUT1 and IOUT2 equals  $AVDD + 0.5\text{ V}$ . Exceeding the minimum output compliance voltage adversely affects distortion performance and integral non-linearity. The optimum distortion performance for a single-ended or differential output is achieved when the maximum full-scale signal at IOUT1 and IOUT2 does not exceed 0.5 V.

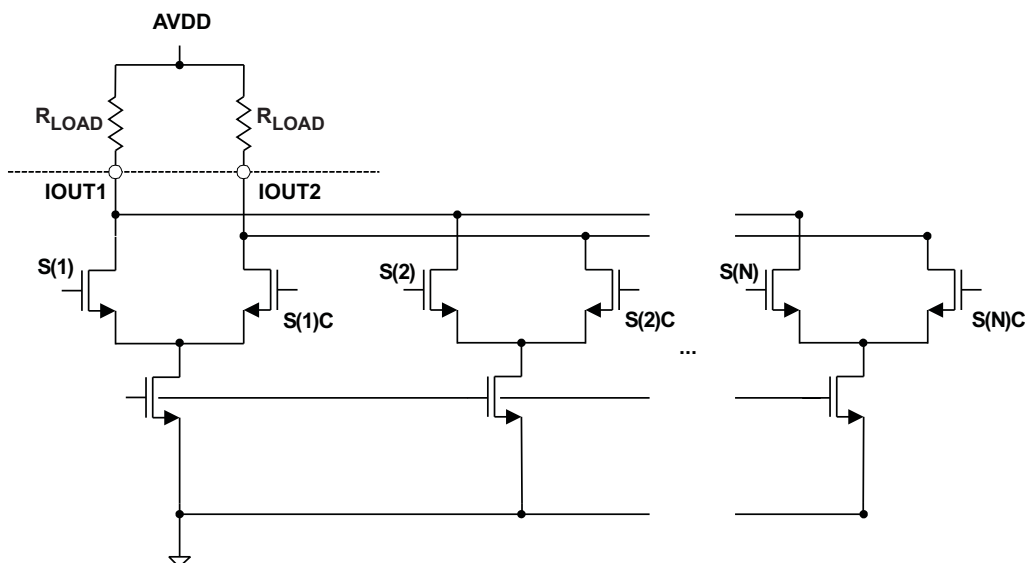


Figure 53. Equivalent Analog Current Output

The DAC3282 can be easily configured to drive a doubly terminated 50 Ω cable using a properly selected RF transformer. Figure 54 and Figure 55 show the 50 Ω doubly terminated transformer configuration with 1:1 and 4:1 impedance ratio, respectively. Note that the center tap of the primary input of the transformer has to be connected to AVDD to enable a DC current flow. Applying a 20 mA full-scale output current would lead to a 0.5 V<sub>pp</sub> for a 1:1 transformer and a 1 V<sub>pp</sub> output for a 4:1 transformer. The low dc-impedance between IOUT1 or IOUT2 and the transformer center tap sets the center of the ac-signal at AVDD, so the 1 V<sub>pp</sub> output for the 4:1 transformer results in an output between  $AVDD + 0.5\text{ V}$  and  $AVDD - 0.5\text{ V}$ .

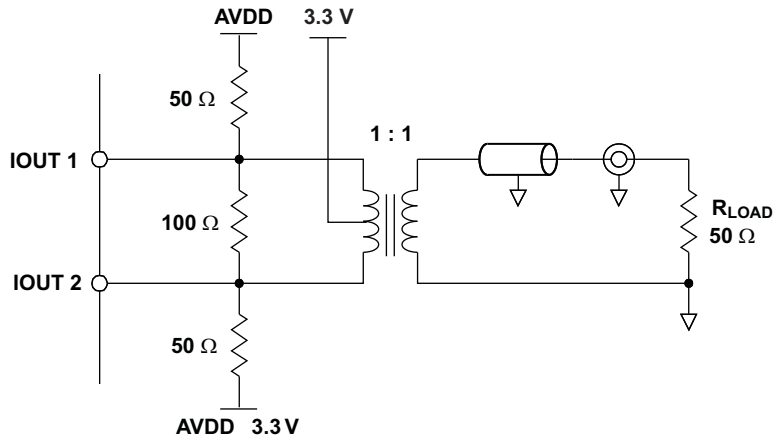


Figure 54. Driving a Doubly Terminated 50 Ω Cable Using a 1:1 Impedance Ratio Transformer

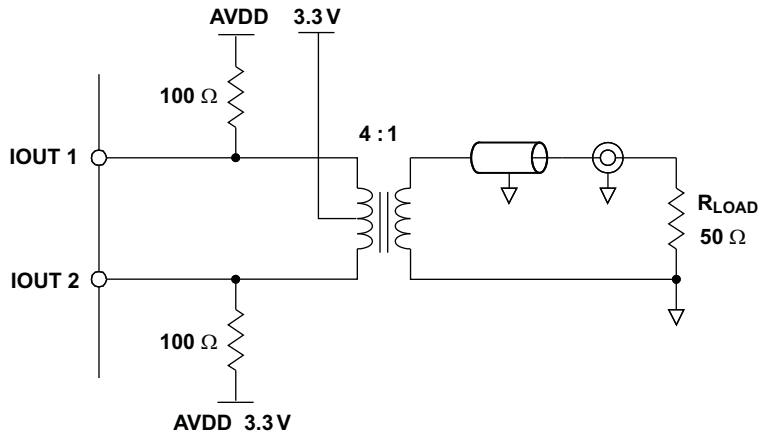


Figure 55. Driving a Doubly Terminated 50 Ω Cable Using a 4:1 Impedance Ratio Transformer



## PASSIVE INTERFACE TO ANALOG QUADRATURE MODULATORS

A common application in communication systems is to interface the DAC to an IQ modulator like the TRF3703 family of modulators from Texas Instruments. The input of the modulator is generally of high impedance and requires a specific common-mode voltage. A simple resistive network can be used to maintain 50Ω load impedance for the DAC3282 and also provide the necessary common-mode voltages for both the DAC and the modulator.

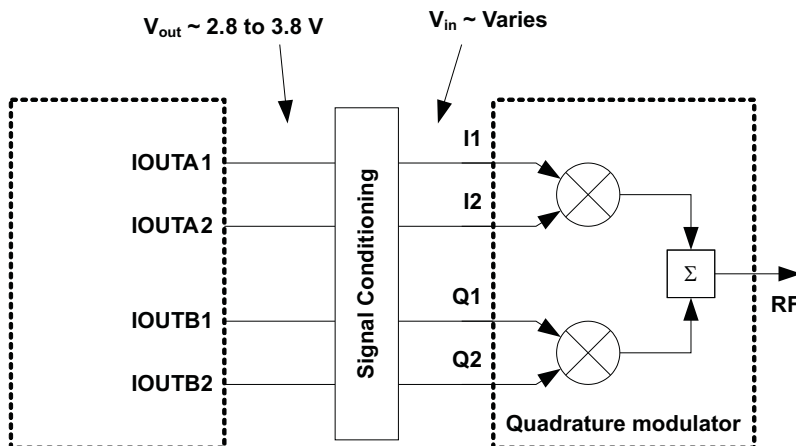


Figure 56. DAC to Analog Quadrature Modulator Interface

The DAC3282 has a maximum 20mA full-scale output and a voltage compliance range of  $AVDD \pm 0.5$  V. The TRF3703 IQ modulator family can be operated at three common-mode voltages: 1.5V, 1.7V, and 3.3V.

Figure 57 shows the recommended passive network to interface the DAC3282 to the TRF3703-17 which has a common mode voltage of 1.7V. The network generates the 3.3V common mode required by the DAC output and 1.7V at the modulator input, while still maintaining 50Ω load for the DAC.

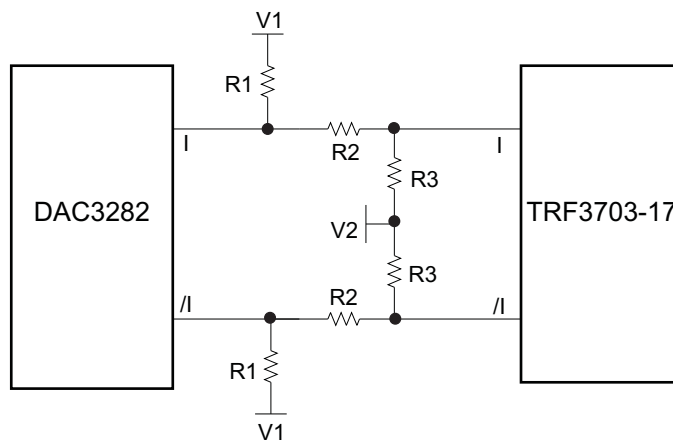
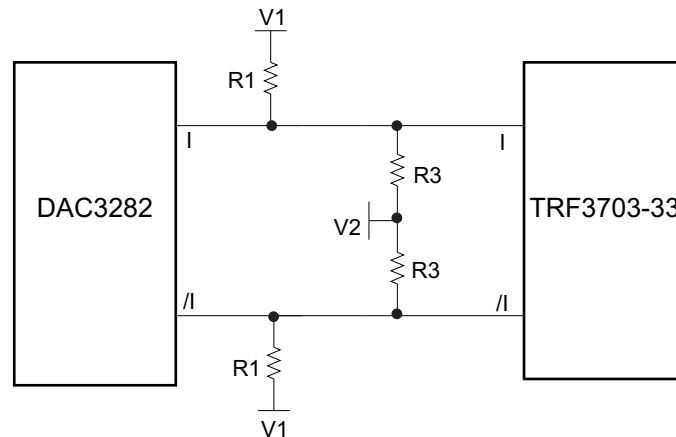


Figure 57. DAC3282 to TRF3703-17 Interface

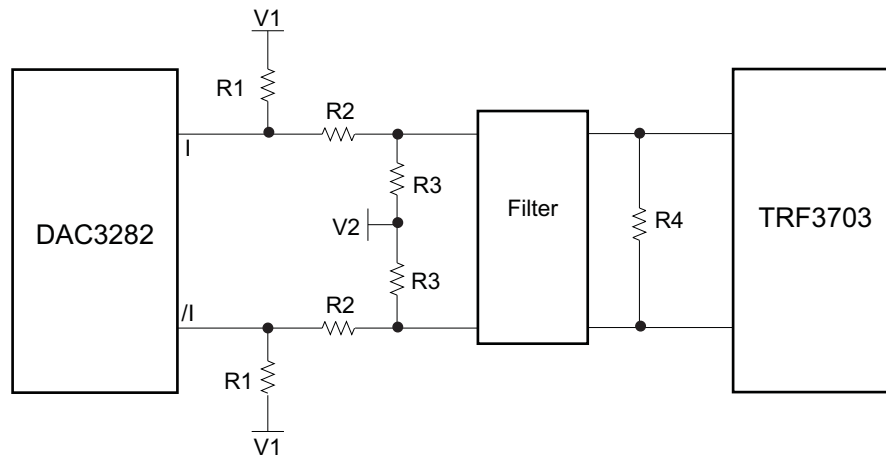
If  $V1$  is set to 5V and  $V2$  is set to -5V, the corresponding resistor values are  $R1 = 57\Omega$ ,  $R2 = 80\Omega$ , and  $R3 = 336\Omega$ . The loss developed through  $R2$  is about -1.86 dB. In the case where there is no -5V supply available and  $V2$  is set to 0V, the resistor values are  $R1 = 66\Omega$ ,  $R2 = 101\Omega$ , and  $R3 = 107\Omega$ . The loss with these values is -5.76dB.

Figure 58 shows the recommended network for interfacing with the TRF3703-33 which requires a common mode of 3.3V. This is the simplest interface as there is no voltage shift. Because there is no voltage shift there isn't any loss in the network. With  $V1 = 5$ V and  $V2 = 0$ V, the resistor values are  $R1 = 66\Omega$  and  $R3 = 208\Omega$ .



**Figure 58. DAC3282 to TRF3703-33 Interface**

In most applications a baseband filter is required between the DAC and the modulator to eliminate the DAC images. This filter can be placed after the common-mode biasing network. For the DAC to modulator network shown in [Figure 59](#), R2 and the filter load R4 need to be considered into the DAC impedance. The filter has to be designed for the source impedance created by the resistor combination of  $R3 \parallel (R2+R1)$ . The effective impedance seen by the DAC is affected by the filter termination resistor resulting in  $R1 \parallel (R2+R3 \parallel (R4/2))$ .



**Figure 59. DAC3282 to Modulator Interface with Filter**

Factoring in R4 into the DAC load, a typical interface to the TRF3703-17 with V1 = 5V and V2 = 0V results in the following values: R1 = 72Ω, R2 = 116Ω, R3 = 124Ω and R4 = 150Ω. This implies that the filter needs to be designed for 75Ω input and output impedance (single-ended impedance). The common mode levels for the DAC and modulator are maintained at 3.3V and 1.7V and the DAC load is 50Ω. The added load of the filter termination causes the signal to be attenuated by -10.8 dB.

A filter can be implemented in a similar manner to interface with the TRF3703-33. In this case it is much simpler to balance the loads and common mode voltages due to the absence of R2. An added benefit is that there is no loss in this network. With V1 = 5V and V2 = 0V the network can be designed such that R1 = 115Ω, R3 = 681Ω, and R4 = 200Ω. This results in a filter impedance of  $R1 \parallel R2 = 100\Omega$ , and a DAC load of  $R1 \parallel R3 \parallel (R4/2)$  which is equal to 50Ω. R4 is a differential resistor and does not affect the common mode level created by R1 and R3. The common-mode voltage is set at 3.3 V for a full-scale current of 20mA.

For more information on how to interface the DAC3282 to an analog quadrature modulator please refer to the application reports *Passive Terminations for Current Output DACs* ([SLAA399](#)) and *Design of Differential Filters for High-Speed Signal Chains* ([SLWA053](#)).

## APPLICATION EXAMPLE

### DIRECT CONVERSION RADIO

Refer to [Figure 60](#) for an example Direct Conversion Radio. The DAC3282 receives an interleaved complex I/Q baseband input data stream and increases the sample rate through interpolation by a factor of 2. By performing digital interpolation on the input data, undesired images of the original signal can be pushed out of the band of interest and more easily suppressed with analog filters.

For a Zero IF (ZIF) frequency plan, complex mixing of the baseband signal is not required. Alternatively, for a Complex IF frequency plan the input data can be pre-placed at an IF within the bandwidth limitations of the interpolation filters. In addition, complex mixing is available using the coarse mixer block to up-convert the signal. The output of both DAC channels is used to produce a Hilbert transform pair and can be expressed as:

$$\begin{aligned} A_{OUT}(t) &= A(t)\cos(\omega_c t) - B(t)\sin(\omega_c t) = m(t) \\ B_{OUT}(t) &= A(t)\sin(\omega_c t) + B(t)\cos(\omega_c t) = m_h(t) \end{aligned}$$

where  $m(t)$  and  $m_h(t)$  connote a Hilbert transform pair and  $\omega_c$  is the mixer frequency. The complex output is input to an analog quadrature modulator (AQM) such as the Texas Instruments TRF3720 for a single side-band (SSB) up conversion to RF. A passive (resistor only) interface to the AQM with an optional LC filter network is recommended. The TRF3720 includes a VCO/PLL to generate the LO frequency. Upper single-sideband upconversion is achieved at the output of the analog quadrature modulator, whose output is expressed as:

$$RF(t) = A(t)\cos(\omega_c + \omega_{LO})t - B(t)\sin(\omega_c + \omega_{LO})t$$

Flexibility is provided to the user by allowing for the selection of negative mixing frequency to produce a lower-sideband upconversion. Note that the process of complex mixing translates the signal frequency from 0Hz means that the analog quadrature modulator IQ imbalance produces a sideband that falls outside the signal of interest. DC offset error in DAC and AQM signal path may produce LO feed-through at the RF output which may fall in the band of interest. To suppress the LO feed-through, the DAC3282 provides a digital offset correction capability for both DAC-A and DAC-B paths.

The complex IF architecture has several advantages over the real IF architecture:

- Uncalibrated side-band suppression ~ 35dBc compared to 0dBc for real IF architecture.
- Direct DAC to AQM interface – no amplifiers required
- DAC 2nd Nyquist zone image is offset  $f_{DAC}$  compared with  $f_{DAC} - 2 \times IF$  for a real IF architecture, reducing the need for filtering at the DAC output.
- Uncalibrated LO feed through for AQM is ~ 35dBc and calibration can reduce or completely remove the LO feed through.

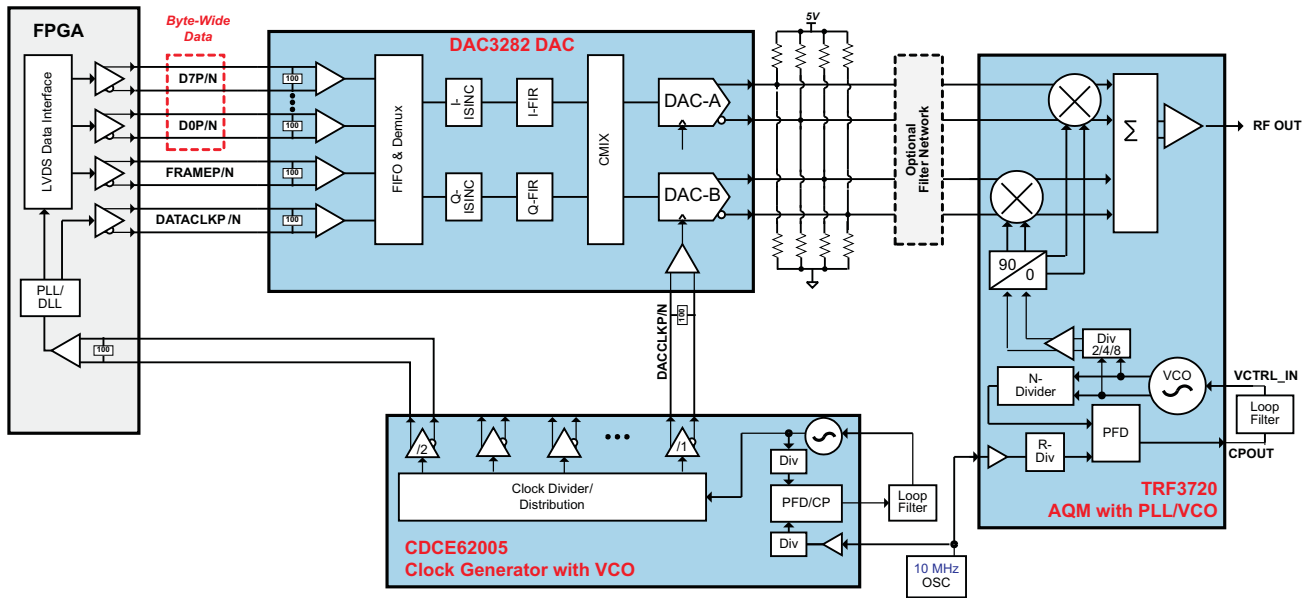


Figure 60. System Diagram of Direct Conversion Radio

## REVISION HISTORY

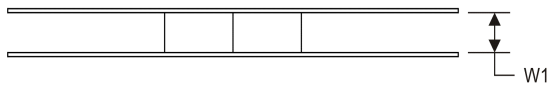
NOTE: Page numbers of current version may differ from previous versions

<b>Changes from Original (December 2009) to Revision A</b>	<b>Page</b>
• Deleted FIFO_OSTRP and FIFO_OSTRN descriptions from Pin Functions table. N/A for this device. ....	4
• Changed Default from 0x41 to 0x43 for Register name <b>VERSION31</b> in <a href="#">Table 3</a> Register Map .....	22
• Changed Default address from 0x41 to 0x43 for Register name: <b>VERSION31</b> ; and Default Value for Bit 5:0 from 000001 to 000011. ....	29

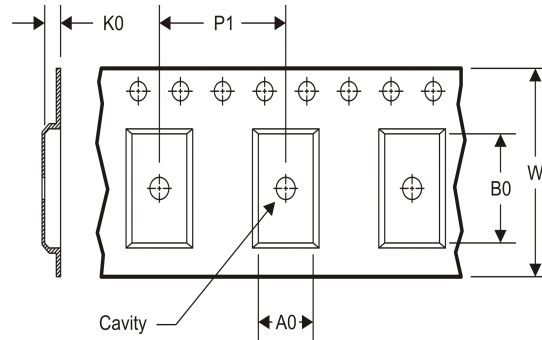
<b>Changes from Revision A (February 2010) to Revision B</b>	<b>Page</b>
• Added to Description of the PIN FUNCTIONS table pin no.31, FROM: Bi-directional in 3-pin....(default). TO: Bi-directional in 3-pin....(default) and 4-pin mode. ....	4
• Deleted SNR definition and added: Noise Spectral...Nyquist zone. ....	19
• Changed from "SDIO is data in only" to "SDIO is bidirectional" in SERIAL INTERFACE section first paragraph, .....	20
• Changed FROM: "In 4 pin....cycle(s)." TO: "In 4 pin configuration, both ALARM_SDO and SDIO are data out from DAC3282." in paragraph under Figure 32. ....	21
• Changed Bit 5 function to: Allows the FRAME input to reset the FIFO write pointer when asserted. AND changed Bit 4 first sentence to: "Allows the FRAME or OSTR signal to reset the FIFO read pointer when asserted." in CONFIG0 Register description .....	23
• Changed CONFIG3 Register table from: "CONFIG1, 0x01 to CONFIG3, 0x03" and in Bit 4:2 Function From: "When the FIFO.....read pointer." TO: "This is the default FIFO read pointer position after the FIFO read pointer has been synchronized." .....	24
• Changed CONFIG7 register table, Bit 6 Function description "This alarm indicates.....more detail." .....	25
• Added text string to CONFIG18 Register table, Bit 1 Function description for clarification. ....	26
• Changed FIFO block diagram <a href="#">Figure 35</a> .....	31
• Added 4th & 5th paragraphs to the INPUT FIFO section, under <a href="#">Figure 35</a> . ....	31
• Changed <a href="#">Figure 36</a> .....	32
• Added "CONFIG19 multi_sync_sel" to FIFO MODES OF OPERATION section .....	33
• Changed Mode descriptions in FIFO Operation Modes, <a href="#">Table 4</a> , from "Enabled" to "Single Sync Source" and added new FIFO Mode, "Dual Sync Sources" .....	33
• Moved the MULTI-DEVICE SYNCHRONIZATION section to follow "Bypass Mode" section. ....	33
• Changed the illustration for <a href="#">Figure 37</a> . ....	34
• Changed the illustration for <a href="#">Figure 39</a> .....	35
• Changed the "DATA PATTERN CHECKER" section text for clarification. ....	35
• Changed the POWER-UP-SEQUENCE section for clarification. ....	43

**TAPE AND REEL INFORMATION**

**REEL DIMENSIONS**



**TAPE DIMENSIONS**



A0	Dimension designed to accommodate the component width
B0	Dimension designed to accommodate the component length
K0	Dimension designed to accommodate the component thickness
W	Overall width of the carrier tape
P1	Pitch between successive cavity centers

**TAPE AND REEL INFORMATION**

\*All dimensions are nominal

Device	Package Type	Package Drawing	Pins	SPQ	Reel Diameter (mm)	Reel Width W1 (mm)	A0 (mm)	B0 (mm)	K0 (mm)	P1 (mm)	W (mm)	Pin1 Quadrant
DAC3282IRGZR	VQFN	RGZ	48	2500	330.0	16.4	7.3	7.3	1.5	12.0	16.0	Q2
DAC3282IRGZT	VQFN	RGZ	48	250	330.0	16.4	7.3	7.3	1.5	12.0	16.0	Q2

TAPE AND REEL BOX DIMENSIONS

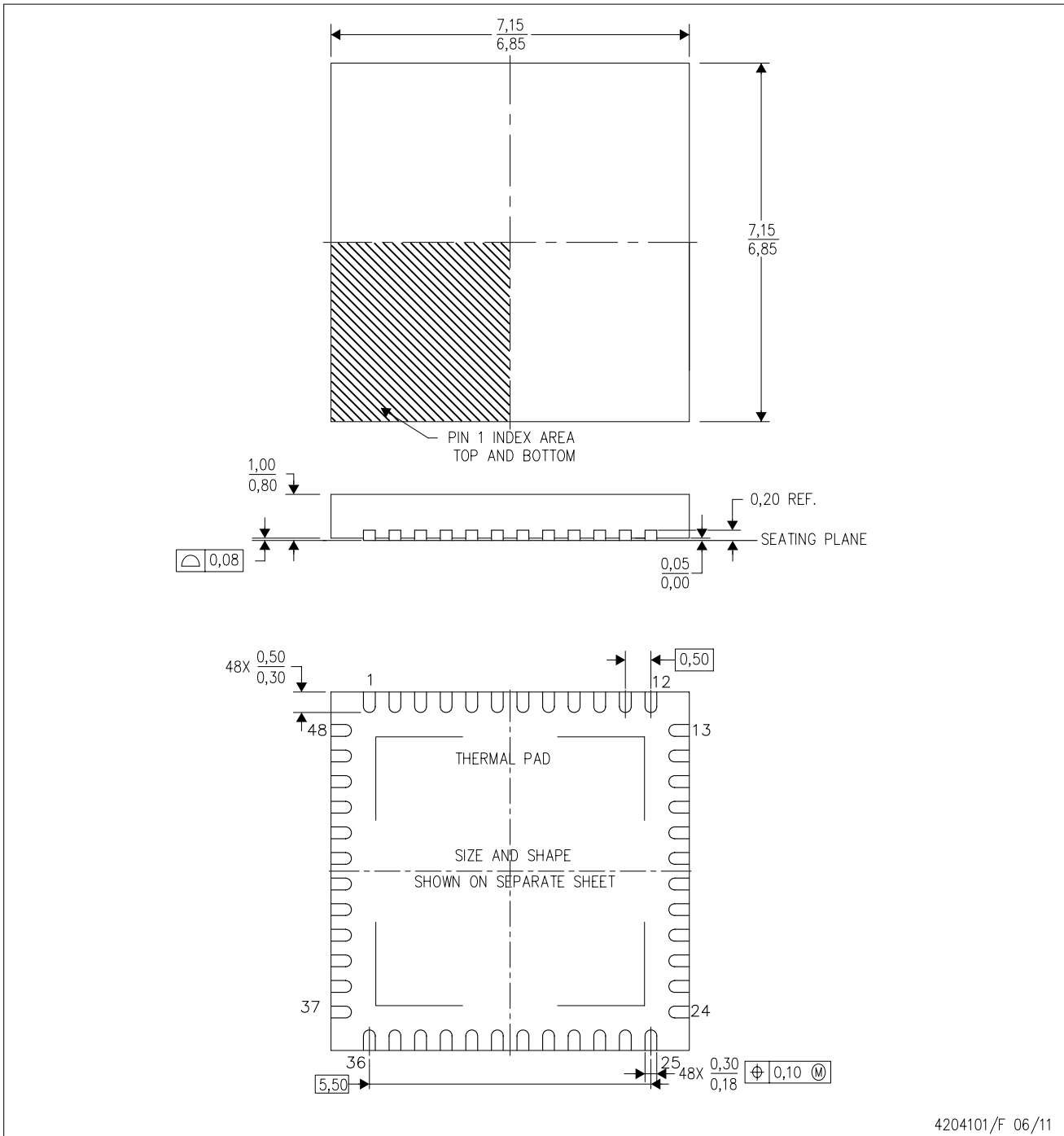


\*All dimensions are nominal

Device	Package Type	Package Drawing	Pins	SPQ	Length (mm)	Width (mm)	Height (mm)
DAC3282IRGZR	VQFN	RGZ	48	2500	336.6	336.6	28.6
DAC3282IRGZT	VQFN	RGZ	48	250	336.6	336.6	28.6

RGZ (S-PVQFN-N48)

PLASTIC QUAD FLATPACK NO-LEAD



4204101/F 06/11

- NOTES:
- A. All linear dimensions are in millimeters. Dimensioning and tolerancing per ASME Y14.5M-1994.
  - B. This drawing is subject to change without notice.
  - C. Quad Flatpack, No-leads (QFN) package configuration.
  - D. The package thermal pad must be soldered to the board for thermal and mechanical performance.
  - E. See the additional figure in the Product Data Sheet for details regarding the exposed thermal pad features and dimensions.
  - F. Falls within JEDEC MO-220.

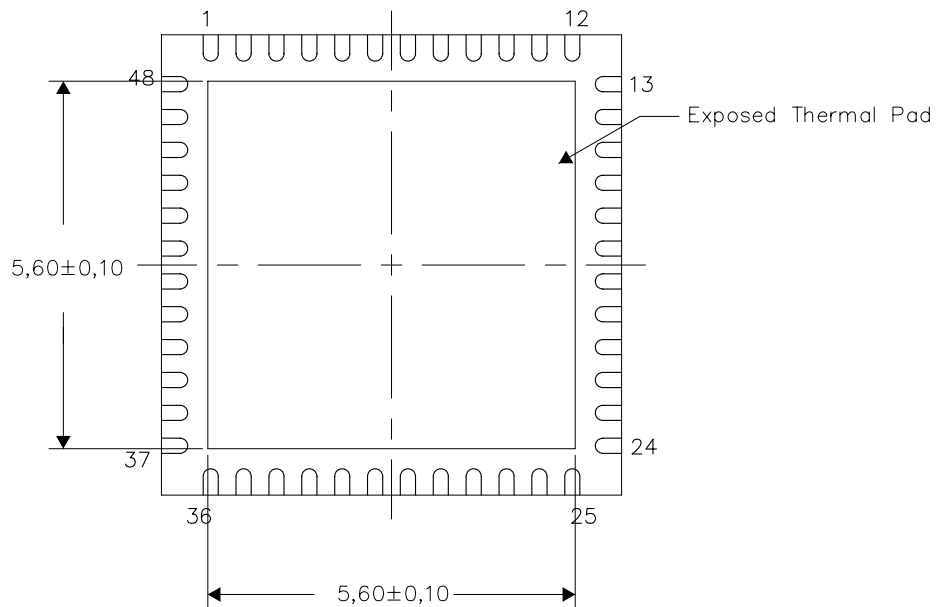


## THERMAL INFORMATION

This package incorporates an exposed thermal pad that is designed to be attached directly to an external heatsink. The thermal pad must be soldered directly to the printed circuit board (PCB). After soldering, the PCB can be used as a heatsink. In addition, through the use of thermal vias, the thermal pad can be attached directly to the appropriate copper plane shown in the electrical schematic for the device, or alternatively, can be attached to a special heatsink structure designed into the PCB. This design optimizes the heat transfer from the integrated circuit (IC).

For information on the Quad Flatpack No-Lead (QFN) package and its advantages, refer to Application Report, QFN/SON PCB Attachment, Texas Instruments Literature No. SLUA271. This document is available at [www.ti.com](http://www.ti.com).

The exposed thermal pad dimensions for this package are shown in the following illustration.



Bottom View

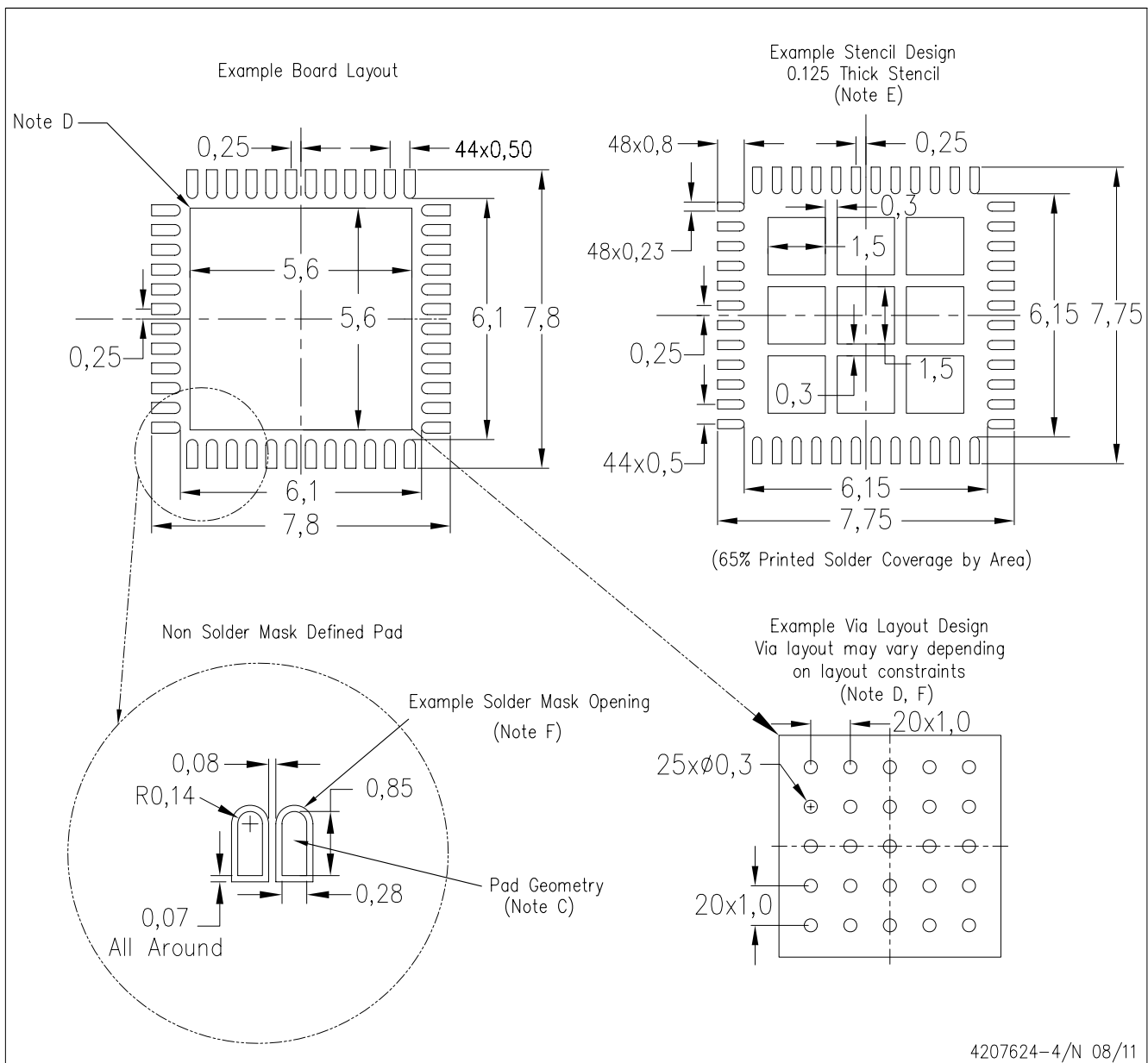
Exposed Thermal Pad Dimensions

4206354-5/R 08/11

NOTE: All linear dimensions are in millimeters

RGZ (S-PVQFN-N48)

PLASTIC QUAD FLATPACK NO-LEAD



- NOTES:
- All linear dimensions are in millimeters.
  - This drawing is subject to change without notice.
  - Publication IPC-7351 is recommended for alternate designs.
  - This package is designed to be soldered to a thermal pad on the board. Refer to Application Note, Quad Flat-Pack Packages, Texas Instruments Literature No. SLUA271, and also the Product Data Sheets for specific thermal information, via requirements, and recommended board layout. These documents are available at [www.ti.com](http://www.ti.com) <<http://www.ti.com>>.
  - Laser cutting apertures with trapezoidal walls and also rounding corners will offer better paste release. Customers should contact their board assembly site for stencil design recommendations. Refer to IPC 7525 for stencil design considerations.
  - Customers should contact their board fabrication site for recommended solder mask tolerances and via tenting recommendations for vias placed in the thermal pad.

## IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, modifications, enhancements, improvements, and other changes to its products and services at any time and to discontinue any product or service without notice. Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All products are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its hardware products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by government requirements, testing of all parameters of each product is not necessarily performed.

TI assumes no liability for applications assistance or customer product design. Customers are responsible for their products and applications using TI components. To minimize the risks associated with customer products and applications, customers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any TI patent right, copyright, mask work right, or other TI intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information published by TI regarding third-party products or services does not constitute a license from TI to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of TI information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. Reproduction of this information with alteration is an unfair and deceptive business practice. TI is not responsible or liable for such altered documentation. Information of third parties may be subject to additional restrictions.

Resale of TI products or services with statements different from or beyond the parameters stated by TI for that product or service voids all express and any implied warranties for the associated TI product or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

TI products are not authorized for use in safety-critical applications (such as life support) where a failure of the TI product would reasonably be expected to cause severe personal injury or death, unless officers of the parties have executed an agreement specifically governing such use. Buyers represent that they have all necessary expertise in the safety and regulatory ramifications of their applications, and acknowledge and agree that they are solely responsible for all legal, regulatory and safety-related requirements concerning their products and any use of TI products in such safety-critical applications, notwithstanding any applications-related information or support that may be provided by TI. Further, Buyers must fully indemnify TI and its representatives against any damages arising out of the use of TI products in such safety-critical applications.

TI products are neither designed nor intended for use in military/aerospace applications or environments unless the TI products are specifically designated by TI as military-grade or "enhanced plastic." Only products designated by TI as military-grade meet military specifications. Buyers acknowledge and agree that any such use of TI products which TI has not designated as military-grade is solely at the Buyer's risk, and that they are solely responsible for compliance with all legal and regulatory requirements in connection with such use.

TI products are neither designed nor intended for use in automotive applications or environments unless the specific TI products are designated by TI as compliant with ISO/TS 16949 requirements. Buyers acknowledge and agree that, if they use any non-designated products in automotive applications, TI will not be responsible for any failure to meet such requirements.

Following are URLs where you can obtain information on other Texas Instruments products and application solutions:

### Products

Audio	<a href="http://www.ti.com/audio">www.ti.com/audio</a>
Amplifiers	<a href="http://amplifier.ti.com">amplifier.ti.com</a>
Data Converters	<a href="http://dataconverter.ti.com">dataconverter.ti.com</a>
DLP® Products	<a href="http://www.dlp.com">www.dlp.com</a>
DSP	<a href="http://dsp.ti.com">dsp.ti.com</a>
Clocks and Timers	<a href="http://www.ti.com/clocks">www.ti.com/clocks</a>
Interface	<a href="http://interface.ti.com">interface.ti.com</a>
Logic	<a href="http://logic.ti.com">logic.ti.com</a>
Power Mgmt	<a href="http://power.ti.com">power.ti.com</a>
Microcontrollers	<a href="http://microcontroller.ti.com">microcontroller.ti.com</a>
RFID	<a href="http://www.ti-rfid.com">www.ti-rfid.com</a>
OMAP Mobile Processors	<a href="http://www.ti.com/omap">www.ti.com/omap</a>
Wireless Connectivity	<a href="http://www.ti.com/wirelessconnectivity">www.ti.com/wirelessconnectivity</a>

### Applications

Automotive and Transportation	<a href="http://www.ti.com/automotive">www.ti.com/automotive</a>
Communications and Telecom	<a href="http://www.ti.com/communications">www.ti.com/communications</a>
Computers and Peripherals	<a href="http://www.ti.com/computers">www.ti.com/computers</a>
Consumer Electronics	<a href="http://www.ti.com/consumer-apps">www.ti.com/consumer-apps</a>
Energy and Lighting	<a href="http://www.ti.com/energy">www.ti.com/energy</a>
Industrial	<a href="http://www.ti.com/industrial">www.ti.com/industrial</a>
Medical	<a href="http://www.ti.com/medical">www.ti.com/medical</a>
Security	<a href="http://www.ti.com/security">www.ti.com/security</a>
Space, Avionics and Defense	<a href="http://www.ti.com/space-avionics-defense">www.ti.com/space-avionics-defense</a>
Video and Imaging	<a href="http://www.ti.com/video">www.ti.com/video</a>

TI E2E Community Home Page

[e2e.ti.com](http://e2e.ti.com)

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265  
Copyright © 2012, Texas Instruments Incorporated