ADS125P08 8/16-Channel, 1MSPS, 24-Bit, Delta-Sigma ADC

1 Features

- Programmable data rate up to 1.067MSPS
- Analog multiplexer with 17 independently selectable inputs:
  - Up to 8 fully differential inputs
  - Up to 16 single ended inputs
- Rail-to-rail analog input buffer and reference buffer
- Internal voltage reference, selectable output 2.5V or 4.096V
- Fault detection and monitor circuits
- General-Purpose I/Os
- Internal Oscillator: 25.6MHz, 1% Accuracy
- Channel auto-sequencer and FIFO buffer
- Simultaneous 50Hz and 60Hz rejection at ≤ 25 SPS with low-latency digital filter
- Power-scalable architecture with four speed modes

2 Applications

- Factory automation and control:
  - Condition monitoring
  - Analog input modules
  - Field transmitter: Flow meter
- Patient Monitoring:
  - ECG
  - Infusion pump
- Test and measurement:
  - Data acquisition (DAQ)
  - Semiconductor test equipment

3 Description

The ADS125P08 is a multiplexed 8/16-Channel, 24 bit, delta-sigma (ΔΣ), analog-to-digital converter (ADC) with data rates up to 1MSPS and is configurable to accept up to 8 fully differential analog inputs or up to 16 single-ended analog inputs. The device offers an excellent combination of ac performance and dc precision with low power consumption.

The device integrates high-impedance input and reference buffers to reduce signal loading. The ADS125P08 is equipped with a channel auto-sequencer and a FIFO (first-in, first-out) buffer. The power-scalable architecture provides four speed modes to optimize data rate, resolution, and power consumption.

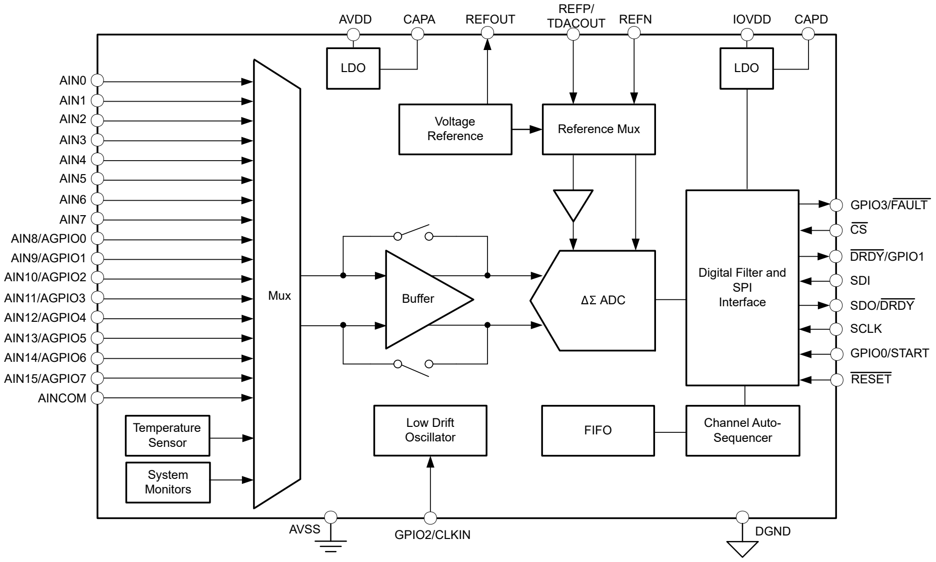
Input and output data and register settings are validated by a cyclic-redundancy check (CRC) feature to enhance operational reliability.

The small 5mm × 5mm VQFN package is designed for limited space applications. The device is fully specified for operation over the –40°C to +125°C temperature range.

Packaging Information

PART NUMBER	PACKAGE[1]	PACKAGE SIZE[2]
ADS125P08	RHB (VQFN, 36)	5.00mm × 5.00mm

[1] For more information, see Section 11.
[2] The package size (length × width) is a nominal value and includes pins, where applicable.



Simplified Block Diagram

An IMPORTANT NOTICE at the end of this data sheet addresses availability, warranty, changes, use in safety-critical applications, intellectual property matters and other important disclaimers. PRODUCTION DATA.

Table of Contents

4 Pin Configuration and Functions

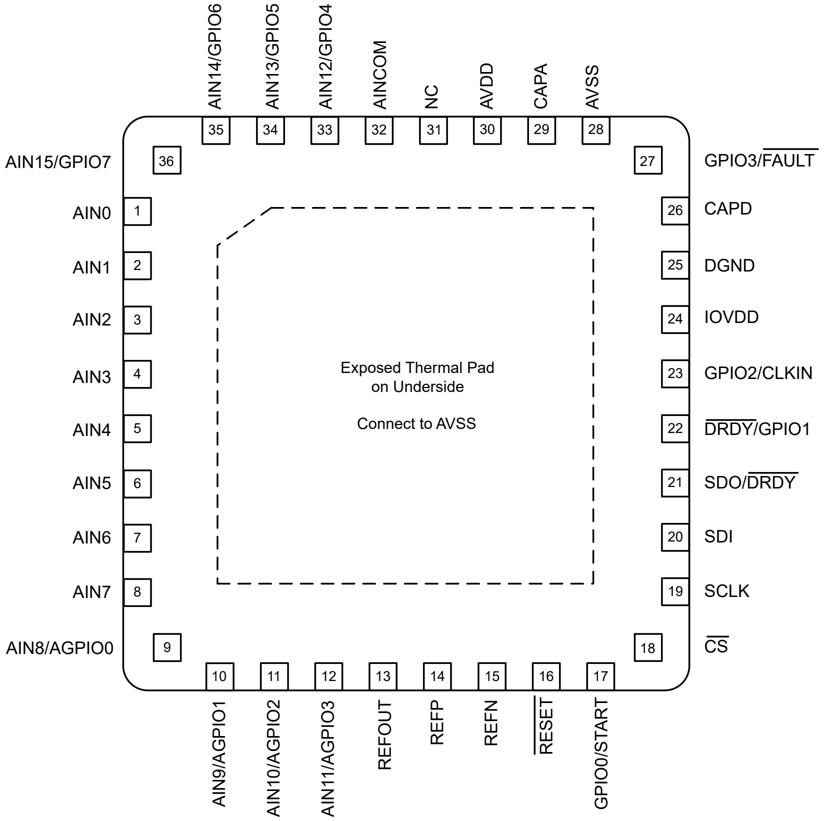


Figure 4-1. RHB Package, 36-Pin VQFN (Top View)

Table 4-1. Pin Functions

NAME	NO.	TYPE	DESCRIPTION
AIN0	1	Analog Input	Analog input 0
AIN1	2	Analog Input	Analog input 1
AIN2	3	Analog Input	Analog input 2
AIN3	4	Analog Input	Analog input 3
AIN4	5	Analog Input	Analog input 4
AIN5	6	Analog Input	Analog input 5
AIN6	7	Analog Input	Analog input 6
AIN7	8	Analog Input	Analog input 7
AIN8/AGPIO0	9	Analog Input	Analog input 8/General-purpose input/output 0
AIN9/AGPIO1	10	Analog Input	Analog input 9/General-purpose input/output 1
AIN10/AGPIO2	11	Analog Input	Analog input 10/General-purpose input/output 2
AIN11/AGPIO3	12	Analog Input	Analog input 11/General-purpose input/output 3
AIN12/AGPIO4	33	Analog Input	Analog input 12/General-purpose input/output 4
AIN13/AGPIO5	34	Analog Input	Analog input 13/General-purpose input/output 5
AIN14/AGPIO6	35	Analog Input	Analog input 14/General-purpose input/output 6
AIN15/AGPIO7	36	Analog Input	Analog input 15/General-purpose input/output 7
AINCOM	32	Analog Input	Analog input common. Voltage inputs are referenced to AINCOM when the inputs are configured as single-ended. Connect VINCOM to analog ground.
AVDD	30	Analog Supply	Positive analog supply
AVSS	28	Analog Supply	Negative analog supply
CAPA	29	Analog Supply	Analog voltage regulator output external bypass
CAPD	26	Analog Supply	Digital voltage regulator output external bypass

Table 4-1. Pin Functions (continued)

NAME	NO.	TYPE	DESCRIPTION
IOVDD	24	Digital Supply	I/O supply voltage
DGND	25	Ground	Digital ground
REFOUT	13	Analog Output	Voltage reference output
REFP/TDACOUT	14	Analog Input/Output	Positive reference input/Test DAC output
REFN	15	Analog Input	Negative reference input
$\overline{\text{RESET}}$	16	Digital Input	Reset, active low
GPIO0/START	17	Digital Input	General-purpose input/output 0. Pin can be configured as dedicated START input for conversion control.
$\overline{\text{CS}}$	18	Digital Input	Chip select, active low
SCLK	19	Digital Input	Serial data clock
SDI	20	Digital Input	Serial data input
SDO/$\overline{\text{DRDY}}$	21	Digital IO	Serial data output and data ready, active low (optional)
$\overline{\text{DRDY}}$/GPIO1	22	Digital IO	Data ready, active low/General-purpose input/output 1
GPIO2/CLKIN	23	Digital IO	General-purpose input/output 2. Pin can be configured as external clock input.
GPIO3/$\overline{\text{FAULT}}$	27	Digital IO	General-purpose input/output 3. Pin can be configured as dedicated $\overline{\text{FAULT}}$ output.
NC	31	-	No connect, leave unconnected [1]
Thermal Pad		N/A	Thermal power pad; connect to AVSS

[1] See the *Unused Inputs and Outputs* section for details on how to connect unused pins.

5 Specifications

5.1 Absolute Maximum Ratings

over operating ambient temperature range (unless otherwise noted) [1]

		MIN	MAX	UNIT
Power supply voltage	AVDD to AVSS	–0.3	6.5	V
	AVSS to DGND	–3	0.3	
	IOVDD to DGND	–0.3	6.5	
	IOVDD to AVSS		8.5	
Analog input voltage	AINx to AVSS	AVSS – 0.3	AVDD + 0.3	V
Reference input voltage	REFP, REFN	AVSS – 0.3	AVDD + 0.3	V
Analog output voltage	CAPA	AVSS	1.65	V
	CAPD	DGND	1.65	
Digital input/output voltage	$\overline{\text{RESET}}$, GPIO0/START, SDO/$\overline{\text{DRDY}}$, $\overline{\text{DRDY}}$/GPIO1, GPIO2/ CLKIN, GPIO3/$\overline{\text{FAULT}}$	DGND – 0.3	IOVDD + 0.3	V
	$\overline{\text{CS}}$, SCLK, SDI	DGND – 0.3	6.5	V
Input current	Continuous, any pin except power-supply pins[2]	–10	10	mA
Temperature	Junction, T_J		150	°C
	Storage, T_{stg}	–65	150	

[1] Operation outside the Absolute Maximum Ratings may cause permanent device damage. Absolute Maximum Ratings do not imply functional operation of the device at these or any other conditions beyond those listed under Recommended Operating Conditions. If briefly operating outside the Recommended Operating Conditions but within the Absolute Maximum Ratings, the device may not sustain damage, but it may not be fully functional – this may affect device reliability, functionality, performance, and shorten the device lifetime.

[2] Analog input pins AINPx, AINNx, REFP, and REFN are diode-clamped to AVDD and AVSS. Limit the input current to 10 mA in the event the analog input voltage is ≥ AVDD + 0.3 V or ≤ AVSS – 0.3 V. Digital I/O pins are diode-clamped to DGND only. Limit the input current to 10 mA in the event the digital pin voltage is below DGND – 0.3 V.

5.2 ESD Ratings

			VALUE	UNIT
$V_{(ESD)}$	Electrostatic discharge	Human-body model (HBM), per ANSI/ESDA/JEDEC JS-001[1]	±1000	V
		Charged-device model (CDM), per ANSI/ESDA/JEDEC JS-002[2]	±750	

[1] JEDEC document JEP155 states that 500-V HBM allows safe manufacturing with a standard ESD control process.

[2] JEDEC document JEP157 states that 250-V CDM allows safe manufacturing with a standard ESD control process.

5.3 Recommended Operating Conditions

over operating ambient temperature range (unless otherwise noted)

				MIN	NOM	MAX	UNIT
POWER SUPPLY							
	Analog power supply	AVDD to AVSS	Speed mode 3	4.5		5.5	V
			Speed mode 2	4.5		5.5	
			Speed mode 1	3		5.5	
			Speed mode 0	3		5.5	
		AVDD to DGND		1.65			V
		AVSS to DGND		–2.75		0	V
	Digital power supply	IOVDD to DGND		1.65		5.5	V
ANALOG INPUTS							
V_{AINP}, V_{AINN}	Absolute input voltage	Input buffer off		AVSS + 0.1		AVDD – 0.1	V
		Input buffer on		AVSS – 0.05		AVDD – 0.2	
V_{IN}	Differential input voltage [1] $V_{IN} = V_{AINP} – V_{AINN}$			$–V_{REF}$		V_{REF}	V
VOLTAGE REFERENCE INPUTS							
V_{REF}	Differential reference voltage $V_{REF} = V_{REFP} – V_{REFN}$			1	4.096	AVDD – AVSS	V
V_{REFN}	Negative reference voltage			AVSS – 0.05	AVSS		V
V_{REFP}	Positive reference voltage	REFP buffer off				AVDD + 0.05	V
		REFP buffer on				AVDD – 0.7	
EXTERNAL CLOCK SOURCE							
f_{CLK}	Clock frequency	Speed mode 3		0.5	25.6	26.2	MHz
		Speed mode 2		0.5	12.8	13.1	
		Speed mode 1		0.5	3.2	3.28	
		Speed mode 0		0.5	1.6	1.64	
DIGITAL INPUTS							
V_{IL}	Logic low input voltage			0		0.3 × IOVDD	V
V_{IH}	Logic high input voltage			0.7 × IOVDD		IOVDD	V
I_{LEAK}	External leakage current	Tri-state pins, floating input		-5		5	µA
TEMPERATURE RANGE							
T_A	Ambient temperature	Operational		–45		125	°C
		Specification		–40		125	

[1] AIN_P and AIN_N denote the positive and negative inputs of the input buffer/ADC. Any of the available analog inputs (AINx) can be selected as either AIN_P or AIN_N by the input multiplexer.

5.4 Thermal Information

THERMAL METRIC [1]		VQFN (RSH)	UNIT
		36 PINS	
$R_{θJA}$	Junction-to-ambient thermal resistance	32.5	°C/W
$R_{θJC(top)}$	Junction-to-case (top) thermal resistance	11.5	°C/W
$R_{θJB}$	Junction-to-board thermal resistance	6.3	°C/W
$Ψ_{JT}$	Junction-to-top characterization parameter	0.1	°C/W
$Ψ_{JB}$	Junction-to-board characterization parameter	6.2	°C/W
$R_{θJC(bot)}$	Junction-to-case (bottom) thermal resistance	1.1	°C/W

[1] For more information about traditional and new thermal metrics, see the Semiconductor and IC Package Thermal Metrics application note.

5.5 Electrical Characteristics

minimum and maximum specifications apply from T_A = –40°C to +125°C; typical specifications are at T_A = 25°C; all specifications are at AVDD - AVSS = 5 V, IOVDD = 1.8 V, V_{IN} = 0 V, V_{CM} = (AVDD - AVSS)/2, (V_{REFP} – $V_{REFN)}$ = 2.5 V, all speed modes, input buffers on, external clock, and reference buffer on (unless otherwise noted)

PARAMETER		TEST CONDITIONS		MIN	TYP	MAX	UNIT
ANALOG INPUTS, SPEED MODE 3							
	Input current, differential input voltage	VCM=AVDD/2	Input buffers off		±3		µA
			Input buffers on		±7		nA
	Input current drift, differential input voltage	VCM=AVDD/2	Input buffers off		5		nA/°C
			Input buffers on		0.07		nA/°C
	Input current, common-mode input voltage	VN=VP	Input buffers off		±3		µA
			Input buffers on		±7		nA
ANALOG INPUTS, SPEED MODE 2							
	Input current, differential input voltage	VCM=AVDD/2	Input buffers off		±1.5		µA
			Input buffers on		±2		nA
	Input current drift, differential input voltage	VCM=AVDD/2	Input buffers off		5		nA/°C
			Input buffers on		0.02		nA/°C
	Input current, common-mode input voltage	VN=VP	Input buffers off		±1.5		µA
			Input buffers on		±3		nA
ANALOG INPUTS, SPEED MODE 1							
	Input current, differential input voltage	VCM=AVDD/2	Input buffers off		±0.4		µA
			Input buffers on		±2		nA
	Input current drift, differential input voltage	VCM=AVDD/2	Input buffers off		5		nA/°C
			Input buffers on		0.02		nA/°C
	Input current, common-mode input voltage	VN=VP	Input buffers off		±0.4		µA
			Input buffers on		±2		nA
ANALOG INPUTS, SPEED MODE 0							
	Input current, differential input voltage	VCM=AVDD/2	Input buffers off		±0.4		µA
			Input buffers on		±1		nA
	Input current drift, differential input voltage	VCM=AVDD/2	Input buffers off		0.2		nA/°C
			Input buffers on		0.01		nA/°C
	Input current, common-mode input voltage	VN=VP	Input buffers off		±0.4		µA
			Input buffers on		±1		nA
DC PERFORMANCE							
	Resolution	OSR ≥ 32			24		Bits
	Noise			See the *Noise Performance* section for details			
INL	Integral nonlinearity	T_A = -40°C to 125°C	Speed mode 3, buffer on		2	7	ppm of FSR
			Speed mode 3, buffer off		6	20	
			All other speed modes		1	4	
	Offset error	T_A = 25°C, external short		–1000	±100	1000	µV
	Offset drift				100	850	nV/°C
	Gain error	T_A = 25°, including internal reference error, V_{REFP} = 2.5V or 4.096V		–1000	±200	1000	ppm of FSR
		T_A = 25°, excluding reference error		–150	±50	150	
	Gain drift	Including internal reference error, V_{REFP} = 2.5V or 4.096V			5	13	ppm of FSR/°C
		Excluding reference error			0.75	2	
TUE	Total Unadjusted Error	Internal reference V_{REFP} = 2.5 V			0.01	0.06	% FSR
		Internal reference V_{REFP} = 4.096 V			0.02	0.09	
		External reference V_{REFP} = 2.5V or 4.096 V			0.01	0.03	
SYSTEM PERFORMANCE							

minimum and maximum specifications apply from T_A = –40°C to +125°C; typical specifications are at T_A = 25°C; all specifications are at AVDD - AVSS = 5 V, IOVDD = 1.8 V, V_{IN} = 0 V, V_{CM} = (AVDD - AVSS)/2, (V_{REFP} – $V_{REFN)}$ = 2.5 V, all speed modes, input buffers on, external clock, and reference buffer on (unless otherwise noted)

PARAMETER		TEST CONDITIONS		MIN	TYP	MAX	UNIT
f_{DATA}	Output data rate, sinc^3 or sinc^4 filter	Speed mode 3 (f_{MOD} = 12.8 MHz)		0.08		1067	kSPS
		Speed mode 2 (f_{MOD} = 6.4 MHz)		0.04		533.3	
		Speed mode 1 (f_{MOD} = 1.6 MHz)		0.01		133.3	
		Speed mode 0 (f_{MOD} = 0.8 MHz)		0.005		66.7	
NMRR	Normal-mode rejection ratio, simultaneous 50/60Hz notch filter enabled	f_{IN} = 50 Hz or 60 Hz (±1 Hz), f_{DATA} = 20 SPS, external f_{CLK} = 25.6 MHz			–95.3		dB
		f_{IN} = 50 Hz or 60 Hz (±1 Hz), f_{DATA} = 20 SPS, internal f_{CLK} = 25.6 MHz			–82.7		
		f_{IN} = 50 Hz or 60 Hz (±1 Hz), f_{DATA} = 25 SPS, external f_{CLK} = 25.6 MHz			–62.7		
		f_{IN} = 50 Hz or 60 Hz (±1 Hz), f_{DATA} = 25 SPS, internal f_{CLK} = 25.6 MHz			–57.9		
CMRR	Common-mode rejection ratio	At dc			110		dB
		f_{CM} = 50Hz or 60Hz (±1Hz), f_{DATA} = 20SPS or 25SPS			125		
		f_{CM} = 50Hz or 60Hz (±1Hz), f_{DATA} > 25SPS			90		
PSRR	Power-supply rejection ratio	AVDD at dc			105		dB
		IOVDD at dc			110		
VOLTAGE REFERENCE INPUTS							
	REFP and REFN input current (REFN = AVSS)	REFP buffer off	Speed mode 3		190		μA/V
			Speed mode 2		130		
			Speed mode 1		80		
			Speed mode 0		70		
	REFP input current	REFP buffer on	Speed mode 3		±0.5		μA
			Speed mode 2		±0.3		
			Speed mode 1		±0.1		
			Speed mode 0		±0.1		
	REFP and REFN input current drift	REFP buffer off	Speed mode 3		2.5		nA/°C
			Speed mode 2		5		
			Speed mode 1		7		
			Speed mode 0		7.5		
	REFP input current drift	REFP buffer on	Speed mode 3		4		nA/°C
			Speed mode 2		2.5		
			Speed mode 1		0.5		
			Speed mode 0		0.5		
INTERNAL VOLTAGE REFERENCE							
	Output voltage	AVDD > 4.5V, REFOUT with respect to AVSS	REF_VAL = 0b		2.5		V
			REF_VAL = 1b		4.096		
		2.85V ≤ AVDD ≤ 4.5 V, REFOUT with respect to AVSS			2.5		
	Initial accuracy	T_A = 25°C		–0.1	±0.02	0.1	%
	Temperature drift[1]	T_A = 0°C to +125°C			4	12	ppm/°C
		T_A = –40°C to +125°C			4	12	
	Reference output load current	Sink or source		–10		10	mA
	Short-circuit current limit	Sink or source			25	40	
	Power supply rejection	AVDD at DC			95		dB
	Load regulation				50		ppm/mA
	Voltage noise	0.1 to 10Hz, C_L = 1μF	V_{REF} = 2.5 V		1.0		μV_{RMS}
			V_{REF} = 4.096 V		1.7		

minimum and maximum specifications apply from T_A = –40°C to +125°C; typical specifications are at T_A = 25°C; all specifications are at AVDD - AVSS = 5 V, IOVDD = 1.8 V, V_{IN} = 0 V, V_{CM} = (AVDD - AVSS)/2, (V_{REFP} – $V_{REFN)}$ = 2.5 V, all speed modes, input buffers on, external clock, and reference buffer on (unless otherwise noted)

PARAMETER		TEST CONDITIONS		MIN	TYP	MAX	UNIT
	Voltage noise density	1kHz, C_L = 1µF	V_{REF} = 2.5 V		200		nV/√Hz
			V_{REF} = 4.096 V		300		
	Capacitive load			0.5	1	2	µF
	Resistive load			2			kΩ
	Start-up time	From power-down mode, C_L = 1 µF, 0.01% settling			1	1.5	ms
INTERNAL OSCILLATOR							
f_{OSCM}	Frequency				25.6		MHz
	Accuracy			–0.75	0.07	0.75	%
TEMPERATURE SENSOR							
TS_{Offset}	Output voltage	T_A = 25°C			120		mV
TS_{TC}	Temperature coefficient				400		µV/°C
	Accuracy	Speed mode 0, internal clock, OSR = 12, input buffers on		–3	±0.5	3	°C
MONITORS							
TH_{REF_UV}	Reference undervoltage threshold [2]			0.5		0.6	V
	External reference voltage readback accuracy	(V_{REFP} – V_{REFN}) / 3			±0.5		%
	Supply voltage readback accuracy	AVDD / 3			±1		%
		DVDD / 3			±1		
		(CAPA – AVSS) / 3			±1		
		(CAPD – DGND) / 3			±1		
		(RESP – RESN) / 3			±1		
OPEN-WIRE DETECTION CURRENT SOURCES AND SINKS							
	Current source settings				2, 8, 85		µA
	Current sink settings				2, 8, 85		µA
	Current source accuracy	85µA current setting			±2		%
		All other current settings			±40		
	Current sink accuracy	85µA current setting			±2		
		All other current settings			±40		
TEST DAC							
	Test DAC reference voltage	AVDD ≥ 4.65 V	TDAC_RANGE = 0b		2.5		V
			TDAC_RANGE = 1b		4.096		
		2.85V ≤ AVDD ≤ 4.65 V			2.5		
	Resolution				5		Bits
	Accuracy				±1		%
	Offset error	Measured by the line passing through two codes			±1	10	mV
	Startup Time	From rising edge of TDAC enable to DAC output voltage reaching 99% of its final value. Applies to the slowest DAC config (buffer or no buffer).			10		µs
	Buffer output impedance	Buffer shorted to ground with code = 11111, reference voltage = 4.096 V	Buffer shorted to ground with code = 11111, reference voltage = 4.096 V		150		Ω
	Buffer Load Current, Shorted	Buffer shorted to ground with code = 11111, reference voltage = 4.096 V			15		mA
ANALOG GENERAL-PURPOSE INPUTS/OUTPUTS (GPIO)							
V_{IL}	Logic input level, low					0.35 AVDD	V
V_{IH}	Logic input level, high			0.65 AVDD			V
V_{OL}	Logic output level, low	I_{OL} = 0.4 mA				0.45 AVDD	V

minimum and maximum specifications apply from T_A = –40°C to +125°C; typical specifications are at T_A = 25°C; all specifications are at AVDD - AVSS = 5 V, IOVDD = 1.8 V, V_{IN} = 0 V, V_{CM} = (AVDD - AVSS)/2, (V_{REFP} – $V_{REFN)}$ = 2.5 V, all speed modes, input buffers on, external clock, and reference buffer on (unless otherwise noted)

PARAMETER		TEST CONDITIONS		MIN	TYP	MAX	UNIT
V_{OH}	Logic output level, high	I_{OH} = –0.4 mA		0.6 AVDD			V
	Input hysteresis				30		mV
	Input current			–2		2	μA
DIGITAL INPUTS/OUTPUTS							
V_{IL}	Logic input level, low			DGND		0.3 × IOVDD	V
V_{IH}	Logic input level, high			0.7 × IOVDD			V
V_{OL}	Logic output level, low	OUT_DRV = 0b, I_{OL} = 2 mA				0.2 × IOVDD	V
		OUT_DRV = 1b, I_{OL} = 1 mA				0.2 × IOVDD	
V_{OH}	Logic output level, high	OUT_DRV = 0b, I_{OH} = –2 mA		0.8 × IOVDD			V
		OUT_DRV = 1b, I_{OH} = –1 mA		0.8 × IOVDD			
	Input hysteresis				100		mV
	Input current	Excluding $\overline{\text{RESET}}$ pin		–1		1	μA
	$\overline{\text{RESET}}$ pin pullup resistor	Internal pullup resistor			20		kΩ
ANALOG SUPPLY CURRENT							
I_{AVDD}, I_{AVSS}	AVDD and AVSS current (All buffers off, internal reference off)	Speed mode 3			6.5	7.5	mA
		Speed mode 2			4	5	
		Speed mode 1			1.5	2	
		Speed mode 0			1.25	1.5	
		Standby mode			230		μA
		Power-down mode			550		nA
	AVDD and AVSS additional current (per function)	Input buffer	Speed mode 3		2.5	3	mA/buffer
			Speed mode 2		1.75	2	
			Speed mode 1		0.5	0.75	
			Speed mode 0		0.4	0.6	
		REFP buffer	Speed mode 3		1.2		mA/buffer
			Speed mode 2		0.8		
			Speed mode 1		0.3		
			Speed mode 0		0.25		
		Internal reference and reference buffer			100	200	μA
		Test DAC, 2.5 V or 4.096 V range			700		μA
		Test DAC buffer			800		μA
DIGITAL SUPPLY CURRENT							
I_{IOVDD}	IOVDD current	Speed mode 3, OSR = 32			0.9	1	mA
		Speed mode 2, OSR = 32			0.5	0.6	
		Speed mode 1, OSR = 32			0.2	0.25	
		Speed mode 0, OSR = 32			0.12	0.2	
		Standby mode, external clock			5		μA
		Standby mode, internal oscillator			60		
		Power-down mode			10		
POWER DISSIPATION							
P_D	Power dissipation	AVDD = 5 V, Input and REFP buffers off	Speed mode 3		33		mW
			Speed mode 2		20		
			Speed mode 1		6.6		
			Speed mode 0		6		

[1] Specified by design and characterization, not production tested.
[2] Undervoltage monitor does always trip below the specified MIN value and does never trip above the specified MAX value.

5.6 Timing Requirements

over operating ambient temperature range (unless otherwise noted)

		MIN	MAX	UNIT
$t_{c(SC)}$	SCLK period	40	1/(4 f_{DATA})	ns
$t_{w(SCL)}$	Pulse duration, SCLK low	10		ns
$t_{w(SCH)}$	Pulse duration, SCLK high	20		ns
$t_{d(CSSC)}$	Delay time, first SCLK rising edge after $\overline{CS}$ falling edge	5		ns
$t_{d(SCCS)}$	Delay time, $\overline{CS}$ rising edge after final SCLK falling edge	5		ns
$t_{w(CSH)}$	Pulse duration, $\overline{CS}$ high	5		ns
$t_{d(FIFORD)}$	Delay time between subsequent FIFO read frames	5		t_{CLK}
$t_{su(DI)}$	Setup time, SDI valid before SCLK falling edge	3		ns
$t_{h(DI)}$	Hold time, SDI valid after SCLK falling edge	4		ns
$t_{d(fr2fr)}$	Delay time, between frames in 3-wire SPI mode	5		ns
$t_{h(DIIR)}$	Hold time, SDI high to force interface resynchronization (3-wire SPI mode only). Interface resynchronization happens on first SCLK falling edge where SDI is low again.	63		t_{SCLK}
$t_{d(RSSC)}$	Delay time, SPI communication start after $\overline{RESET}$ rising edge or after software reset using SPI reset pattern or RESET[7:0] bit field		500	µs
$t_{d(POR)}$	Delay time, first SCLK rising edge after IOVDD power-up (= after IOVDD crosses minimum IOVDD voltage)	5		ms
$t_{w(STL)}$	Pulse duration, START low	4		t_{CLK}
$t_{w(STH)}$	Pulse duration, START high	4		t_{CLK}
$t_{su(STCL)}$	Setup time, START transition before CLKIN rising edge [(1)]	9		ns
$t_{h(STCL)}$	Hold time, START transition after CLKIN rising edge [(1)]	9		ns
$t_{su(STFS)}$	Setup time, START falling edge or STOP bit before FSYNC rising edge to stop next conversion (start/stop conversion mode)	24		t_{CLK}
$t_{w(RSL)}$	Pulse duration, $\overline{RESET}$ low	4		t_{CLK}
$t_{h(DIRS1)}$	Hold time, SDI high to force device reset using RESET pattern. Device reset happens on first SCLK falling edge where SDI is low again.	1023		t_{SCLK}
$t_{h(DIRS2)}$	Hold time, SDI high to force device reset using RESET pattern. Device reset happens on $\overline{CS}$ rising edge (4-wire SPI mode only).	1024		t_{SCLK}

[(1)] START rising edge must not be applied between the setup and hold time period at the rising edge of CLKIN

5.7 Switching Characteristics

over operating ambient temperature range, C_{LOAD} = 20pF (unless otherwise noted)

PARAMETER		TEST CONDITIONS	MIN	TYP	MAX	UNIT
$t_{p(CSDO)}$	Propagation delay time, $\overline{CS}$ falling edge to SDO/$\overline{DRDY}$ driven				20	ns
$t_{p(CSDOZ)}$	Propagation delay time, $\overline{CS}$ rising edge to SDO/$\overline{DRDY}$ high impedance state				10	ns
$t_{h(SCDO)}$	Hold time, SCLK rising edge to invalid SDO/$\overline{DRDY}$		1			ns
$t_{p(SCDO)}$	Propagation delay time, SCLK rising edge to valid SDO/$\overline{DRDY}$				18	ns
$t_{w(DRH)}$	Pulse duration, $\overline{DRDY}$ high		2			t_{MOD}
$t_{p(SCDR)}$	Propagation delay time, 8th SCLK falling edge to $\overline{DRDY}$ return high				5	t_{MOD}
$t_{p(DRDO)}$	Propagation delay time, first SCLK rising edge of read operation for SDO/$\overline{DRDY}$ transition from $\overline{DRDY}$ mode to valid SDO	SDO_DRDY = 1b	30		46	ns
$t_{p(DODR)}$	Propagation delay time, last SCLK falling edge of read operation for SDO/$\overline{DRDY}$ transition from SDO to $\overline{DRDY}$ mode	SDO_DRDY = 1b	30		45	ns
$t_{p(GPIO)}$	Propagation delay time, last SCLK falling edge (3-wire) or $\overline{CS}$ rising edge (4-wire), of GPIO_CFG/STEP_GPIO_DATA_OUT to GPIOx output valid				14	ns

5.8 Timing Diagrams

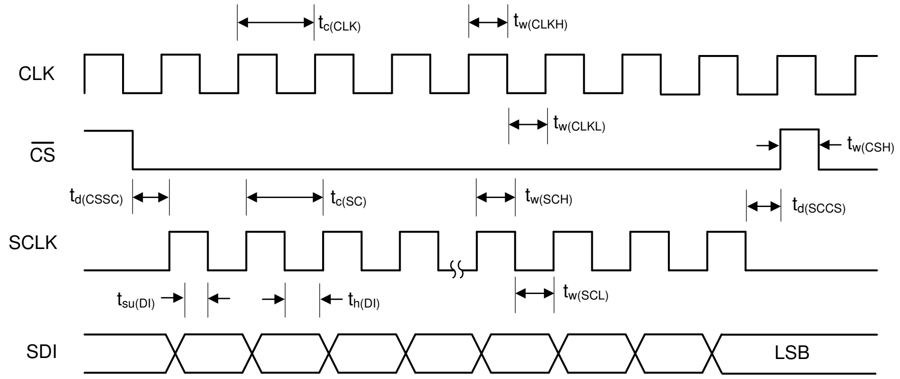


Figure 5-1. Clock and Serial Interface Timing Requirements

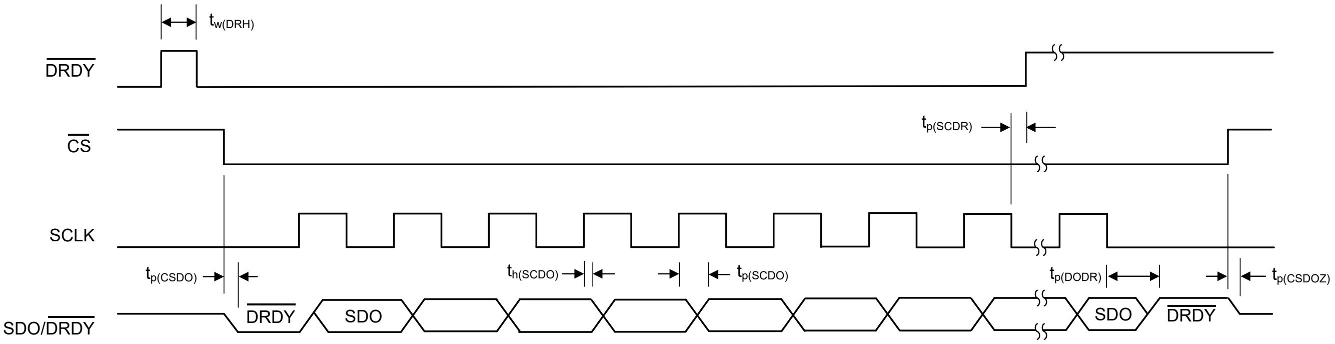


Figure 5-2. Serial Interface Switching Characteristics

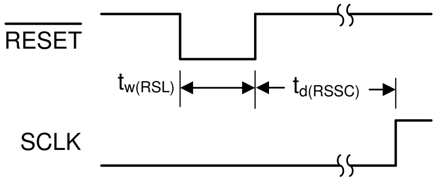


Figure 5-3. $\overline{\text{RESET}}$ Pin Timing

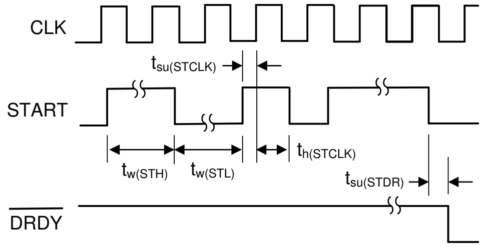


Figure 5-4. START Pin Timing

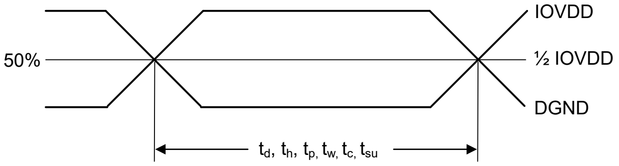
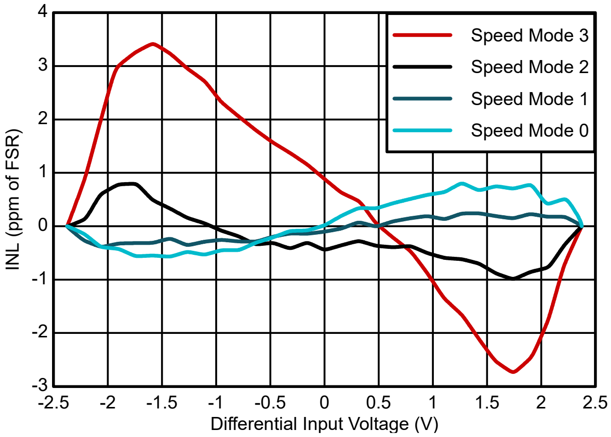


Figure 5-5. Timing Reference

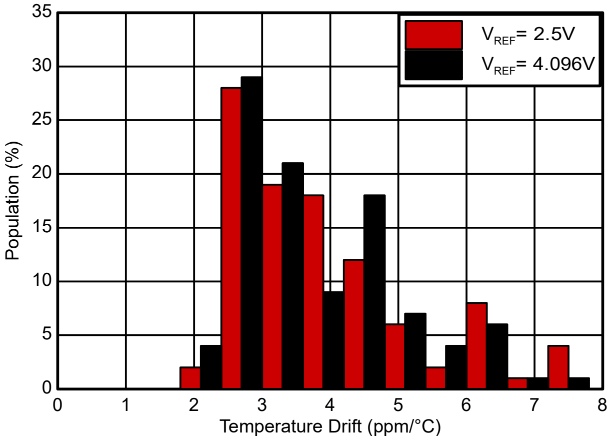
5.9 Typical Characteristics

AVDD = 5V, AVSS = 0V, IOVDD = 1.8V, V_{REF} = 2.5V, OSR = 32, input buffers on, and T_A = 25°C (unless otherwise noted).



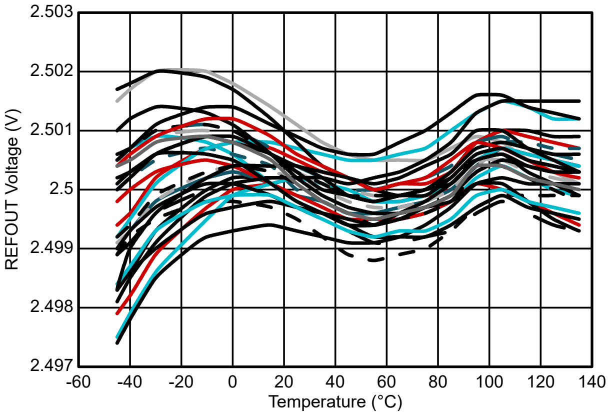
OSR = 128, sinc4, AIN10-AIN11, external 2.5V reference

Figure 5-6. INL vs Differential Input Voltage



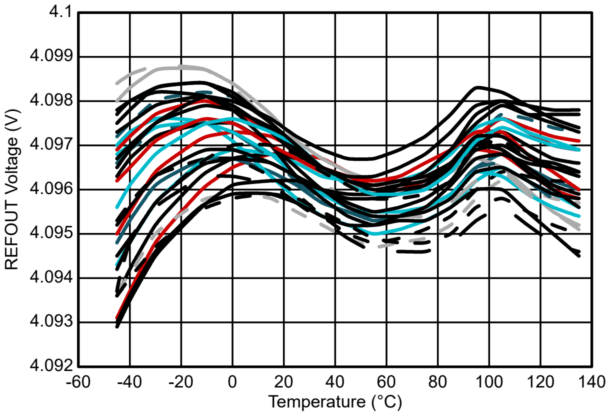
65 units, ADC in standby mode

Figure 5-7. Internal Reference Temperature Drift



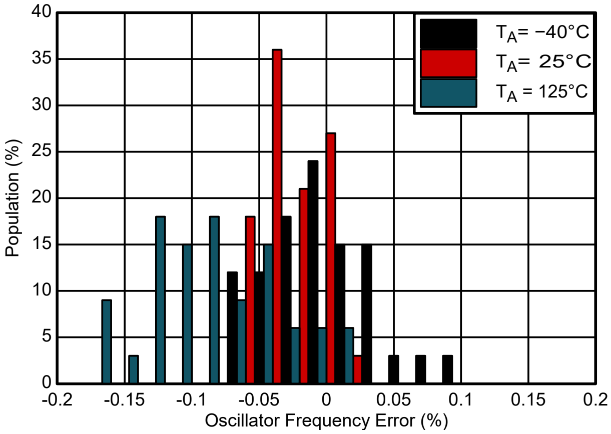
REFOUT = 2.5V, 65 units, ADC in standby mode

Figure 5-8. Internal Reference Voltage vs Temperature, 2.5V



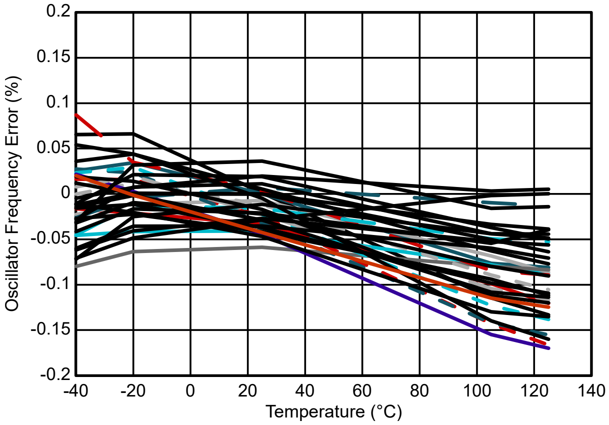
REFOUT = 4.096V, 65 units, ADC in standby mode

Figure 5-9. Internal Reference Voltage vs Temperature, 4.096V



35 units

Figure 5-10. Internal Oscillator Frequency Histogram



35 units

Figure 5-11. Internal Oscillator Frequency vs Temperature

6 Parameter Measurement Information

6.1 Offset Error Measurement

Offset error of the ADS125P08 is measured with the ADC inputs externally shorted together. The input common-mode voltage is fixed to the mid-point of the AVDD1 and AVSS power-supply range. Offset error is specified at T_A = 25°C.

6.2 Offset Drift Measurement

Offset drift is defined as the change in offset voltage measured at multiple points over the specified temperature range. Offset drift is calculated using the *box method* in which a box is formed over the maximum and minimum offset voltages and over the specified temperature range. The box method specifies limits for the temperature error but does not specify the exact shape and slope of the device under test.

Use this equation for the offset drift calculation using the box method:

$$\text{Offset Drift (nV/°C)} = 10^9 \cdot (V_{OFSMAX} - V_{OFSMIN}) / (T_{MAX} - T_{MIN}) \tag{1}$$

where:

- V_{OFSMAX} and V_{OFSMIN} = Maximum and minimum offset voltages over the specified temperature range
- T_{MAX} and T_{MIN} = Maximum and minimum temperatures

6.3 Gain Error Measurement

Gain error is defined as the difference between the actual and the ideal slopes of the ADC transfer function. Gain error is measured by applying dc test voltages at –95% and 95% of FSR. The error is calculated by subtracting the difference of the dc test voltages (ideal slope) from the difference in the ADC output voltages (actual slope). The difference in the slopes is divided by the ideal slope and multiplied by 10^6 to convert the error to ppm of FSR. Error resulting from the ADC reference voltage is excluded from the gain error measurement. The gain error is specified at T_A = 25°C. Equation 2 shows the calculation of gain error:

$$\text{Gain Error (ppm of FSR)} = 10^6 \times (\Delta V_{OUT} - \Delta V_{IN}) / \Delta V_{IN} \tag{2}$$

where:

- ΔV_{OUT} = Difference of two ADC output voltages
- ΔV_{IN} = Difference of two input test voltages

6.4 Gain Drift Measurement

Gain drift is defined as the change of gain error measured at multiple points over the specified temperature range. The box method is used in which a box is formed over the maximum and minimum gain errors over the specified temperature range. The box method specifies limits for the temperature error but does not specify the exact shape and slope of the device under test. Equation 3 describes gain drift using the box method.

$$\text{Gain Drift (ppm/°C)} = (GE_{MAX} - GE_{MIN}) / (T_{MAX} - T_{MIN}) \tag{3}$$

where:

- GE_{MAX} and GE_{MIN} = Maximum and minimum gain errors over the specified temperature range
- T_{MAX} and T_{MIN} = Maximum and minimum temperatures

6.5 NMRR Measurement

Normal-mode rejection ratio (NMRR) specifies the ability of the ADC to reject normal-mode input signals at specific frequencies, typically expressed at 50Hz and 60Hz input frequencies. Normal-mode rejection is uniquely determined by the frequency response of the digital filter. In this case, the nulls in the frequency response of the low-latency sinc3 filter option located at 50Hz and 60Hz provide rejection at these frequencies.

6.6 CMRR Measurement

Common-mode rejection ratio (CMRR) specifies the ability of the ADC to reject common-mode input signals. CMRR is expressed as dc and ac parameters. For measurement of CMRR (dc), three common-mode test voltages equal to AVSS + 50mV, (AVDD1 + AVSS) / 2, and AVDD1 – 50mV are applied with the inputs externally shorted together. The maximum change of the ADC offset voltage is recorded versus the change in common-mode test voltage. Equation 4 shows how CMRR (dc) is computed.

$$\text{CMRR (dc) (dB)} = 20 \times \log(\Delta V_{CM} / \Delta V_{OS}) \quad (4)$$

where:

- ΔV_{CM} = Change of dc common-mode test voltage
- ΔV_{OS} = Change of corresponding offset voltage

For the measurement of CMRR (ac), an ac common-mode signal is applied at various test frequencies at 95% full-scale range. An FFT is computed from the ADC data with the common-mode signal applied. As shown in Equation 5, the nine largest amplitude spurious frequencies in the frequency spectrum are summed as powers and related to the amplitude of the common-mode test signal.

$$\text{PSRR (ac) (dB)} = 20 \times \log(V_{CM} / V_{O}) \quad (5)$$

where:

- V_{CM} (RMS) = Common-mode input signal amplitude
- V_O (RMS) = Root-sum-square amplitude of spurious frequencies = $\sqrt{}(V_0{}^2 + V_1{}^2 + ...V_8{}^2)$

6.7 PSRR Measurement

Power-supply rejection ratio (PSRR) specifies the ability of the ADC to reject power-supply interference. PSRR is expressed as ac and dc parameters. For measurement of PSRR (dc), the power-supply voltage is changed over the range of minimum, nominal, and maximum specified voltages with the inputs externally shorted together. The maximum change of ADC offset voltage is recorded versus the change in power-supply voltage. PSRR (dc) is computed as shown in Equation 6 as the ratio of change of the power-supply voltage step to the change of offset voltage.

$$\text{PSRR (dc) (dB)} = 20 \times \log(\Delta V_{PS} / \Delta V_{OS}) \quad (6)$$

where:

- ΔV_{PS} = Change of power-supply voltage
- ΔV_{OS} = Change of offset voltage

For the measurement of PSRR (ac), the power-supply voltage is modulated by a 100mVpp (35mV$_{RMS}$) signal at various test frequencies. An FFT of the ADC data with power-supply modulation is performed. As shown in Equation 7, the nine largest amplitude spurious frequencies in the frequency spectrum are summed as powers and related to the amplitude of the power-supply modulation signal.

$$\text{PSRR (ac) (dB)} = 20 \times \log(V_{PS} / V_{O}) \quad (7)$$

where:

- V_{PS} (RMS) = 100mV ac power-supply modulation signal
- V_O (RMS) = Root-sum-square amplitude of spurious frequencies = $\sqrt{}(V_0{}^2 + V_1{}^2 + ...V_8{}^2)$

6.8 SNR Measurement

Signal-to-noise ratio (SNR) is a measure of noise performance with a full-scale ac input signal. For the SNR measurement, a –0.2dBFS, 1kHz test signal is used with V_{CM} equal to the mid-supply voltage. As shown in Equation 8, SNR is the ratio of the rms value of the input signal to the root-sum-square of all other frequency components derived from the FFT result of the ADC output samples. DC and harmonics of the original signal are excluded from the SNR calculation. In a test case where an FFT window function is used because of non-coherent sampling, the spectral leakage of adjacent frequency bins surrounding dc, the original signal and signal harmonics are removed to calculate SNR.

$$\text{SNR (dB)} = 20 \times \log(V_{IN} / e_n) \tag{8}$$

where:

- V_{IN} = Input test signal
- e_n = Root-sum-square of frequency components excluding dc and signal harmonics

6.9 INL Error Measurement

Integral nonlinearity (INL) error specifies the linearity of the ADC dc transfer function. INL is measured by applying a series of dc test voltages along a straight line computed from the slope and offset transfer function of the ADC. INL is the difference between a set of dc test voltages [$V_{IN(N)}$] to the corresponding set of output voltages [$V_{OUT(N)}$]. Equation 9 shows the *end-point method* of calculating INL error.

$$\text{INL (ppm of FSR)} = \text{maximum absolute value of INL test series } [10^6 \times (V_{IN(N)} - V_{OUT(N)}) / \text{FSR}] \tag{9}$$

where:

- N = Index of dc test voltage
- [$V_{IN(N)}$] = Set of test voltages over the range –95% to 95% of FSR
- [$V_{OUT(N)}$] = Set of corresponding ADC output voltages
- FSR (full-scale range) = 2 × V_{REF} (1x input range) or 4 × V_{REF} (2x input range)

The INL *best-fit method* uses a least-squared error (LSE) calculation to determine a new straight line to minimize the root-sum-square of the INL errors above and below the original end-point line.

6.10 THD Measurement

Total harmonic distortion (THD) specifies the dynamic linearity of the ADC with an ac input signal. For the THD measurement, a –0.2dBFS, 1kHz differential input signal with V_{CM} equal to the mid-supply voltage is applied. A sufficient number of data points are collected to yield an FFT result with frequency bin widths of 5Hz or less. The 5Hz bin width reduces the noise in the harmonic bins for consistent THD measurements. As shown in Equation 10, THD is calculated as the ratio of the root-sum-square amplitude of harmonics to the input signal amplitude.

$$\text{THD (dB)} = 20 \times \log(V_H / V_{IN}) \tag{10}$$

where:

- V_H = Root-sum-square of harmonics: $\sqrt{(V_2{}^2 + V_3{}^2 + \ldots V_n{}^2)}$, where V_n = Ninth harmonic voltage
- V_{IN} = Input signal fundamental

6.11 SFDR Measurement

Spurious-free dynamic range (SFDR) is the ratio of the rms value of a single-tone ac input to the highest spurious signal in the ADC frequency spectrum. SFDR measurement includes harmonics of the original signal. For the SFDR measurement, a –0.2dBFS, 1kHz input signal with V_{CM} equal to the mid-supply voltage is applied. As shown in Equation 11, SFDR is the ratio of the rms values of the input signal to the single highest spurious signal, including harmonics of the original signal.

$$\text{SFDR (dB)} = 20 \times \log(V_{IN} / V_{SPUR}) \tag{11}$$

where:

- V_{IN} = Input test signal
- V_{SPUR} = Single highest spurious level

6.12 Noise Performance

Delta-sigma (ΔΣ) analog-to-digital converters (ADCs) are based on the principle of oversampling. The input signal of a ΔΣ ADC is sampled at a high frequency (modulator frequency) and subsequently filtered and decimated in the digital domain to yield a conversion result at the respective output data rate. The ratio between modulator frequency and output data rate is called the oversampling ratio (OSR). By increasing the OSR, and thus reducing the output data rate, the noise performance of the ADC can be optimized. In other words, the input-referred noise drops when reducing the output data rate because more samples of the internal modulator are averaged to yield one conversion result.

The ADC provides four speed modes that allow trade-offs between ADC resolution, power consumption, and signal bandwidth. The modes are speed mode 3, speed mode 2, speed mode 1 and speed mode 0, with decreasing orders of device power consumption.

The digital filter provides the options of sinc3, sinc4, and sinc4 + sinc1 configurations. The sinc3 and sinc4 filters offer data rates up to 1.066MSPS in speed mode 3, up to 533kSPS in speed mode 2, up to 133kSPS in speed mode 1 and up to 66.67kSPS in speed mode 0. The device also offers custom 20SPS and 25SPS filter modes with simultaneous 50Hz/60Hz rejection.

The following tables in this section summarize the noise performance and resulting effective resolution of the various filter modes. Noise performance is shown for a 2.5V and a 4.096V reference voltage. In general, decreasing the reference voltage from 4.096V to 2.5V decreases dynamic range by 4dB (typical).

The data shown are typical input-referred noise results ($e_{n(RMS)}$) in units of μV_{RMS} with the analog inputs shorted together and are representative of typical performance at T_A = 25°C. A minimum of 1,000 or 10 seconds of consecutive conversions (whichever occurs first) are used to measure RMS noise ($e_{n(RMS)}$). Because of the statistical nature of noise, repeated noise measurements can yield higher or lower noise results.

Equation 12 or Equation 13 calculate effective resolution from the measured μV_{RMS} numbers, depending on the selected coding scheme.

$$\text{Effective Resolution}_{\text{Binary two's complement coding}} = \ln[(\text{FSR} / e_{n(RMS)}] / \ln(2) \tag{12}$$

$$\text{Effective Resolution}_{\text{Unipolar straight binary coding}} = \ln[(0.5 \times \text{FSR}) / e_{n(RMS)}] / \ln(2) \tag{13}$$

where:

- FSR = 2 × V_{REF}
- $e_{n(RMS)}$ = Noise voltage (RMS)

Note that all values in this section are generated using the ±10V input range, thus FSR = 20V.

Input-referred noise (e_n) in units of μV_{PP} can be estimated as $e_{n(PP)} = 6.6 \times e_{n(RMS)}$. Use Equation 14 or Equation 15 to calculate noise-free resolution from the estimated μV_{PP} numbers, depending on the selected coding scheme.

$$\text{Noise-free Resolution}_{\text{Binary two's complement coding}} = \ln[\text{FSR} / e_{n(PP)}] / \ln(2) \tag{14}$$

$$\text{Noise-free Resolution}_{\text{Unipolar straight binary coding}} = \ln[(0.5 \times \text{FSR}) / e_{n(PP)}] / \ln(2) \tag{15}$$

When evaluating ADC noise performance, consider the effect of external components to the total noise performance. The noise performance of the ADC can be evaluated in isolation by selecting the input short test connection of the input multiplexer.

Table 6-1. Sinc3 and Sinc4 Filter Noise Performance (V_{REF} = 2.5V)

OSR	DATA RATE (kSPS)	NOISE ($e_{n(RMS)}$, μV_{RMS})[1]		EFFECTIVE RESOLUTION ±2.5V RANGE (Bits)	
		SINC3	SINC4	SINC3	SINC4
SPEED MODE 3 (f_{MOD} =12.8MHz)					
12	1066.67	146.67	41.20	15.1	16.9
16	800.00	62.88	17.52	16.3	18.1
24	533.33	20.41	9.52	17.9	19.0
32	400.00	11.22	7.71	18.8	19.3
64	200.00	5.85	5.34	19.7	19.8
128	100.00	4.06	3.76	20.2	20.3
256	50.00	2.92	2.68	20.7	20.8
512	25.00	1.98	1.88	21.3	21.3
1,024	12.50	1.41	1.35	21.8	21.8
2,048	6.25	1.03	0.97	22.2	22.3
4,000	3.20	0.75	0.69	22.7	22.8
8,000	1.60	0.55	0.51	23.1	23.2
16,000	0.80	0.41	0.39	23.5	23.6
26,667	0.48	0.39	0.35	23.6	23.8
32,000	0.40	0.35	0.37	23.8	23.7
96,000	0.13	0.34	0.26	23.8	24.2
160,000	0.08	0.25	0.27	24.3	24.1
SPEED MODE 2 (f_{MOD} =12.8MHz)					
12	533.33	148.06	41.26	15.0	16.9
16	400.00	62.62	17.28	16.3	18.1
24	266.67	20.22	9.31	17.9	19.0
32	200.00	10.94	7.46	18.8	19.4
64	100.00	5.69	5.23	19.7	19.9
128	50.00	3.92	3.70	20.3	20.4
256	25.00	2.76	2.59	20.8	20.9
512	12.50	2.01	1.86	21.2	21.4
1,024	6.25	1.38	1.34	21.8	21.8
2,048	3.13	1.00	0.93	22.3	22.4
4,000	1.60	0.72	0.67	22.7	22.8
8,000	0.80	0.50	0.49	23.3	23.3
16,000	0.40	0.37	0.35	23.7	23.8
26,667	0.24	0.30	0.29	24.0	24.0
32,000	0.20	0.28	0.26	24.1	24.2
96,000	0.07	0.18	0.17	24.7	24.8
160,000	0.04	0.15	0.15	25.0	25.0
SPEED MODE 1 (f_{MOD} =1.6MHz)					
12	533.33	148.02	40.24	15.0	16.9
16	400.00	63.02	17.53	16.3	18.1
24	266.67	20.45	9.18	17.9	19.1
32	200.00	11.23	7.57	18.8	19.3
64	100.00	5.61	5.17	19.8	19.9
128	50.00	3.89	3.64	20.3	20.4

Table 6-1. Sinc3 and Sinc4 Filter Noise Performance (V_{REF} = 2.5V) (continued)

OSR	DATA RATE (kSPS)	NOISE ($e_{n(RMS)}$, μV_{RMS})[1]		EFFECTIVE RESOLUTION ±2.5V RANGE (Bits)	
		SINC3	SINC4	SINC3	SINC4
256	25.00	2.75	2.56	20.8	20.9
512	12.50	1.95	1.83	21.3	21.4
1,024	6.25	1.38	1.30	21.8	21.9
2,048	3.13	0.96	0.93	22.3	22.4
4,000	1.60	0.71	0.66	22.7	22.9
8,000	0.80	0.51	0.49	23.2	23.3
16,000	0.40	0.38	0.36	23.6	23.7
26,667	0.24	0.31	0.28	23.9	24.1
32,000	0.20	0.28	0.26	24.1	24.2
96,000	0.07	0.19	0.19	24.6	24.6
160,000	0.04	0.18	0.15	24.7	25.0
SPEED MODE 0 (f_{MOD} =0.8MHz)					
12	533.33	147.24	40.20	15.1	16.9
16	400.00	62.46	16.98	16.3	18.2
24	266.67	20.56	9.21	17.9	19.1
32	200.00	10.94	7.42	18.8	19.4
64	100.00	5.60	5.19	19.8	19.9
128	50.00	3.92	3.72	20.3	20.4
256	25.00	2.78	2.55	20.8	20.9
512	12.50	1.98	1.79	21.3	21.4
1,024	6.25	1.36	1.32	21.8	21.9
2,048	3.13	0.96	0.91	22.3	22.4
4,000	1.60	0.71	0.66	22.7	22.9
8,000	0.80	0.50	0.47	23.3	23.3
16,000	0.40	0.35	0.33	23.8	23.9
26,667	0.24	0.29	0.29	24.0	24.0
32,000	0.20	0.25	0.25	24.3	24.3
96,000	0.07	0.19	0.17	24.6	24.8
160,000	0.04	0.15	0.15	25.0	25.0

[1] High OSR values can yield varying noise results because of the limits of 24-bit quantization: 2.5V / 2^{23} = 0.298µV / code.

Table 6-2. Sinc3 and Sinc4 Filter Noise Performance (V_{REF} = 4.096V)

OSR	DATA RATE (kSPS)	NOISE ($e_{n(RMS)}$, μV_{RMS}) [1]		EFFECTIVE RESOLUTION ±4.096V RANGE (Bits)	
		SINC3	SINC4	SINC3	SINC4
SPEED MODE 3 (f_{MOD} =12.8MHz)					
12	1066.67	240.01	64.46	15.1	17.0
16	800.00	101.48	24.34	16.3	18.4
24	533.33	30.98	10.36	18.0	19.6
32	400.00	15.13	8.19	19.0	19.9
64	200.00	6.14	5.54	20.3	20.5
128	100.00	4.16	3.93	20.9	21.0
256	50.00	2.92	2.77	21.4	21.5
512	25.00	2.09	1.97	21.9	22.0

Table 6-2. Sinc3 and Sinc4 Filter Noise Performance (V_{REF} = 4.096V) (continued)

OSR	DATA RATE (kSPS)	NOISE ($e_{n(RMS)}$, μV_{RMS}) [1]		EFFECTIVE RESOLUTION ±4.096V RANGE (Bits)	
		SINC3	SINC4	SINC3	SINC4
1,024	12.50	1.47	1.36	22.4	22.5
2,048	6.25	1.04	0.99	22.9	23.0
4,000	3.20	0.79	0.74	23.3	23.4
8,000	1.60	0.58	0.56	23.8	23.8
16,000	0.80	0.45	0.44	24.1	24.2
26,667	0.48	0.39	0.41	24.3	24.3
32,000	0.40	0.39	0.37	24.3	24.4
96,000	0.13	0.35	0.36	24.5	24.4
160,000	0.08	0.32	0.31	24.6	24.7
SPEED MODE 2 (f_{MOD} =12.8MHz)					
12	1066.67	236.10	65.35	15.1	16.9
16	800.00	101.12	24.12	16.3	18.4
24	533.33	30.74	10.15	18.0	19.6
32	400.00	14.93	7.78	19.1	20.0
64	200.00	6.04	5.40	20.4	20.5
128	100.00	4.08	3.69	20.9	21.1
256	50.00	2.84	2.68	21.5	21.5
512	25.00	2.04	1.89	21.9	22.0
1,024	12.50	1.44	1.37	22.4	22.5
2,048	6.25	1.02	0.96	22.9	23.0
4,000	3.20	0.74	0.70	23.4	23.5
8,000	1.60	0.54	0.51	23.9	23.9
16,000	0.80	0.40	0.37	24.3	24.4
26,667	0.48	0.33	0.32	24.6	24.6
32,000	0.40	0.31	0.29	24.7	24.8
96,000	0.13	0.19	0.19	25.4	25.4
160,000	0.08	0.19	0.19	25.4	25.4
SPEED MODE 1 (f_{MOD} =1.6MHz)					
12	1066.67	241.10	63.37	15.1	17.0
16	800.00	99.71	24.14	16.3	18.4
24	533.33	30.61	10.22	18.0	19.6
32	400.00	15.05	7.88	19.1	20.0
64	200.00	5.91	5.36	20.4	20.5
128	100.00	4.01	3.80	21.0	21.0
256	50.00	2.81	2.65	21.5	21.6
512	25.00	2.02	1.86	22.0	22.1
1,024	12.50	1.38	1.29	22.5	22.6
2,048	6.25	1.02	0.95	22.9	23.0
4,000	3.20	0.73	0.66	23.4	23.6
8,000	1.60	0.53	0.51	23.9	23.9
16,000	0.80	0.41	0.39	24.3	24.3
26,667	0.48	0.32	0.33	24.6	24.6
32,000	0.40	0.30	0.30	24.7	24.7

Table 6-2. Sinc3 and Sinc4 Filter Noise Performance (V_{REF} = 4.096V) (continued)

OSR	DATA RATE (kSPS)	NOISE ($e_{n(RMS)}$, μV_{RMS}) [1]		EFFECTIVE RESOLUTION ±4.096V RANGE (Bits)	
		SINC3	SINC4	SINC3	SINC4
96,000	0.13	0.24	0.23	25.0	25.1
160,000	0.08	0.22	0.24	25.2	25.0
SPEED MODE 0 (f_{MOD} =0.8MHz)					
12	1066.67	235.34	63.38	15.1	17.0
16	800.00	100.08	23.57	16.3	18.4
24	533.33	31.39	10.32	18.0	19.6
32	400.00	15.03	7.72	19.1	20.0
64	200.00	5.89	5.39	20.4	20.5
128	100.00	3.94	3.75	21.0	21.1
256	50.00	2.78	2.61	21.5	21.6
512	25.00	1.96	1.86	22.0	22.1
1,024	12.50	1.43	1.33	22.4	22.6
2,048	6.25	1.02	0.95	22.9	23.0
4,000	3.20	0.73	0.69	23.4	23.5
8,000	1.60	0.52	0.50	23.9	24.0
16,000	0.80	0.39	0.37	24.3	24.4
26,667	0.48	0.31	0.31	24.7	24.7
32,000	0.40	0.32	0.28	24.6	24.8
96,000	0.13	0.22	0.22	25.2	25.2
160,000	0.08	0.19	0.15	25.4	25.7

[1] High OSR values can yield varying noise results because of the limits of 24-bit quantization: 4.096V / 2^{23} = 0.488µV / code.

Table 6-3. Sinc4 + Sinc1 Filter Noise Performance (V_{REF} = 2.5V)

OSR	DATA RATE (kSPS)	NOISE ($e_{n(RMS)}$, μV_{RMS}) [1]	EFFECTIVE RESOLUTION ±2.5V RANGE (Bits)
SPEED MODE 3 (f_{MOD} =12.8MHz)			
64 (32 × 2)	200.00	6.63	19.5
128 (32 × 4)	100.00	5.00	19.9
256 (32 × 8)	50.00	3.71	20.4
512 (32 × 16)	25.00	2.65	20.8
1024 (32 × 32)	12.50	1.92	21.3
2048 (32 × 64)	6.25	1.35	21.8
4000 (32 × 125)	3.20	1.00	22.3
8000 (32 × 250)	1.60	0.73	22.7
16000 (32 × 500)	0.80	0.53	23.2
26656 (32 × 833)	0.48	0.45	23.4
32000 (32 × 1000)	0.40	0.43	23.5
96000 (32 × 3000)	0.13	0.33	23.9
160000 (32 × 5000)	0.08	0.29	24.0
SPEED MODE 2 (f_{MOD} =6.4MHz)			
64 (32 × 2)	200.00	6.46	19.6
128 (32 × 4)	100.00	4.94	19.9

Table 6-3. Sinc4 + Sinc1 Filter Noise Performance (V_{REF} = 2.5V) (continued)

OSR	DATA RATE (kSPS)	NOISE ($e_{n(RMS)}$, μV_{RMS}) [(1)]	EFFECTIVE RESOLUTION ±2.5V RANGE (Bits)
256 (32 × 8)	50.00	3.61	20.4
512 (32 × 16)	25.00	2.62	20.9
1024 (32 × 32)	12.50	1.85	21.4
2048 (32 × 64)	6.25	1.31	21.9
4000 (32 × 125)	3.20	0.96	22.3
8000 (32 × 250)	1.60	0.68	22.8
16000 (32 × 500)	0.80	0.50	23.3
26656 (32 × 833)	0.48	0.39	23.6
32000 (32 × 1000)	0.40	0.36	23.7
96000 (32 × 3000)	0.13	0.22	24.4
160000 (32 × 5000)	0.08	0.20	24.6
SPEED MODE 1 (f_{MOD} =1.6MHz)			
64 (32 × 2)	200.00	6.37	19.6
128 (32 × 4)	100.00	4.94	19.9
256 (32 × 8)	50.00	3.62	20.4
512 (32 × 16)	25.00	2.63	20.9
1024 (32 × 32)	12.50	1.83	21.4
2048 (32 × 64)	6.25	1.31	21.9
4000 (32 × 125)	3.20	0.95	22.3
8000 (32 × 250)	1.60	0.66	22.9
16000 (32 × 500)	0.80	0.49	23.3
26656 (32 × 833)	0.48	0.39	23.6
32000 (32 × 1000)	0.40	0.36	23.7
96000 (32 × 3000)	0.13	0.23	24.4
160000 (32 × 5000)	0.08	0.20	24.6
SPEED MODE 0 (f_{MOD} =0.8MHz)			
64 (32 × 2)	200.00	6.38	19.6
128 (32 × 4)	100.00	4.82	20.0
256 (32 × 8)	50.00	3.57	20.4
512 (32 × 16)	25.00	2.54	20.9
1024 (32 × 32)	12.50	1.85	21.4
2048 (32 × 64)	6.25	1.33	21.8
4000 (32 × 125)	3.20	0.94	22.3
8000 (32 × 250)	1.60	0.67	22.8
16000 (32 × 500)	0.80	0.47	23.3
26656 (32 × 833)	0.48	0.36	23.7
32000 (32 × 1000)	0.40	0.36	23.7
96000 (32 × 3000)	0.13	0.22	24.4

Table 6-3. Sinc4 + Sinc1 Filter Noise Performance (V_{REF} = 2.5V) (continued)

OSR	DATA RATE (kSPS)	NOISE ($e_{n(RMS)}$, μV_{RMS}) [1]	EFFECTIVE RESOLUTION ±2.5V RANGE (Bits)
160000 (32 × 5000)	0.08	0.17	24.8

Table 6-4. Sinc4 + Sinc1 Filter Noise Performance (V_{REF} = 4.096V)

OSR	DATA RATE (kSPS)	NOISE ($e_{n(RMS)}$, μV_{RMS}) [1]	EFFECTIVE RESOLUTION ±4.096V RANGE (Bits)
SPEED MODE 3 (f_{MOD} =12.8MHz)			
64 (32 × 2)	200.00	6.82	20.2
128 (32 × 4)	100.00	5.24	20.6
256 (32 × 8)	50.00	3.83	21.0
512 (32 × 16)	25.00	2.73	21.5
1024 (32 × 32)	12.50	1.96	22.0
2048 (32 × 64)	6.25	1.43	22.4
4000 (32 × 125)	3.20	1.04	22.9
8000 (32 × 250)	1.60	0.74	23.4
16000 (32 × 500)	0.80	0.57	23.8
26656 (32 × 833)	0.48	0.46	24.1
32000 (32 × 1000)	0.40	0.45	24.1
96000 (32 × 3000)	0.13	0.38	24.4
160000 (32 × 5000)	0.08	0.31	24.7
SPEED MODE 2 (f_{MOD} =6.4MHz)			
64 (32 × 2)	200.00	6.76	20.2
128 (32 × 4)	100.00	5.14	20.6
256 (32 × 8)	50.00	3.73	21.1
512 (32 × 16)	25.00	2.72	21.5
1024 (32 × 32)	12.50	1.94	22.0
2048 (32 × 64)	6.25	1.38	22.5
4000 (32 × 125)	3.20	0.99	23.0
8000 (32 × 250)	1.60	0.72	23.4
16000 (32 × 500)	0.80	0.51	23.9
26656 (32 × 833)	0.48	0.41	24.3
32000 (32 × 1000)	0.40	0.39	24.3
96000 (32 × 3000)	0.13	0.26	24.9
160000 (32 × 5000)	0.08	0.21	25.2
SPEED MODE 1 (f_{MOD} =1.6MHz)			
64 (32 × 2)	200.00	6.62	20.2
128 (32 × 4)	100.00	5.13	20.6
256 (32 × 8)	50.00	3.67	21.1
512 (32 × 16)	25.00	2.63	21.6
1024 (32 × 32)	12.50	1.91	22.0
2048 (32 × 64)	6.25	1.36	22.5

Table 6-4. Sinc4 + Sinc1 Filter Noise Performance (V_{REF} = 4.096V) (continued)

OSR	DATA RATE (kSPS)	NOISE ($e_{n(RMS)}$, μV_{RMS}) [1]	EFFECTIVE RESOLUTION ±4.096V RANGE (Bits)
4000 (32 × 125)	3.20	0.98	23.0
8000 (32 × 250)	1.60	0.70	23.5
16000 (32 × 500)	0.80	0.51	23.9
26656 (32 × 833)	0.48	0.42	24.2
32000 (32 × 1000)	0.40	0.40	24.3
96000 (32 × 3000)	0.13	0.27	24.9
160000 (32 × 5000)	0.08	0.24	25.0
SPEED MODE 0 (f_{MOD} =0.8MHz)			
64 (32 × 2)	200.00	6.63	20.2
128 (32 × 4)	100.00	5.02	20.6
256 (32 × 8)	50.00	3.67	21.1
512 (32 × 16)	25.00	2.64	21.6
1024 (32 × 32)	12.50	1.90	22.0
2048 (32 × 64)	6.25	1.34	22.5
4000 (32 × 125)	3.20	0.98	23.0
8000 (32 × 250)	1.60	0.69	23.5
16000 (32 × 500)	0.80	0.49	24.0
26656 (32 × 833)	0.48	0.40	24.3
32000 (32 × 1000)	0.40	0.39	24.3
96000 (32 × 3000)	0.13	0.26	24.9
160000 (32 × 5000)	0.08	0.21	25.2

[1] High OSR values can yield varying noise results because of the limits of 24-bit quantization: 4.096V / 2^{23} = 0.488µV / code.

Table 6-5 and Table 6-6 summarize the noise performance and effective resolution of the custom 20SPS and 25SPS filter modes with simultaneous 50Hz/60Hz rejection.

Table 6-5. 20SPS Filter and 25SPS Filter Noise Performance (V_{REF} = 2.5V)

SPEED MODE	f_{CLK} (MHz)	DATA RATE (SPS)	NOISE ($e_{n(RMS)}$, μV_{RMS})	EFFECTIVE RESOLUTION ±2.5V RANGE (Bits)
3	12.8	20	0.24	24.3
2	6.4	20	0.16	24.9
1	1.6	20	0.29	24.0
0	0.8	20	0.36	23.7
3	12.8	25	0.26	24.2
2	6.4	25	0.17	24.8
1	1.6	25	0.30	24.0
0	0.8	25	0.40	23.6

Table 6-6. 20SPS Filter and 25SPS Filter Noise Performance (V_{REF} = 4.096V)

SPEED MODE	f_{CLK} (MHz)	DATA RATE (SPS)	NOISE ($e_{n(RMS)}$, μV_{RMS})	EFFECTIVE RESOLUTION ±4.096V RANGE (Bits)
3	12.8	20	0.31	24.7

Table 6-6. 20SPS Filter and 25SPS Filter Noise Performance (V_{REF} = 4.096V) (continued)

SPEED MODE	f_{CLK} (MHz)	DATA RATE (SPS)	NOISE ($e_{n(RMS)}$, μV_{RMS})	EFFECTIVE RESOLUTION ±4.096V RANGE (Bits)
2	6.4	20	0.19	25.4
1	1.6	20	0.31	24.7
0	0.8	20	0.39	24.3
3	12.8	25	0.29	24.8
2	6.4	25	0.19	25.4
1	1.6	25	0.32	24.6
0	0.8	25	0.43	24.2

6.13 TUE (Total Unadjusted Error) Measurement

TUE (Total Unadjusted Error) defines the maximum deviation between the actual and the ideal transfer functions without performing any system-level calibration for gain or offset errors. TUE provides a single, worst-case accuracy number for the device or system. As shown in Equation 16, TUE is the ratio of the measured error (delta between the measurement and the expected value) and the full scale input range FSR of the device in %. If only a subsection of the full range is utilized in a specific application, a reduced range can be used in Equation 16 to calculate TUE for this particular system.

$$\text{TUE (\%FSR)} = \left(\left(V_{Measured} - V_{Ideal/Expected}\right) / \text{FSR} \right) \times 100 \tag{16}$$

where:

- $V_{Measured}$ = Measured output
- $V_{Ideal/Expected}$ = Calculated/expected/ideal measurement result
- FSR = full scale input range of the device, or alternatively a sub-range, selected according to the application

7 Detailed Description

7.1 Overview

The ADS125P08 is a multiplexed, high performance, 24 bit delta-sigma (ΔΣ) analog-to-digital converter (ADC) offering an excellent combination of dc accuracy and ac precision. The device is optimized to provide high resolution with low power consumption. Integrated high-impedance input buffers and a reference buffer simplify the driver requirements.

The *Functional Block Diagram* shows the features of the ADS125P08.

Following the input multiplexer (Mux), the device features two high-impedance, rail-to rail input buffer for the positive and negative ADC input.

The delta-sigma modulator produces low-resolution, high-frequency data proportional to the signal magnitude. Noise shaping within the modulator shifts the quantization noise of the low-resolution data to an out-of-band frequency range where the noise is removed by the digital filter. The noise remaining within the pass band is white, which is reduced by the digital filter. The digital filter simultaneously decimates and filters the modulator data to provide the high-resolution final output data. The modulator is a third-order, multibit delta-sigma design that measures the differential input signal, $V_{IN} = (V_{AINP} – V_{AINN})$, against the differential reference, $V_{REF} = (V_{REFP} – V_{REFN})$.

The digital filter offers several filter configurations: sinc3, sinc4, the option of a cascaded sinc1 stage following the sinc4 (sinc4 + sinc1), and a 50/60Hz notch filter option, allowing optimization between noise performance and latency. Programmable oversampling ratio (OSR) and four speed modes allow optimized choices of bandwidth, resolution, and device power consumption.

The ADS125P08 is equipped with a channel sequencer that automatically steps through the configured multiplexer inputs, selects them for measurement, and starts ADC conversions. The device also features a FIFO (first-in, first-out) buffer to store ADC conversion results and status information until the host controller is ready to read data from the device.

The SPI-compatible serial interface is used to configure the device and read conversion data. The interface features daisy-chaining capability for convenient connection of multichannel systems. Integrated cyclic redundancy check (CRC) error monitoring improves system-level reliability.

The main clock for the ADS125P08 is either provided by the internal 25.6MHz oscillator or by an external clock provided at the CLKIN pin. The START pin synchronizes the digital filter process. The $\overline{\text{RESET}}$ pin resets the ADC. $\overline{\text{DRDY}}$ is the conversion data-ready output signal.

Supply voltage AVDD powers the input buffers, input sampling switches and the modulator via an internal regulator (CAPA). Supply voltage IOVDD is the digital I/O voltage that also powers the digital core via an internal regulator (CAPD). The internal regulators minimize overall power consumption and provide consistent levels of performance.

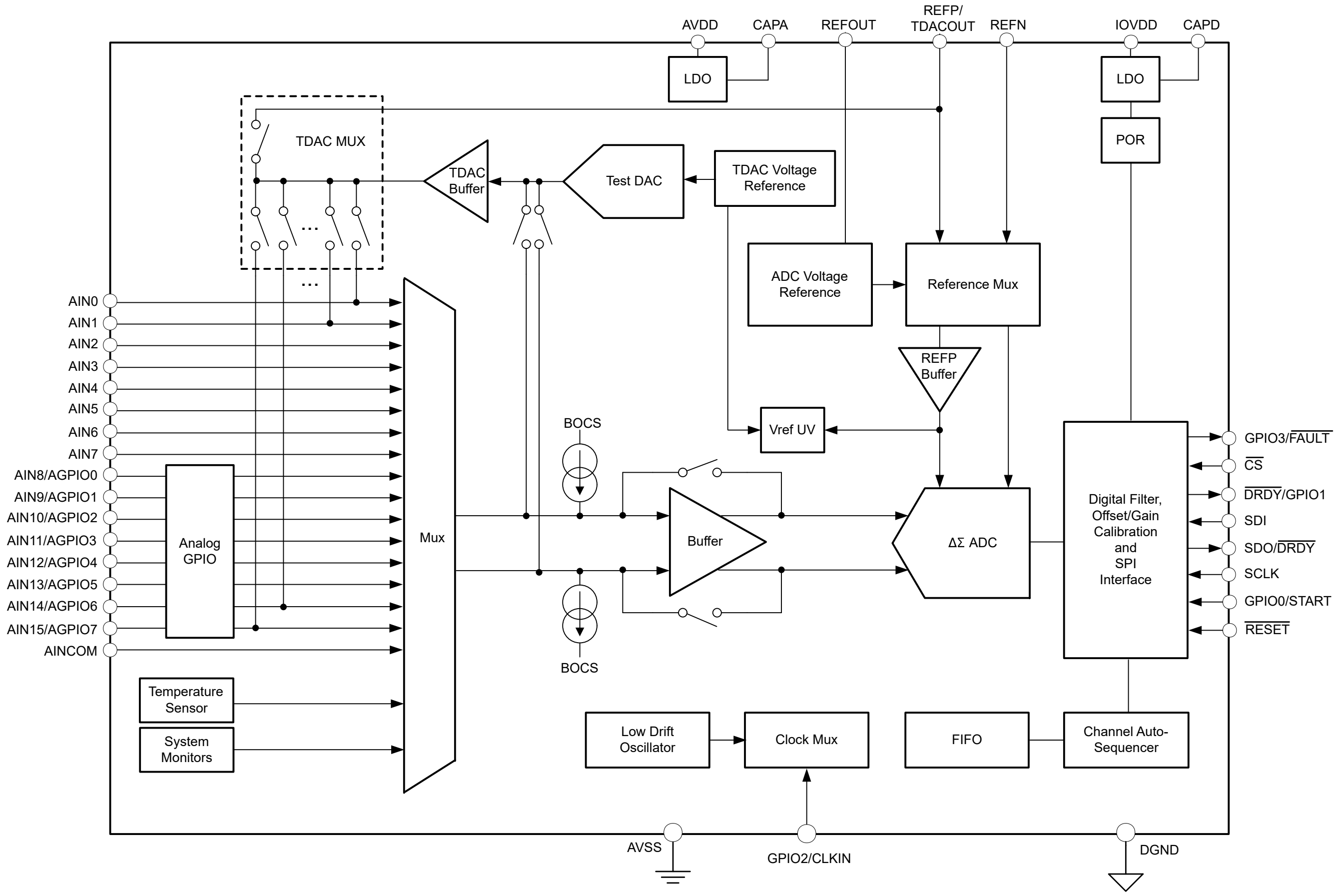
Several monitoring and diagnostic features are integrated in the device to mitigate and detect random hardware faults to aid in the development of functionally safe systems, such as:

- Supply and voltage reference undervoltage monitors
- Supply and voltage reference readback capability through the ADC
- Open-wire detection current source and sink
- Modulator overrange monitor
- Integrated temperature sensor
- Integrated Test DAC for generating test voltages to diagnose the ADC and the input multiplexer
- Cyclic redundancy check (CRC) on the SPI to achieve high data integrity for the communication
- Page or register access fault monitor
- FIFO overflow and underflow monitor
- FIFO CRC failure detector
- FIFO depth indicator
- Register and memory map CRC

- ADC conversion and sequence counters

The device offers up to eight GPIOs with logic levels based on AVDD (analog GPIOs: AGPIO0 through AGPIO7), and up to four GPIOs with logic levels based on IOVDD (GPIO0 through AGPIO3) . GPIO0 can alternatively be configured as a START input, GPIO1 is configured as a DRDY output by default, GPIO2 can be configured as a CLKIN input, and GPIO3 can be configured as a FAULT output.

7.2 Functional Block Diagram



7.3 Feature Description

7.3.1 Input Multiplexer

The ADS125P08 has 17 voltage input pins, AIN0 to AIN15 and VINCOM. Each pin connects to the internal multiplexer. The multiplexer enables these inputs to be configured as input pairs. The multiplexer output connects to the input of the integrated true rail-to-rail buffers. The ADS125P08 can have up to 16 active channels. The ADS125P08 can be set up to have sixteen single-ended inputs, eight fully differential inputs or a combination of singe-ended and differential inputs. When the channel auto-sequencer is enabled, the channels are automatically sequenced in the order given by the sequencer configuration. See the *Channel Auto-Sequencer* section for details.

The simplified circuit of Figure 7-1 represents the analog input structure including input multiplexer, ESD diodes and rail-to rail input buffer.

The input multiplexer controls which signals are routed to the positive and negative inputs of the ADC in each sequence step. Configure the input multiplexer using the STEPx_AINP[4:0] and STEPx_AINN[4:0] bits (x = 0 to 31), as well as the STEPx_SYS_MON[3:0] bits (x = 0 to 31).

The input multiplexer allows the following inputs to be connected to the ADC:

- Any of the sixteen analog inputs, AIN0 to AIN15, in single-ended measurement configuration when AINCOM is selected as the negative multiplexer input.
- Any pair of the sixteen analog inputs, AIN0 to AIN15, in differential measurement configuration when one input is connected to the positive ADC input and the other input to the negative ADC input.
- Internal short to AVSS. Use this setting for self-offset calibration of the ADC. Use the STEPx_SYS_MON[3:0] bits (x = 0 to 31) to select the internal short.
- Any of the internal system monitors such as the analog supply (AVDD-AVSS)/3, the digital supply (IOVDD-DGND)/3, internal subregulated supplies (CAPA-AVSS) or (CAPD-AVSS), or the reference voltage (REFP-REFN)/3. Use the STEPx_SYS_MON[3:0] bits (x = 0 to 31) to select the system monitors.
- Internal temperature sensor. Select the internal temperature sensor using the STEPx_SYS_MON[3:0] bits (x = 0 to 31).
- DC test signal provided by the Test DAC. Select the Test DAC signal using the STEPx_TDAC_VAL[4:0] and STEPx_TDAC_SEL[4:0] bits (x = 0 to 31).

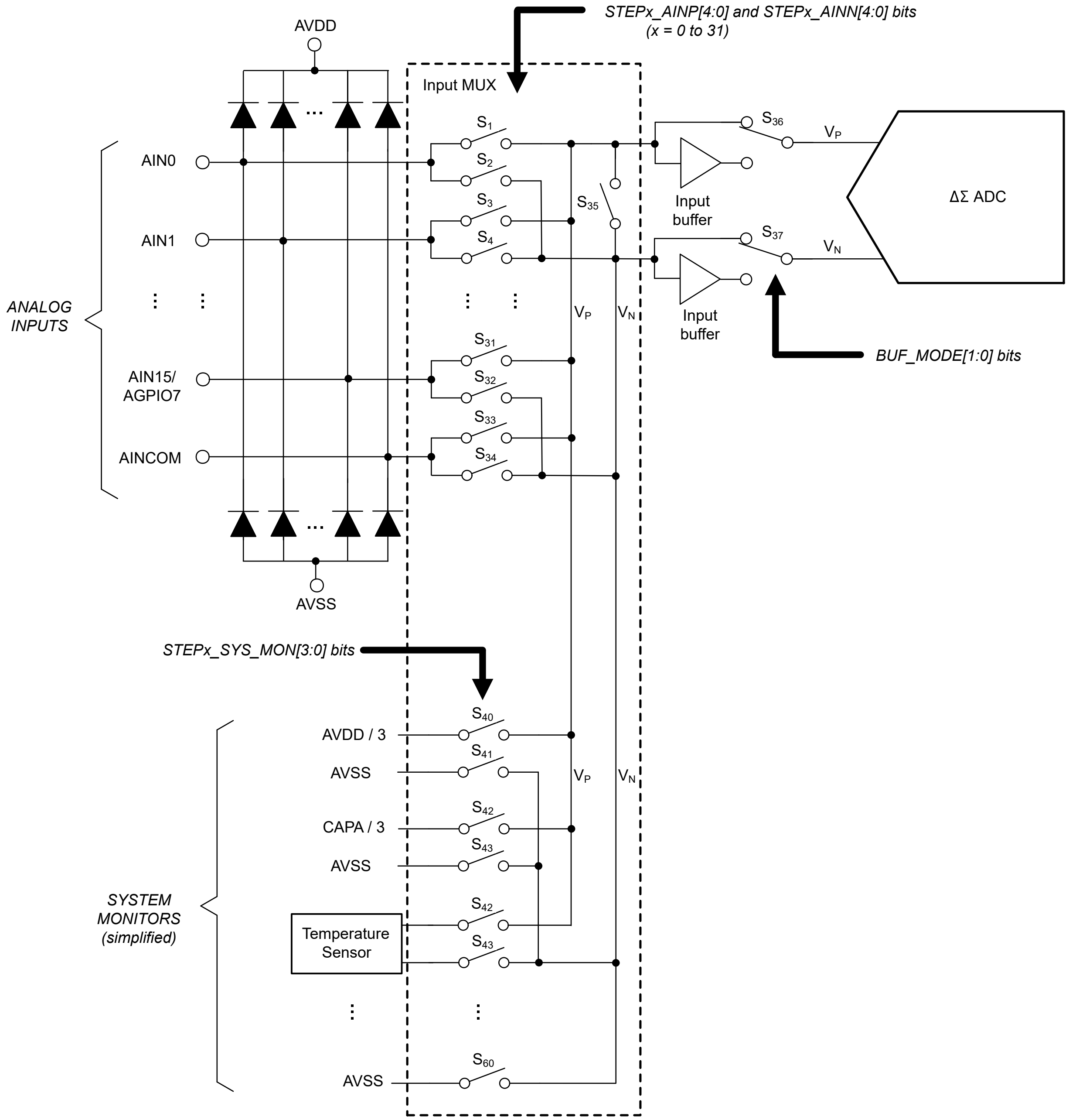


Figure 7-1. Input Multiplexer

Table 7-1 and Table 7-2 show the switch configurations of the input multiplexer circuit depicted in the *Input Multiplexer* section.

Table 7-1. Input Multiplexer Configurations, Positive ADC Input

STEPx_AINP[4:0] (x = 0 to 31)	SWITCH	DESCRIPTION
00000b	S_1	Select AIN0
00001b	S_3	Select AIN1
00010b	S_5	Select AIN2
00011b	S_7	Select AIN3
...	...	...
01110b	S_{29}	Select AIN14

Table 7-1. Input Multiplexer Configurations, Positive ADC Input (continued)

STEPx_AINP[4:0] (x = 0 to 31)	SWITCH	DESCRIPTION
01111b	S_{31}	Select AIN15
10000b	S_{33}	Select AINCOM
all other codes	N/A	All switches open

Table 7-2. Input Multiplexer Configurations, Negative ADC Input

STEPx_AINN[4:0] (x = 0 to 31)	SWITCH	DESCRIPTION
00000b	S_2	Select AIN0
00001b	S_4	Select AIN1
00010b	S_6	Select AIN2
00011b	S_8	Select AIN3
...	...	...
01110b	S_{30}	Select AIN14
01111b	S_{32}	Select AIN15
10000b	S_{34}	Select AINCOM
all other codes	N/A	All switches open

The analog input of the ADC is differential, with the input defined as a difference voltage: $V_{IN} = V_{AINP} – V_{AINN}$, where AINP and AINN represent any selected pair of the analog inputs. For best performance, drive the input with a differential signal with the common-mode voltage centered to mid-supply (AVDD + AVSS) / 2.

The ADC can accept either unipolar or bipolar input signals by configuring AVDD and AVSS accordingly. Unipolar Differential Input Signal shows an example of a differential signal with the supplies configured to unipolar operation. Symmetric input voltage headroom is available when the common-mode voltage is at mid-supply (AVDD / 2). Use AVDD = 5 V and AVSS = 0V for unipolar operation (AVDD can be reduced to 3V in the lower speed modes). Bipolar Differential Input Signal shows an example of a differential signal configured for bipolar operation. The common-mode voltage of the signal is normally at 0V. Use AVDD and AVSS = ±2.5V for bipolar operation (AVDD and AVSS can be reduced to ±1.5V in the lower speed modes).

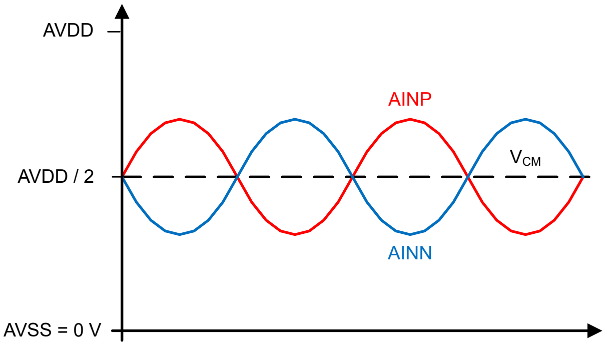


Figure 7-2. Unipolar Differential Input Signal

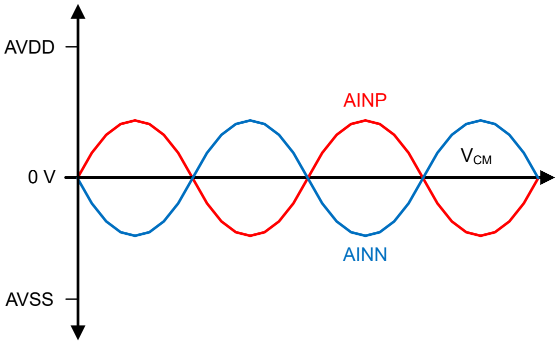


Figure 7-3. Bipolar Differential Input Signal

In both bipolar and unipolar power-supply configurations, the ADC can accept single-ended input signals by tying the AINN input to AVSS or ground, or to mid-supply. However, because AINN is now a fixed voltage, the voltage range of the ADC is limited by the input swing range of AINP (±2.5V or 0V to 5V for a 5V supply).

Diodes as shown in the *Input Multiplexer* section protect the ADC inputs from electrostatic discharge (ESD) events that occur during the manufacturing process and during printed circuit board (PCB) assembly when manufactured in an ESD-controlled environment. If the inputs are driven below AVSS – 0.3V, or above AVDD

+ 0.3V, the protection diodes can conduct. If these conditions are possible, use external clamp diodes, series resistors, or both to limit the input current to the specified value.

7.3.2 High-Impedance Input Buffers

The ADS125P08 features a rail-to-rail, integrated precision unity gain buffer ("HiZ input buffer") on each ADC analog input as shown in the *Input Multiplexer* section. The buffers provide the benefit of high input impedance with less than 30nA typical input current, allowing high impedance sources to be connected directly to the ADC analog inputs. The buffers fully drive the internal ADC switch capacitor sampling network, and thus simplifying the analog front-end circuit requirements.

The buffers use an optional level shifting scheme to support true ground inputs at the expense of allowable input voltage to the upper rail. A negative charge pump provides a negative supply voltage for the buffer stages in front of each ADC input to allow signal measurements below ground.

Use the BUF_MODE[1:0] bits to configure the HiZ input buffers. Table 7-3 shows the configurations for the input buffers.

Table 7-3. HiZ Input Buffer Configurations

BUF_MODE[1:0] bits	INPUT BUFFER CONFIGURATION	INPUT RANGE MIN	INPUT RANGE MAX
00b	Both HiZ input buffers off	AVSS + 0.1V	AVDD – 0.1V
01b	Only P HiZ input buffer turned on without level shift	AVSS + 0.1V	AVDD – 0.1V
10b	Both P and N HiZ input buffers turned on with level shift	AVSS – 0.05V	AVDD – 0.2V
11b	Both P and N HiZ input buffers turned on without level shift	AVSS + 0.1V	AVDD – 0.1V

7.3.3 Input Range

The ADC has two programmable input ranges: 1x (standard range) and 1.25x (extended range), where the 1x range is defined by V_{IN} = ±V_{REF} and the 1.25x range is defined by V_{IN} = ±1.25 × V_{REF}. In the 1.25x (extended range) mode, the input range is extended by 25% to provide signal headroom before clipping of the signal occurs. In this mode, output data are scaled such that the positive and negative full-scale output codes are at ±1.25 × V_{REF}. Configure the input range using the STEPx_EXT_RNG bit (x = 0 to 31).

ADC Input Range summarizes the ADC input range options.

Table 7-4. ADC Input Range

STEPx_EXT_RNG bit	INPUT RANGE (V)
0	±V_{REF}
1	±1.25×V_{REF}

The best available SNR performance is achieved by using a 4.096V or 5V reference voltage, which gives a typical improvement of 4dB compared to a 2.5V reference voltage.

The SNR performance degrades when the signal exceeds 110% standard full-scale range because of modulator saturation. The MOD_OVR_FAULTn bit indicates when modulator saturation occurs. Extended Range SNR Performance shows SNR performance when operating in the extended range.

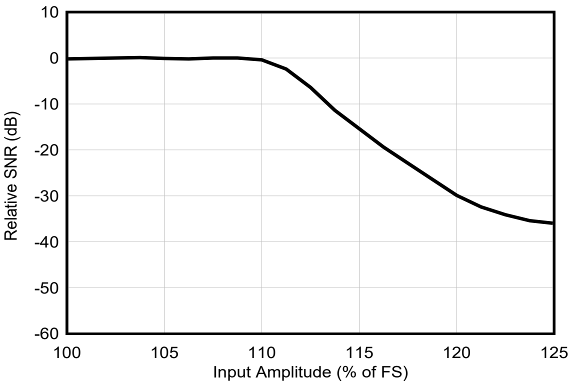


Figure 7-4. Extended Range SNR Performance

7.3.4 ADC Reference Voltage

A reference voltage is required for operation. The reference voltage input is differential, defined as: V_{REF} = V_{REFP} – V_{REFN}, as applied to the REFP and REFN inputs. The reference voltage is provided internally by the internal reference, or externally at the external reference input pins REFP and REFN.

Figure 7-5 shows the block diagram of the reference input and the reference multiplexer. Use the STEPx_REF_SEL bit to select the internal reference or the external reference inputs for each sequence step. When any of the system monitors (except the internal short) is selected as an input during a sequence step, the internal reference is selected and the STEPx_REF_SEL bit is ignored for this step.

If the internal reference is selected for any sequence step (enabled or disabled), or if a system monitor other than the internal short is selected for any sequence step, then the internal reference is enabled and available at the REFOUT pin. Otherwise, if all sequence steps are either using the external reference or internal short setting of the system monitor,the internal reference is disabled (powered down) and not available at REFOUT.

The internal reference is configurable to provide a 2.5V or a 4.096V reference voltage. The default value for the internal reference is 2.5V. Set the REF_VAL bit to 1b to select the 4.096V reference voltage.

As shown in Figure 7-5, the reference inputs have an input structure similar to the analog inputs. ESD diodes protect the reference inputs. To keep these diodes from turning on, make sure the voltages on the reference pins do not go below AVSS by more than 0.3V, or above AVDD by 0.3V. If these conditions are possible, use external clamp diodes, series resistors, or both to limit the input current to the specified value. Tie the REFN input pin externally to AVSS for stable performance.

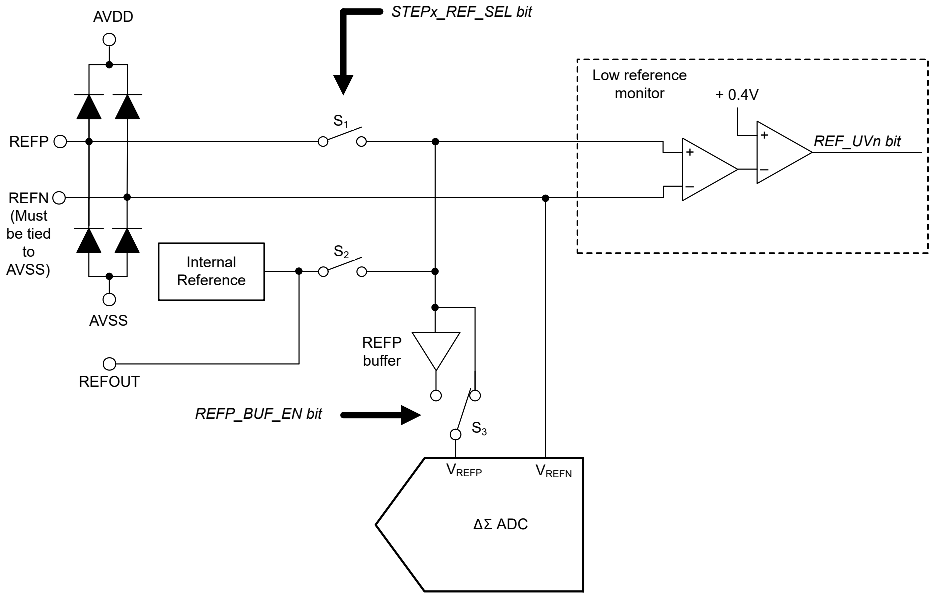


Figure 7-5. Reference Input and Reference Multiplexer Circuit

The reference voltage is sampled by a sampling capacitor C_{REF}. In unbuffered mode, current flows through the reference inputs to charge the sampling capacitor. The current consists of a dc component and an ac component that varies with the frequency of the modulator sampling clock. The effect of charging the reference sampling capacitor requires the external reference driver to settle at the end of the sample phase t = 1 / (2 × f_{MOD}). Incomplete settling of the reference voltage can lead to excessive gain error and gain error drift. Operation in a lower speed mode reduces the modulator sampling clock frequency, therefore allowing more time for the reference driver to settle.

The ADC provides a reference input buffer option for the REFP input to reduce the charge drawn by the sampling capacitor. Because the buffer reduces the charge demand of the sampling capacitor, the bandwidth requirement of the external driver is greatly reduced. Use the REFP_BUF_EN bit to configure the REFP input buffer, as shown in Figure 7-5. Many applications either ground REFN, or connect REFN to AVSS. A reference input buffer for REFN is not necessary because the REFN pin must be connected to AVSS for normal operation.

The device incorporates a low-reference monitor to detect a low or missing reference, as shown in Figure 7-5. If the differential reference voltage (V_{REF} = V_{REFP} – V_{REFN}) falls below 0.4V (typical), the low reference alarm triggers. The REF_UVn bit indicates the low reference alarm. Use the low-reference monitor to detect a missing or failed reference voltage connection. Connect a 100kΩ resistor across the reference inputs to provide the necessary bias. If either reference input is missing or unconnected, this external resistor biases the reference inputs to each other. The low-reference monitor is a fast-responding analog comparator; therefore, transients in the reference voltage can trigger the alarm.

7.3.5 Power Supplies

The device requires two power supplies: analog (AVDD) and digital (IOVDD). The analog power supply can be independently chosen from the digital power supply. The IOVDD supply sets the logic levels for the serial interface pins ($\overline{\text{CS}}$, SCLK, SDI, SDO/$\overline{\text{DRDY}}$) and other digital I/O pins. The AVDD supply sets the logic levels for the GPIOs (GPIO0 to GPIO3).

7.3.5.1 AVDD and AVSS

AVDD and AVSS power the analog circuitry of the device. The ADS125P08 can be configured either for bipolar input operation (such as using ±2.5V power supplies), or for unipolar input operation (such as AVDD = 5V and AVSS = DGND). Use a parallel combination of 1µF and 0.1µF bypass capacitors across the AVDD supply voltage and the AVSS pin, with a 3Ω series resistor placed in series between the capacitors and the AVDD pin. Place the resistor and capacitors as close as possible to the AVDD pin.

Table 7-5 shows examples of AVDD and AVSS power-supply configurations.

Table 7-5. AVDD and AVSS Power-Supply Configuration Examples (All Voltages with Respect to DGND)

SUPPLY CONFIGURATION	SPEED MODE 3 or SPEED MODE 2		SPEED MODE 1 or SPEED MODE 0	
	AVDD	AVSS	AVDD	AVSS
Unipolar	5V	0V	3V to 5V	0V
Bipolar	2.5V	–2.5V	1.5V to 2.5V	–1.5V to –2.5V

7.3.5.2 IOVDD

IOVDD is the digital I/O supply voltage pin for the device. IOVDD is internally regulated to 1.25V to supply power to the digital core. Bypass IOVDD to DGND with a parallel combination of 1µF and 0.1µF capacitors. The voltage level of IOVDD is independent of the analog supply configuration.

7.3.5.3 CAPA and CAPD

CAPA and CAPD are the output voltages of the internal analog and digital voltage regulators. The regulators are used to reduce the supply voltage to operate internal sub-circuits and are not designed to drive external loads. CAPA is the analog regulator voltage output and is powered from AVDD. The output voltage is 1.6V with respect to AVSS. Bypass CAPA with a 1µF capacitor to AVSS. CAPD is the digital regulator voltage output, powered from IOVDD. The regulator output is 1.25V with respect to DGND. Bypass CAPD with a 1µF capacitor to DGND.

7.3.5.4 Power-On Reset (POR)

The ADC uses power-supply monitors to detect power-up and supply brownout events. Power-up or power-cycling of the IOVDD digital supply results in device reset. Power-up or power-cycling of the analog power supplies does not reset the ADC.

Figure 7-6 shows the digital power-on thresholds of the IOVDD and the internal CAPD voltages. When the voltages are below the respective thresholds, the ADC is reset (using an internal POR signal) and the AVDD_UVn flag is set to 0b indicating a brownout condition. At power-up, AVDD_UVn flag is set to 0b and $\overline{\text{DRDY}}$ later transitions high when the SPI is ready for communication. Once the device is ready for SPI communication, write 1b to clear the AVDD_UVn flag.

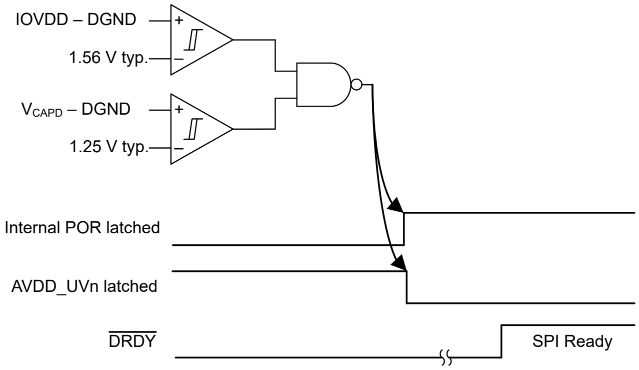


Figure 7-6. Digital Supply Threshold

Figure 7-7 shows the power-on thresholds of the analog power supplies. Three monitors are used for four analog supply voltage conditions (AVDD – DGND), (AVDD – AVSS), and (CAPA – AVSS). Valid conversion data are available after all power supplies and the reference voltage are stabilized after power-on. The AVDD_UVn bit of the ADC_REF_STATUS register sets to 0b when any analog power voltage falls below the respective threshold. Write 1b to clear the bit to detect the next analog supply low-voltage condition. Power cycling the analog power supplies does not reset the ADC. Because a low voltage on the IOVDD supply also resets the internal analog LDO (CAPA), the analog low-voltage flag (AVDD_UVn) is also set.

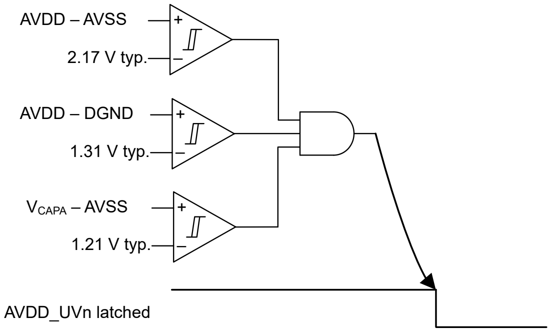


Figure 7-7. Analog Supply Threshold

7.3.6 Clock Operation

The ADS125P08 requires a main clock (CLK) for the ADC to operate. As shown in Figure 7-8, the clock to the ADC is provided in one of two ways:

- By the internal 25.6MHz oscillator, or
- By an external clock on the GPIO2/CLKIN pin

The CLK_SEL bit selects the clock source for the ADC. At device power-up or after device reset, the internal oscillator is selected as the clock source by default.

A clock divider divides the external clock frequency f_{CLKIN} by a factor of one, two, eight or sixteen to create the main clock frequency f_{CLK}, as shown in Figure 7-8. Use the CLK_DIV[1:0] bits to configure the clock divider. See the *External Clock* section for details.

The frequency of the internal oscillator automatically scales to the speed mode operation as selected by the SPEED_MODE[1:0] bits. See the *Internal Oscillator* section for details.

The modulator clock of the delta-sigma ADC is derived from the main clock. A clock divider divides the main clock frequency (f_{CLK}) by a factor of two to create the modulator frequency (f_{MOD} = f_{CLK} / 2) with a duty cycle of 50%.

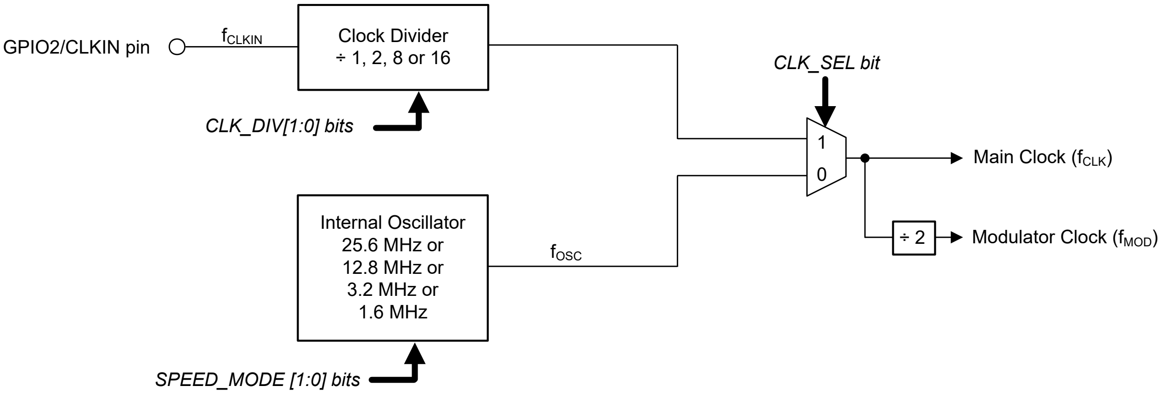


Figure 7-8. Clock Block Diagram

The nominal value of f_{CLK} is 25.6MHz in speed mode 3, 12.8MHz in speed mode 2, 3.2MHz in speed mode 1 and 1.6MHz in speed mode 0. Table 7-6 shows the nominal clock frequencies for the respective speed modes and the corresponding data rates at the minimum OSR setting.

Table 7-6. ADC Clock Frequency

SPEED_MODE[1:0] bits	SPEED MODE	NOMINAL CLOCK FREQUENCY (MHz)	MAXIMUM RATED DATA RATE (kSPS)
11b	Speed mode 3	25.6	1066.6
10b	Speed mode 2	12.8	533.3

Table 7-6. ADC Clock Frequency (continued)

SPEED_MODE[1:0] bits	SPEED MODE	NOMINAL CLOCK FREQUENCY (MHz)	MAXIMUM RATED DATA RATE (kSPS)
01b	Speed mode 1	3.2	133.333
00b	Speed mode 0	1.6	66.67

Before changing the clock source using the CLK_SEL bit, set the device in powerdown mode to prevent clock glitching during the clock switchover. When switching from an external clock source to the internal oscillator, keep the external clock running until after the device switched over to the internal main oscillator.

7.3.6.1 Internal Oscillator

At power-up and after reset, the ADC defaults to internal oscillator mode (CLK_SEL bit = 0b). The frequency of the internal oscillator automatically scales to the selected speed mode operation, as shown in Table 7-7. The CLK_DIV[1:0] bits are ignored when the internal oscillator is selected as the clock source. Because of the clock jitter associated with the internal oscillator, only use the internal oscillator for dc signal measurements. AC signal measurements are not recommended when using the internal oscillator.

Table 7-7. Internal Clock Frequency Settings

SPEED_MODE[1:0] bits	SPEED MODE	f_{OSC} (MHz), f_{CLK} (MHz)	f_{MOD}
11b	Speed mode 3	25.6	12.8
10b	Speed mode 2	12.8	6.4
01b	Speed mode 1	3.2	1.6
00b	Speed mode 0	1.6	0.8

7.3.6.2 External Clock

To operate the ADC with an external clock, configure the GPIO2/CLKIN pin to a clock input using the GPIO2_CFG[1:0] bits. Then program the CLK_SEL bit to 1b and apply the clock signal to the CLK pin. The clock can be decreased from the nominal clock frequency to yield specific data rates. Configure the external clock divider as shown in Table 7-8 using the CLK_DIV[1:0] bits. However, the conversion noise when operating at the reduced clock frequency is the same as the higher clock frequency. Reducing the conversion noise is only possible by increasing the OSR value or changing the filter mode.

Table 7-8. External Clock Divider Settings

CLK_DIV[1:0]	DIVIDE BY	f_{CLK} (MHz) for f_{CLKIN} = 25.6MHz	f_{MOD} (MHz) for f_{CLKIN} = 25.6MHz
00b	1	25.6	12.8
01b	2	12.8	6.4
10b	8	3.2	1.6
11b	16	1.6	0.8

Clock jitter results in timing variations in the modulator sampling that leads to degraded SNR performance. A low-jitter clock is essential to meet data sheet SNR performance. For example, with a 200kHz signal frequency, an external clock with < 10ps (rms) jitter is required. For lower signal frequencies, the clock jitter requirement can be relaxed by –20dB per decade of signal frequency. For example, with f_{IN} = 20kHz, a clock with 100ps jitter can be used. Many types of RC oscillators exhibit high levels of jitter and must be avoided for ac signal measurements. Instead, use a crystal-based clock oscillator as the clock source. Avoid ringing on the clock input. A series resistor placed at the output of the clock buffer often helps reduce ringing.

7.3.7 Modulator

The ADS125P08 uses a switched-capacitor, third-order, singe-loop modulator with a 5-bit internal quantizer. This modulator topology achieves excellent noise and linearity performance while consuming very low power. As with most high-order modulators driven by high amplitude out-of-band signals, modulator saturation can occur. When saturated, the in-band signal still converts, however the noise floor increases. Figure 7-9 illustrates the amplitude limit of out-of-band signals to avoid modulator saturation. The limit of dc and in-band signal amplitudes are 1dB above standard full scale.

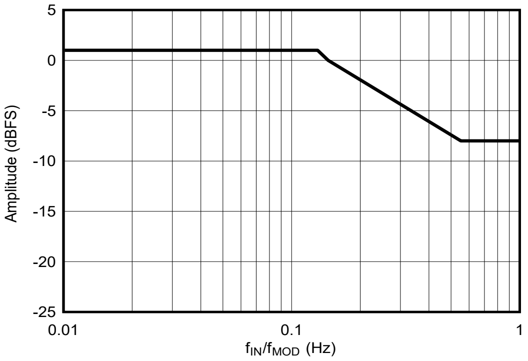


Figure 7-9. Amplitude Limit to Avoid Modulator Saturation

Modulator saturation is indicated by the MOD_OVR_FAULTn bit of the ADC_REF_STATUS register. The modulator saturation status is latched during the conversion period and is cleared by writing 1b to the register bit. Modulator saturation resulting from out-of-band signals can be avoided by using an antialias filter at the ADC inputs.

7.3.8 Digital Filter

The digital filter averages and decimates the low-resolution, high-speed data from the modulator to produce high-resolution, low-speed output data. The programmable oversampling ratio (OSR) determines the amount of filtering that affects the signal bandwidth and conversion noise, and the output data rate through decimation. The output data rate is defined by: $f_{DATA} = f_{MOD}$ / OSR.

The digital filter is a cascaded-integrator-comb (CIC) topology that minimizes the delay (latency) as the conversion data propagates through the filter. The CIC filter is otherwise known as a sinc filter because of the characteristic sinx/x (sinc) frequency response. The short latency time makes the filter designed for fast acquisition of dc signals or for use in control loops.

As shown in Figure 7-10, the device offers programmable OSR and several filter configurations: sinc3, sinc4, the option of a cascaded sinc1 stage following the sinc4 (sinc4 + sinc1), and a 50/60Hz notch filter option. The configurations of the digital filter allow trade-offs between acquisition time, noise performance, and line-cycle rejection.

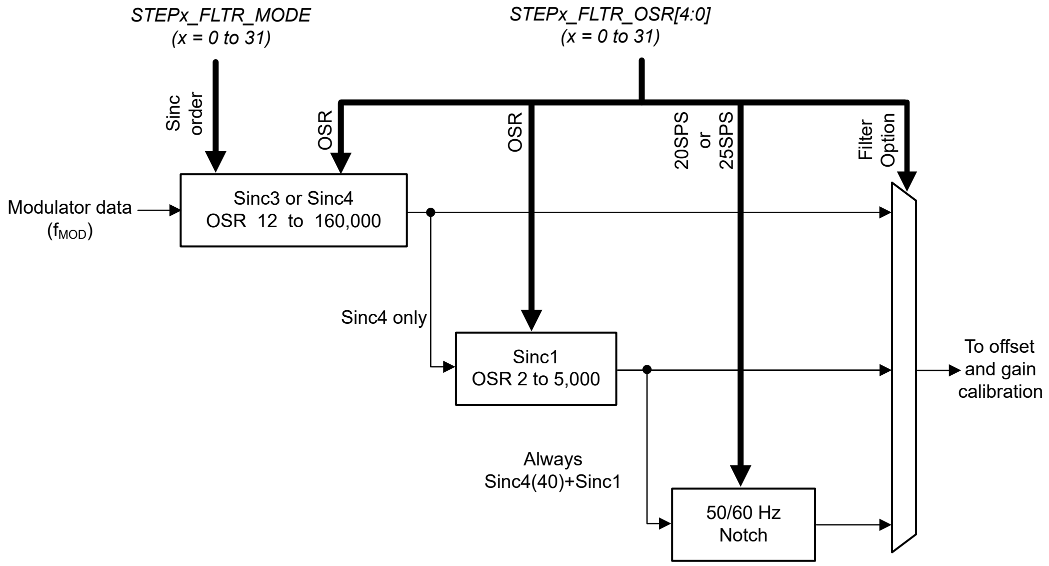


Figure 7-10. Digital Filter Block Diagram

The available filter options are:

- Sinc3 only with variable OSR from 12 to 160,000 (STEPx_FLTR_MODE = 1b)
- Sinc4 only with variable OSR from 12 to 160,000 (STEPx_FLTR_MODE = 0b)
- Sinc4 with OSR = 32 followed by sinc1 with variable OSR from 2 to 5,000: For this option, the STEPx_FLTR_MODE bit that selects sinc3 or sinc4 is ignored because the sinc4 filter is always used in the first stage.
- Sinc4 with OSR = 40 followed by sinc1 with preset OSRs followed by a 50/60Hz Notch FIR filter: For this option, two data rates are available, i.e. a 20SPS data rate or a 25SPS data rate. See the *50/60Hz Notch Filters* section for details.

The ADS125P08 controls ADC conversion by means of a highly flexible channel auto-sequencer, see the *Channel Auto-Sequencer* section for details. The filter configuration is individually programmable for each sequence step. The OSR is set by the STEPx_FLTR_OSR[4:0] bits (x = 0 to 31) in the STEPx_FLTR1_CFG registers, and the order of the sinc filter (sinc3 or sinc4) is set by the STEPx_FLTR_MODE bit in the STEPx_FLTR1_CFG registers. See the *Configuring the Auto-Sequencer* section for details on how to configure the filter for each sequencer step individually.

Equation 17 is the general expression of the sinc-filter frequency response. For single-stage sinc filter options (for example, the single-stage sinc3 or sinc4 filter), the second term is not used.

$$\left|H_{(f)}\right| = \left|\frac{\sin\left[\frac{A\pi f}{f_{MOD}}\right]}{A\sin\left[\frac{\pi f}{f_{MOD}}\right]}\right|^{n} \cdot \left|\frac{\sin\left[\frac{AB\pi f}{f_{MOD}}\right]}{B\sin\left[\frac{A\pi f}{f_{MOD}}\right]}\right| \tag{17}$$

where:

- f = Signal frequency
- A = Stage 1 OSR
- B = Stage 2 OSR
- $f_{MOD} = f_{CLK} / 2$ = 12.8MHz (speed mode 3, nominal), 6.4MHz (speed mode 2, nominal), 1.6MHz (speed mode 1, nominal), 0.8MHz (speed mode 0 mode, nominal)
- n = Order of the stage 1 filter (3 or 4)

7.3.8.1 Digital Filter Latency

When starting or restarting conversions, the digital filter resets and requires a certain amount of time to provide settled output data. This time is called the latency time, $t_{LATENCY}$. The ADS125P08 hide the unsettled data internally and only indicate when settled conversion data are available, by means of a falling $\overline{\text{DRDY}}$ edge or the DRDY bit. Table 7-9 and Table 7-12 summarize the latency times for the various speed modes and digital filter settings. The latency times are measured from the rising $\overline{\text{CS}}$ edge of the register write frame where the START bit is set to 1b in idle mode (or the assertion/rising edge of the START pin), to the first falling $\overline{\text{DRDY}}$ edge. Because the $\overline{\text{CS}}$ signal in the SPI clock domain is latched by the digital filter logic running on the modulator clock domain, the latency times provided have an uncertainty of ±1 t_{MOD}. The conversion period for the second and all subsequent conversions equals $t_{DATA} = 1 / f_{DATA} = OSR / f_{MOD}$ as shown in Figure 7-11.

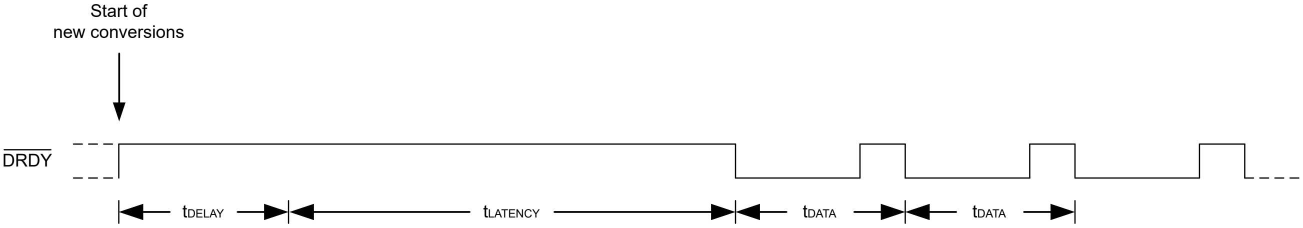


Figure 7-11. Latency Time and Conversion Period

The latency time increases in certain situations:

- when starting conversions from standby mode: adds 10 up to 33 t_{MOD} (this can depend on the speed mode).
- when restarting ongoing conversions by writing to a register which restarts conversions: the conversion is stopped, and the user needs to write the START bit (or assert the START pin) to start the conversion again.

In addition, a programmable delay time can be added to delay the start of the conversion cycle after the START bit is set (or after the START pin rising edge). This delay time allows for settling of external components, such as the voltage reference after exiting standby mode, or for additional settling time when switching the signal through the multiplexer. The delay time is only added to the first conversion after a conversion start as shown in Figure 7-11, or at the beginning of every new sequence step when the sequencer is active. In sequencer operation, the programmable delay time can be configured independently for each individual sequence step. See the *Conversion-Start Delay Time* section for details on the programmable delay time.

See also the *Auto-Sequencer and $\overline{\text{DRDY}}$ Behavior* section for more details on latency and settling when using the sequencer.

7.3.8.2 Sinc3 and Sinc4 Filters

The sinc filter averages and decimates the high-speed modulator data to produce high-resolution output data at reduced data rate. Increasing the OSR value decreases the data rate and simultaneously reduces signal bandwidth and conversion noise resulting from increased decimation and data averaging. Table 7-9 lists the sinc3 and sinc4 filter –3dB frequencies and latency times. The latency times (shown in µs) are given for the nominal clock frequencies, and the values scale with the clock frequency.

Table 7-9. Sinc3 and Sinc4 Filter Characteristics

SPEED MODE	f_{CLK} (MHz)	OSR	DATA RATE (kSPS)	–3dB FREQUENCY (kHz)		LATENCY TIME (µs)	
				SINC3	SINC4	SINC3	SINC4
3	25.6	12	1066.67	279.5	242.3	3.9	4.8
2	12.8		533.33	139.7	121.2	7.7	9.6
1	3.2		133.33	34.9	30.3	30.9	38.4
0	1.6		66.67	17.5	15.1	61.9	76.9

Table 7-9. Sinc3 and Sinc4 Filter Characteristics (continued)

SPEED MODE	f_{CLK} (MHz)	OSR	DATA RATE (kSPS)	–3dB FREQUENCY (kHz)		LATENCY TIME (µs)	
				SINC3	SINC4	SINC3	SINC4
3	25.6	16	800	209.6	181.8	4.8	6.1
2	12.8		400	104.8	90.9	9.6	12.1
1	3.2		100	26.2	22.7	38.4	48.4
0	1.6		50	13.1	11.4	76.9	96.9
3	25.6	24	533.33	139.7	121.2	6.7	8.6
2	12.8		266.67	69.9	60.6	13.4	17.1
1	3.2		66.67	17.5	15.1	53.4	68.4
0	1.6		33.33	8.7	7.6	106.9	136.9
3	25.6	32	400	104.8	90.9	8.6	11.1
2	12.8		200	52.4	45.4	17.1	22.1
1	3.2		50	13.1	11.4	68.4	88.4
0	1.6		25	6.6	5.7	136.9	176.9
3	25.6	64	200	52.4	45.4	16.1	21.1
2	12.8		100	26.2	22.7	32.1	42.1
1	3.2		25	6.6	5.7	128.4	168.4
0	1.6		12.5	3.3	2.8	256.9	336.9
3	25.6	128	100	26.2	22.7	31.1	41.1
2	12.8		50	13.1	11.4	62.1	82.1
1	3.2		12.5	3.3	2.8	248.4	328.4
0	1.6		6.25	1.6	1.4	496.9	656.9
3	25.6	256	50	13.1	11.36	61.1	81.1
2	12.8		25	6.55	5.68	122.1	162.1
1	3.2		6.25	1.64	1.42	488.4	648.4
0	1.6		3.13	0.82	0.71	976.9	1296.9
3	25.6	512	25	6.55	5.68	121.1	161.1
2	12.8		12.5	3.28	2.84	242.1	322.1
1	3.2		3.13	0.82	0.71	968.4	1288.4
0	1.6		1.56	0.41	0.35	1936.9	2576.9
3	25.6	1024	12.5	3.28	2.84	241.1	321.1
2	12.8		6.25	1.64	1.42	482.1	642.1
1	3.2		1.56	0.41	0.35	1928.4	2568.4
0	1.6		0.78	0.204	0.177	3856.9	5136.9
3	25.6	2048	6.25	1.638	1.42	481.1	641.1
2	12.8		3.13	0.82	0.711	962.1	1282.1
1	3.2		0.78	0.204	0.177	3848.4	5128.4
0	1.6		0.39	0.102	0.089	7696.9	10256.9
3	25.6	4000	3.2	0.838	0.727	938.6	1251.1
2	12.8		1.6	0.419	0.364	1877.1	2502.1
1	3.2		0.4	0.105	0.091	7508.4	10008.4
0	1.6		0.2	0.052	0.045	15016.9	20016.9
3	25.6	8000	1.6	0.419	0.364	1876.1	2501.1
2	12.8		0.8	0.21	0.182	3752.1	5002.1
1	3.2		0.2	0.052	0.045	15008.4	20008.4
0	1.6		0.1	0.026	0.023	30016.9	40016.9

Table 7-9. Sinc3 and Sinc4 Filter Characteristics (continued)

SPEED MODE	f_{CLK} (MHz)	OSR	DATA RATE (kSPS)	–3dB FREQUENCY (kHz)		LATENCY TIME (µs)	
				SINC3	SINC4	SINC3	SINC4
3	25.6	16000	0.8	0.21	0.182	3751.1	5001.1
2	12.8		0.4	0.105	0.091	7502.1	10002.1
1	3.2		0.1	0.026	0.023	30008.4	40008.4
0	1.6		0.05	0.013	0.011	60016.9	80016.9
3	25.6	26667	0.48	0.126	0.109	6251.1	8334.5
2	12.8		0.24	0.063	0.055	12502.3	16669
1	3.2		0.06	0.016	0.014	50009.1	66675.9
0	1.6		0.03	0.008	0.007	100018.1	133351.9
3	25.6	32000	0.4	0.105	0.091	7501.1	10001.1
2	12.8		0.2	0.052	0.045	15002.1	20002.1
1	3.2		0.05	0.013	0.011	60008.4	80008.4
0	1.6		0.03	0.008	0.007	120016.9	160016.9
3	25.6	96000	0.13	0.034	0.03	22501.1	30001.1
2	12.8		0.07	0.018	0.016	45002.1	60002.1
1	3.2		0.02	0.005	0.005	180008.4	240008.4
0	1.6		0.008	0.002	0.002	360016.9	480016.9
3	25.6	160000	0.08	0.021	0.018	37501.1	50001.1
2	12.8		0.04	0.01	0.009	75002.1	100002.1
1	3.2		0.01	0.003	0.002	300008.4	400008.4
0	1.6		0.005	0.001	0.001	600016.9	800016.9

Figure 7-12 and Figure 7-13 show the sinc filter frequency response. The frequency response consists of a series of response nulls occurring at f_{DATA} and multiples thereof. At the null frequencies, the filter has zero gain. Figure 7-13 shows the folding of the frequency response starting at f_{MOD} / 2. No attenuation is provided by the filter at input frequencies near n × f_{MOD} (n = 1, 2, 3, and so on).

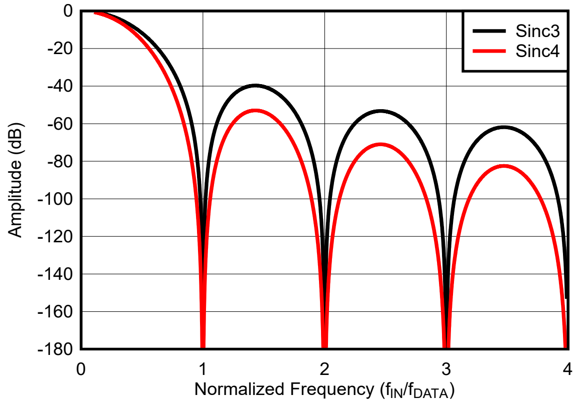


Figure 7-12. Sinc3 and Sinc4 Frequency Response

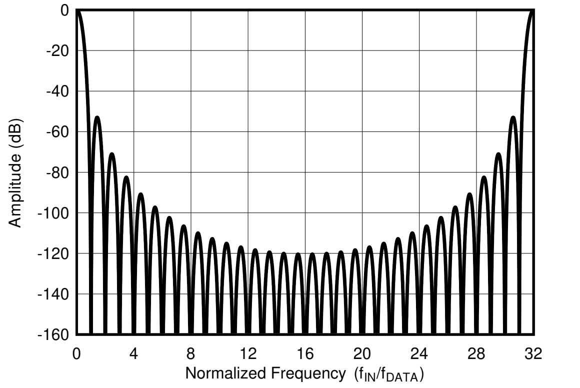


Figure 7-13. Sinc4 Frequency Response to f_{MOD} (OSR = 32)

Table 7-10 shows the normal-mode rejection of a few selected filter settings for data rates equal to common line-cycle frequencies.

Table 7-10. Normal-Mode Rejection

SPEED MODE [1]	OSR	f_{DATA} (SPS)	DIGITAL FILTER Response (dB)			
			2% CLOCK VARIATION		6% CLOCK VARIATION	
			SINC3 FILTER	SINC4 FILTER	SINC3 FILTER	SINC4 FILTER
1	96000	$16.\overline{6}$	–100dB	–135dB	–72dB	–95dB
1	32000	50				
1	26667	60				
1	8000	200				
1	4000	400				

[1] Nominal clock frequency for each speed mode is used: f_{CLK} = 25.6MHz (speed mode 3), 12.8MHz (speed mode 2), 3.2MHz (speed mode 1), 1.6MHz (speed mode 0).

If the ADC connection leads are in close proximity to industrial motors and conductors, coupling of 50Hz and 60Hz power line frequencies can occur. The coupled noise interferes with the signal voltage, and can lead to inaccurate or unstable conversions. The digital filter provides enhanced rejection of power-line coupled noise for data rates of 60 SPS and less. Program the filter to tradeoff data rate and conversion latency versus the desired level of line cycle rejection. Table 7-11 summarizes the 50Hz and 60Hz line-cycle rejection based on ±1Hz tolerance of power-line to ADC clock frequency, and additional clock tolerance of 0% (e.g. external clock) and 1% (e.g. internal clock). Best possible power line rejection is provided by the high-order sinc filter and by using an accurate ADC clock.

Table 7-11. 50Hz and 60Hz Line Cycle Rejection

SPEED MODE [1]	OSR	Filter type	f_{DATA} (SPS)	DIGITAL FILTER RESPONSE (dB)			
				50Hz ±1Hz		60Hz ±1Hz	
				CLOCK TOLERANCE: [2]			
				0%	1%	0%	1%
0	160000	Sinc4	5	-137.5	-126.1	-144.0	-131.0
0	160000	Sinc3	5	-103.1	-94.6	-108.0	-98.3
1	160000	Sinc4	10	-135.8	-122.1	-142.2	-126.5
1	160000	Sinc3	10	-101.8	-91.6	-106.7	-94.8
1	96000	Sinc4	$16.\overline{6}$	-135.4	-121.2	-84.0	-83.3
1	96000	Sinc3	$16.\overline{6}$	-101.6	-90.9	-63.0	-62.5
0	32000	Sinc4	25	-135.3	-121.0	-71.4	-71.3
0	32000	Sinc3	25	-101.5	-90.7	-53.5	-53.5
1	32000	Sinc4	50	-135.2	-120.8	-62.3	-61.1
1	32000	Sinc3	50	-101.4	-90.6	-46.7	-45.9
1	26667	Sinc4	60	-53.8	-52.1	-141.7	-125.0
1	26667	Sinc3	60	-40.4	-39.1	-106.3	-93.8
0	16000	Sinc4	50	-135.2	-120.8	-62.3	-61.1
0	16000	Sinc3	50	-101.4	-90.6	-46.7	-45.9

[1] Nominal clock frequency for each speed mode is used: f_{CLK} = 25.6MHz (speed mode 3), 12.8MHz (speed mode 2), 3.2MHz (speed mode 1), 1.6MHz (speed mode 0).
[2] 0% clock tolerance corresponds to external clock, 1% clock tolerance corresponds to internal clock.

7.3.8.3 Sinc4 + Sinc1 Cascade Filter

For selected data rates, the sinc4 filter offers the option of a cascade sinc1 filter section. Compared to a single-stage sinc3 or sinc4 filter, cascading the sinc1 filter shortens latency time when operated at the same data rate. However, the sinc3 and sinc4 filters provide greater rejection of interference signals close to the notch frequencies because of the wide frequency-rejection range at the data rate frequency. When operated in cascade mode, the OSR of the sinc4 stage is fixed at 32 (OSR = A) and the decimation of the sinc1 stage (OSR

= B) determines the output data rate. The first stage of the cascade filter is fixed to sinc4, meaning the sinc filter configuration set by the STEPx_FLTR_MODE bit in the STEPx_FLTR1_CFG registers is ignored (where x = 0 to 31 for the sequence step). Table 7-12 summarizes the cascade filter characteristics.

Table 7-12. Sinc4 + Sinc1 Cascade Filter Characteristics

SPEED MODE	f_{CLK} (MHz)	OSR (A × B) [1]	DATA RATE (SPS)	–3dB FREQUENCY (Hz)	LATENCY TIME (µs)
3	25.6	64 (32 × 2)	200000	88320	13.60
2	12.8		100000	44160	27.10
1	3.2		25000	11040	108.40
0	1.6		12500	5520	216.90
3	25.6	128 (32 × 4)	100000	44160	18.60
2	12.8		50000	22080	37.10
1	3.2		12500	5520	148.40
0	1.6		6250	2760	296.90
3	25.6	256 (32 × 8)	50000	22080	28.60
2	12.8		25000	11040	57.10
1	3.2		6250	2760	228.40
0	1.6		3125	1380	456.90
3	25.6	512 (32 × 16)	25000	11040	48.60
2	12.8		12500	5520	97.10
1	3.2		3125	1380	388.40
0	1.6		1562.5	690	776.90
3	25.6	1024 (32 × 32)	12500	5520	88.60
2	12.8		6250	2760	177.10
1	3.2		1562.5	690	708.40
0	1.6		781.25	345	1416.90
3	25.6	2048 (32 × 64)	6250	2760	168.60
2	12.8		3125	1380	337.10
1	3.2		781.25	345	1348.40
0	1.6		390.63	172.5	2696.90
3	25.6	4000 (32 × 125)	3200	1413.12	321.10
2	12.8		1600	706.56	642.10
1	3.2		400	176.64	2568.40
0	1.6		200	88.32	5136.90
3	25.6	8000 (32 × 250)	1600	706.56	633.60
2	12.8		800	353.28	1267.10
1	3.2		200	88.32	5068.40
0	1.6		100	44.16	10136.90
3	25.6	16000 (32 × 500)	800	353.28	1258.60
2	12.8		400	176.64	2517.10
1	3.2		100	44.16	10068.40
0	1.6		50	22.08	20136.90
3	25.6	26656 (32 × 833)	480.19	212.052	2091.10
2	12.8		240.1	106.028	4182.10
1	3.2		60.02	26.505	16728.40
0	1.6		30.01	13.252	33456.90

Table 7-12. Sinc4 + Sinc1 Cascade Filter Characteristics (continued)

SPEED MODE	f_{CLK} (MHz)	OSR (A × B) [(1)]	DATA RATE (SPS)	–3dB FREQUENCY (Hz)	LATENCY TIME (µs)
3	25.6	32000 (32 × 1000)	400	176.64	2508.60
2	12.8		200	88.32	5017.10
1	3.2		50	22.08	20068.40
0	1.6		25	11.04	40136.90
3	25.6	96000 (32 × 3000)	133.33	58.879	7508.60
2	12.8		66.67	29.441	15017.10
1	3.2		16.67	7.361	60068.40
0	1.6		8.33	3.679	120136.90
3	25.6	160000 (32 × 5000)	80	35.328	12508.60
2	12.8		40	17.664	25017.10
1	3.2		10	4.416	100068.40
0	1.6		5	2.208	200136.90

[1] A = OSR of the sinc4 first stage, B = OSR of the sinc1 second stage.

Figure 7-14 illustrates the frequency response of the sinc4 + sinc1 cascade filter for OSR = 26667 and 32000, representing f_{DATA} = 50 SPS and 60 SPS in speed mode 1 operation. Nulls in the frequency response occur at n × f_{DATA}, n = 1, 2, 3, and so on. At the null frequencies, the filter has zero gain. Assuming no ADC clock frequency error, the normal-mode rejection is 34dB (typical) over a ±2% signal frequency variation at the null frequencies.

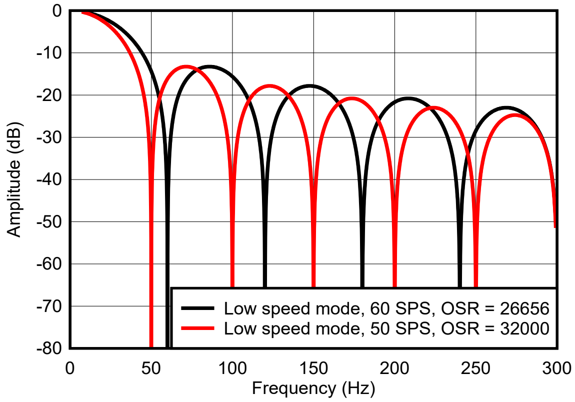


Figure 7-14. Sinc4 + Sinc1 Cascaded Filter Frequency Response

Table 7-13 summarizes the 50Hz and 60Hz line-cycle rejection based on 2% (1Hz in 50Hz case) and 6% ratio tolerance of power-line to ADC clock frequency.

Table 7-13. 50Hz and 60Hz Line Cycle Rejection for Cascade Filter

SPEED MODE [(1)]	OSR	Filter type	f_{DATA} (SPS)	DIGITAL FILTER RESPONSE (dB)			
				50Hz ±1Hz		60Hz ±1Hz	
				CLOCK TOLERANCE: [(2)]			
				0%	1%	0%	1%
0	160000 (32 × 5000)	Sinc4	5	-34.4	-31.5	-36.0	-32.8
1	160000 (32 × 5000)	Sinc4	10	-33.9	-30.5	-35.6	-31.6
1	96000 (32 × 3000)	Sinc4	16.$\overline{6}$	-33.9	-30.3	-21.0	-20.8

Table 7-13. 50Hz and 60Hz Line Cycle Rejection for Cascade Filter (continued)

SPEED MODE [1]	OSR	Filter type	f_{DATA} (SPS)	DIGITAL FILTER RESPONSE (dB)			
				50Hz ±1Hz		60Hz ±1Hz	
				CLOCK TOLERANCE: [2]			
				0%	1%	0%	1%
0	32000 (32 × 1000)	Sinc4	25	-33.8	-30.2	-17.8	-17.8
1	32000 (32 × 1000)	Sinc4	50	-33.8	-30.2	-15.6	-15.3
1	26656 (32 × 833)	Sinc4	60	-15.0	-14.7	-35.2	-31.2
0	16000 (32 × 500)	Sinc4	50	-33.8	-30.2	-15.6	-15.3

[1] Nominal clock frequency for each speed mode is used: f_{CLK} = 25.6MHz (speed mode 3), 12.8MHz (speed mode 2), 3.2MHz (speed mode 1), 1.6MHz (speed mode 0).
[2] 0% clock tolerance corresponds to external clock, 1% clock tolerance corresponds to internal clock.

7.3.8.4 50/60Hz Notch Filters

The digital filter offers a 20SPS and a 25SPS filter option using a custom-coefficient (non-sinc) FIR filter to reject both 50 and 60Hz simultaneously. The 20SPS filter provides better line cycle rejection, the 25SPS filter provides better latency. Both filters maintain a fixed output data rate independent of the selected speed mode. The OSR of the sinc^1 filter preceding the custom FIR notch filter varies with the speed mode to always reduce the input data rate to 800SPS. To achieve the stated normal-mode rejection when using an external clock, select a clock frequency equal to the nominal clock frequency for the respective speed mode.

Table 7-14 and Table 7-15 show the latency and normal-mode rejection for both 20SPS and a 25SPS filter options with and without clock tolerance included. The normal-mode rejection is the same for all speed modes but there is a difference in latency due to differences in settling of the preceding sinc filters versus speed mode.

Table 7-14. 20SPS Filter Latency and 50Hz and 60Hz Line Cycle Rejection

SPEED MODE	f_{CLK} (MHz)	Latency (ms)	DIGITAL FILTER Response (dB)			
			50Hz ±1Hz		60Hz ±1Hz	
			CLOCK TOLERANCE:			
			0%	1%	0%	1%
3	1.6	51.40	–95.3	–82.7	–102.3	–86.1
2	3.2	51.33				
1	12.8	51.27				
0	25.6	51.26				

Table 7-15. 25SPS Filter Latency and 50Hz and 60Hz Line Cycle Rejection

SPEED MODE	f_{CLK} (MHz)	Latency (ms)	DIGITAL FILTER Response (dB)			
			50Hz ±1Hz		60Hz ±1Hz	
			CLOCK TOLERANCE:			
			0%	1%	0%	1%
3	1.6	41.40	–62.7	–57.9	–63.0	–57.9
2	3.2	41.33				
1	12.8	41.27				
0	25.6	41.26				

Figure 7-15 and Figure 7-16 show the frequency response of the 20SPS and the 25SPS filters. Nulls in the frequency response occur at both 50Hz and 60Hz for both filter options.

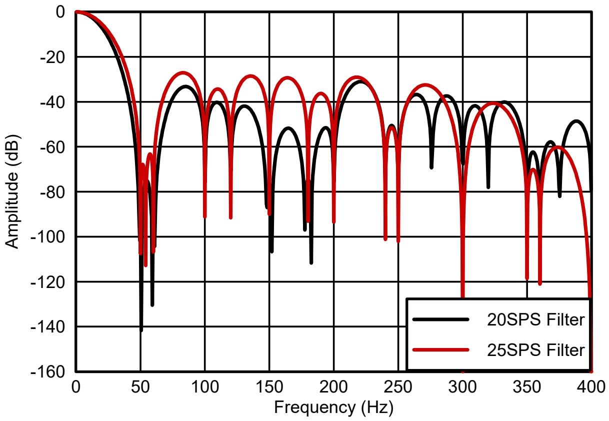


Figure 7-15. 20SPS Filter and 25SPS Filter Frequency Response

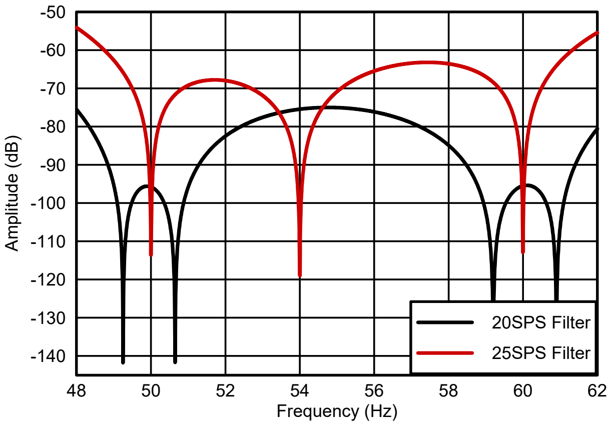


Figure 7-16. 20SPS Filter and 25SPS Filter Frequency Response, 48Hz to 62Hz

7.3.9 FIFO Buffer

The ADS125P08 offers a flexible FIFO (first-in, first-out) buffer to store ADC conversion results and status information until the host controller is ready to read data from the device. The benefit of the FIFO buffer (or "FIFO") is to relax the burden on the host controller when retrieving data from the ADC, especially when the ADC is sampling with a very fast conversion rate.

The FIFO is based on a circular buffer topology with a size of 512x 42 bits, therefore up to 512 data words with 42 bits each can be stored. The 42 bit FIFO word includes conversion data and corresponding status information, as shown in Table 7-16: One FIFO word holds 10 bits of status information, 24 bits of conversion data and 8 bits of CRC. The status bits represent the device status at the completion of each conversion result.

The CRC is calculated as the FIFO is written and then checked on a per-word bases as the FIFO is read. The CRC is based on the CRC-8-ATM (HEC) polynomial $X^8 + X^2 + X^1 + 1$ and initialized to all 1s, see the *SPI CRC* section for details.

Table 7-16. FIFO Buffer 42-Bit Word Content

BIT FIELD	FUNCTION	VALUE
41:37	Status	Status: STEP_INDICTATOR[4:0]
36:33	Status	CONV_COUNT[3:0]
32	Status	ADC_REF_FAULTn
31:8	Data	Conversion Data[23:0]
7:0	CRC	CRC[7:0]

The FIFO is implemented as a circular memory with two internal pointers, a 9 bit read pointer and a 9 bit write pointer, controlling the FIFO operation. See the *FIFO Buffer Read and Write* section for a description of the circular FIFO buffer architecture including read pointer and write pointer operation. The read and write pointers are not accessible to the user, however there are several indicators to monitor the FIFO operation such as FIFO depth, overflow flag and underflow flag, and CRC fault flag. The depth of the FIFO is defined as the difference between the write pointer and read pointer and is user accessible as FIFO_DEPTH[8:0] bits. The depth of the FIFO represents the amount of data stored in the FIFO and available for read. See the *FIFO Depth Indicator* section and the *FIFO Overflow and Underflow* section for details on the FIFO indicators.

Table 7-17 provides an overview of the FIFO buffer architecture.

Table 7-17. FIFO Buffer Architecture

SPECIFICATION	VALUE	DESCRIPTION
Architecture	Circular memory buffer	See *FIFO Buffer Read and Write* section.
FIFO depth	512	Total FIFO address space available.
Width of FIFO word	42	16 bit status + 24 bit data + 8 bit CRC.
Write and read pointer	9 bits each	Write pointer and read pointer are internal pointers: the pointers are not user accessible. See the *FIFO Buffer Read and Write* section.

Enable the FIFO by setting the FIFO_EN bit, and flush the FIFO by resetting this bit to 0b. See the *FIFO Enable and Flush* section for details on FIFO enable and flush.

Read data from the FIFO using the Read FIFO Buffer command, see the *Read FIFO Buffer Command* section for details.

Table 7-18 provides an overview of the FIFO buffer operation and features.

Table 7-18. FIFO Buffer Overview

BIT FIELD	SHORT DESCRIPTION	DESCRIPTION
FIFO_EN	FIFO enable	0b: FIFO disabled and content is flushed; clears all data, resets the read and write pointer. 1b: Conversion data is stored in the FIFO. See the *FIFO Enable and Flush* section.
FIFO_DEPTH[8:0]	FIFO depth indicator	Indicates distance between read pointer and write pointer in FIFO address space: this is the depth of FIFO containing data. See the *FIFO Depth Indicator* section.
FIFO_OFn	FIFO overflow indicator	Indicates write pointer = read pointer – 1 (FIFO full). See the *FIFO Overflow and Underflow* section.
FIFO_UFn	FIFO underflow indicator	Indicates write pointer = read pointer (FIFO empty). See the *FIFO Overflow and Underflow* section.
FIFO_CRC_FAULTn	FIFO data CRC fault indicator	Indicates CRC during FIFO read. CRC is initialized with all 1s. See also the *SPI CRC* section.
FIFO_THRES_A[8:0]	FIFO threshold A configuration	Programmable threshold in FIFO to trigger $\overline{\text{DRDY}}$ transition. See the *FIFO Thresholds* section.
FIFO_THRES_A[8:0]	FIFO threshold B configuration	Programmable threshold in FIFO to trigger $\overline{\text{DRDY}}$ transition. See the *FIFO Thresholds* section.

7.3.9.1 FIFO Buffer Read and Write

The FIFO buffer is implemented as a circular memory with two internal pointers, a read pointer and a write pointer, as shown in Figure 7-17.

In a circular FIFO concept, the memory address of the incoming data is in the write pointer. The address of the first data word in the FIFO that is to be read out is in the read pointer. After reset, both pointers indicate the same memory location. After each write operation, the write pointer is set to the next memory location. The reading of a data word sets the read pointer to the next data word that is to be read out. The read pointer constantly follows the write pointer. When the read pointer reaches the write pointer, the FIFO is empty (underflow). If the write pointer catches up with the read pointer, the FIFO is full (overflow).

Figure 7-17 illustrates the principle of a circular FIFO with two pointers, including an example of reading data and advancing the read pointer as well as writing data and advancing the write pointer. In this example, one data word is read (ADC sample #1) which advances the read pointer to the address of sample #2. Afterwards, one new data word is written (ADC sample #502) which advances the write pointer to the next available address.

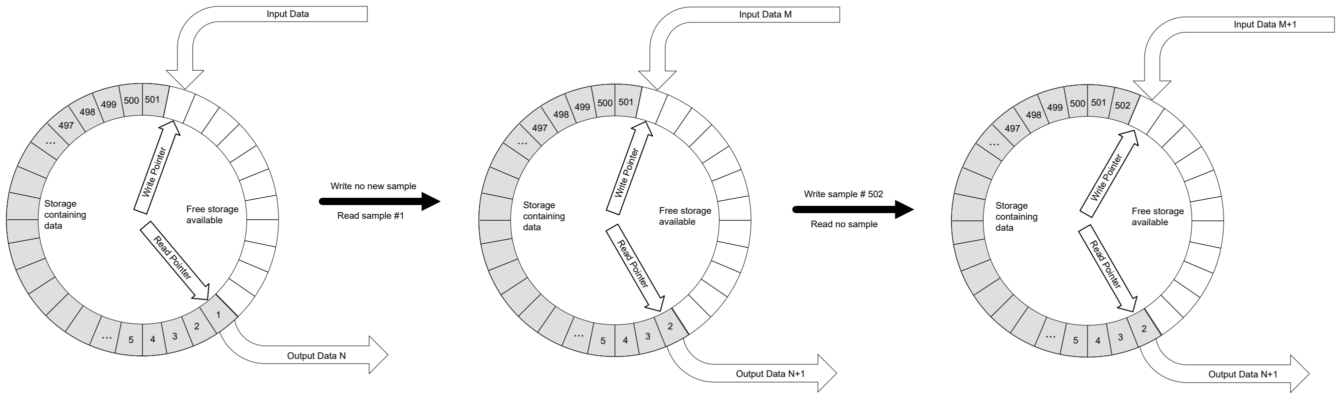


Figure 7-17. FIFO Buffer Read and Write

7.3.9.2 FIFO Overflow and Underflow

Overflow of the FIFO buffer occurs when the write pointer reaches the read pointer, and the FIFO is full. As soon as the write pointer reaches the read pointer, the FIFO_OFn flag in the FIFO_SEQ_STATUS register is set to 0b, indicating that overflow occurred. Subsequently, any additional write operations are ignored by the FIFO, thus additional conversion results are not stored in the FIFO until another read operation frees up memory space. This can lead to discontinuities in the signals acquired. In the overflow condition, the value of the FIFO_DEPTH indicator exceeds the size of the FIFO buffer. Write 1b to clear the FIFO_OFn bit to 1b.

Figure 7-18 illustrates an example for an overflow of the FIFO buffer. Initially four free data words are available in the FIFO in this example. A FIFO read operation results in five free data words available. Subsequently, the ADC captures six conversions, but the FIFO only stores the first five conversion results due to overflow. As soon as the fifth conversion result is stored in the FIFO, overflow occurs and the sixth conversion result is lost.

Typically overflow occurs when, on average, data is written faster to the FIFO than retrieved from the FIFO.

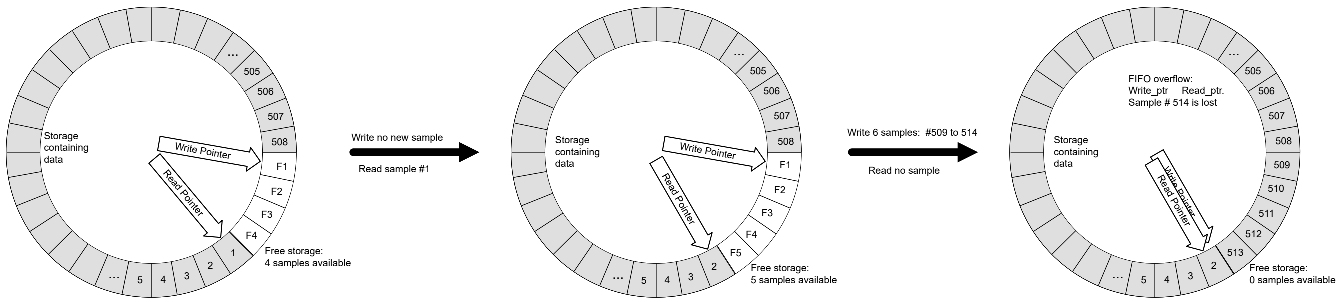


Figure 7-18. FIFO Buffer Overflow Example

Underflow of the FIFO buffer occurs when the read pointer reaches the write pointer, and the FIFO is empty. As soon as the read pointer reaches the write pointer, the FIFO_UFn flag in the FIFO_SEQ_STATUS register is set to 0b, indicating that underflow occurred. Subsequently, any additional read operations result in the same data output. The same conversion result is read multiple times, until new data is written to the FIFO. In an underflow condition, the read pointer does not increment to the next address when a new read command is issued. In the underflow condition, the value of the FIFO_DEPTH indicator is 000000000b. Write 1b to clear the FIFO_UFn bit to 1b.

Figure 7-19 illustrates an example for an underflow of the FIFO buffer. Initially, five data words are stored in the FIFO and available for read (ADC samples #1 to #5). A FIFO read operation results in four data words available for read (ADC samples # to #5). Subsequently, five read commands are executed, however after reading the last available data word (ADC sample #5), underflow occurs and the last data word is read twice as the FIFO is empty.

Typically underflow occurs when, on average, data is written slower to the FIFO than retrieved from the FIFO.

However note that for the ADS125P08, whenever samples have been stored in the FIFO after a FIFO reset (for example by setting and resetting the FIFO_EN bit), the FIFO_UFn does not indicate an underflow condition correctly. Only when no samples have been stored in the FIFO buffer after the reset, and the FIFO read operation is attempted, only then the FIFO_UFn triggers correctly to report the underflow condition.

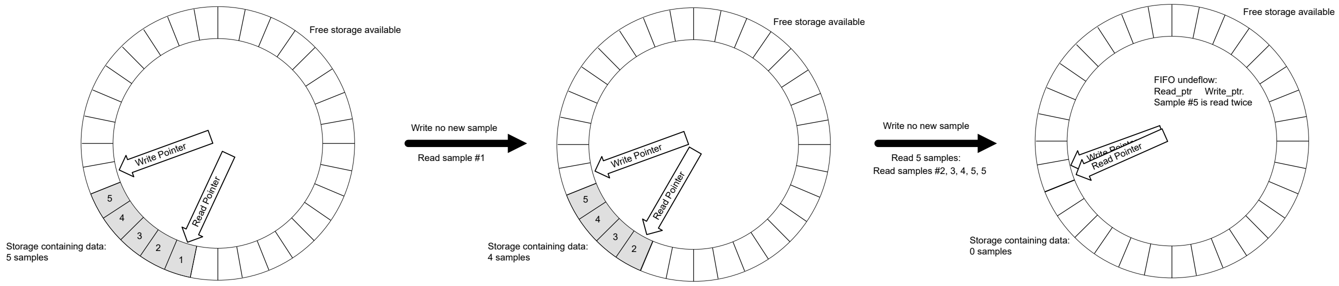


Figure 7-19. FIFO Buffer Underflow Example

7.3.9.3 FIFO Depth Indicator

The FIFO_DEPTH[8:0] indicator is 9 bits wide (512 values), corresponding to the total size of the FIFO buffer (512 words). The FIFO_DEPTH[8:0] bits reside on the Status and General Configuration page in the FIFO_DEPTH_MSB and FIFO_DEPTH_LSB registers. The depth of the FIFO is defined as the difference between the write pointer and read pointer. The depth of the FIFO represents the amount of data stored in the FIFO and available for read.

7.3.9.4 FIFO Enable and Flush

Set the FIFO_EN bit in the FIFO_CFG register to enable the FIFO buffer. As long as FIFO_EN is 0b, the FIFO buffer is not storing conversion results. In this case, only the last completed conversion result is available for read using the conversion data read operation as described in the *Read Conversion Data* section. Alternatively, use the continuous read mode as described in the *Continuous Read Mode* section.

When resetting the FIFO_EN bit to 0b, the FIFO is disabled and flushed. This clears all FIFO data, and resets the read pointer and write pointer.

7.3.9.5 FIFO Thresholds

Two FIFO threshold levels, FIFO_THRES_A[8:0] and FIFO_THRES_B[8:0], are available to monitor FIFO operation. The $\overline{\text{DRDY}}$ pin can be programmed to detect whenever the depth of the FIFO (indicated by the FIFO_DEPTH[8:0] bits) reaches any of the two thresholds. Set the DRDY_CFG[1:0] bits to 11b to enter this mode.

Note that when writing new values to either FIFO_THRES_A[8:0] and FIFO_THRES_B[8:0], the FIFO does not reset or flush. It is recommended that the user stops the sequencer, disables the FIFO, changes the thresholds, enables the FIFO and then starts the sequencer again.

See the *Auto-Sequencer and $\overline{\text{DRDY}}$ Behavior* section for details on the FIFO thresholds and the DRDY_CFG [1:0] = 11b mode.

7.3.10 Channel Auto-Sequencer

The ADS125P08 controls ADC conversion by means of a highly flexible channel auto-sequencer (or "sequencer"). The ADS125P08 sequencer offers up to 32 individual sequence steps. Sequence steps are operating modes which are executed sequentially: Each sequence step (or "step") represents a finite number of ADC conversions (programmable 1 up to 512 per step) and an independent configuration of the ADC for this step, including input channel selection, gain/offset, digital filter settings and timing parameters.

The sequencer stores multiple independent configurations of the ADC in the device memory up front (immediately after power-up), eliminating the need for SPI communication to re-configure the device during the subsequent device operation. This is beneficial in applications where data from very different signal sources is acquired sequentially in a short timeframe. Many of those applications require to switch between multiple ADC configurations for time-interleaving of sensor measurements (high precision, low speed) with diagnostic measurements (low precision, high speed).

The sequencer operation and configuration is controlled by multiple register pages in the user register space: A separate register page exists for each step configuration (pages 1 to 32), referred to as the "Step Configuration Pages". The step registers are identified by the prefix "STEP_x", with x = 0 to 31 indicating the sequence step number, see the *Register Map* section. Page 0 is referred to as the "Status and General Configuration Page" (or just "General Configuration Page") and holds status and generic configuration data.

The 32 sequence steps are individually enabled or disabled using the SEQ_STEP_x_EN bits (x = 0 to 31) on the General Configuration Page. Figure 7-20 shows the register page structure and illustrates the relationship between step enable bits and corresponding Step Configuration Pages.

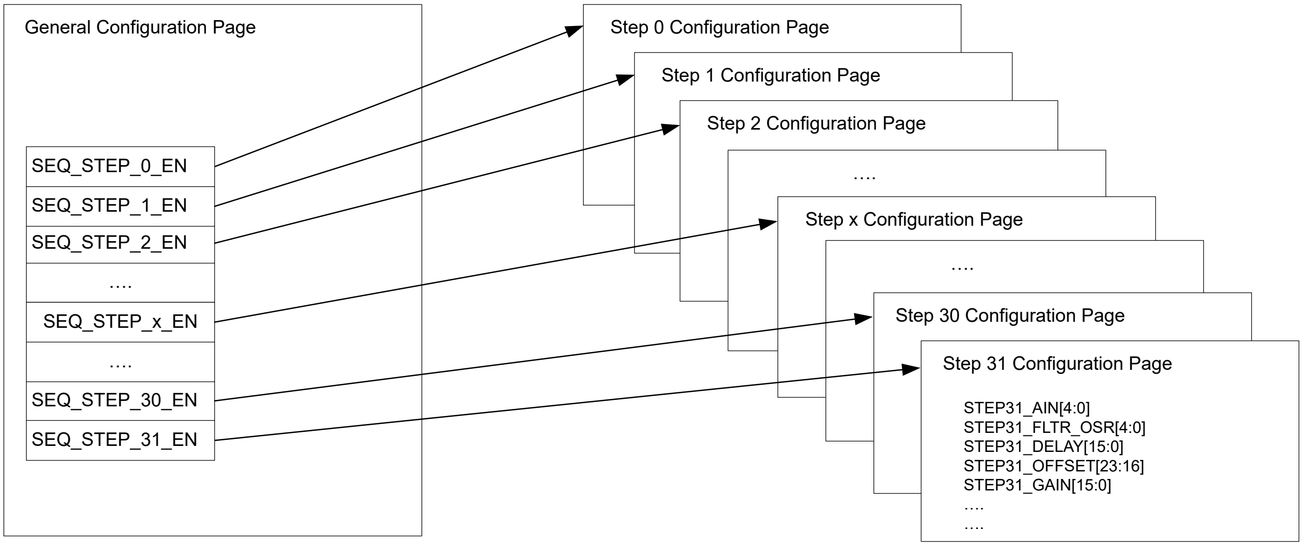


Figure 7-20. Register Page Structure

The General Configuration Page, as well as the individual Step Configuration Pages are addressed using the PAGE_POINTER[7:0] register, which is available on all pages at the same absolute address. See the *Configuring the Auto-Sequencer section* for details on page addressing, and how to configure the auto-sequencer.

The CONVERSION_CTRL, SEQUENCER_CFG and SEQUENCE_STEP_EN_n (n = 0 to 4) registers in the Status and General Configuration Page control the flow of the sequencer. See the *Auto-Sequencer: Basic Operation* section, the *Sequencer Modes* section and the *Starting and Stopping the Sequencer* section for details on how to control the sequencer flow.

The sequencer operation can be monitored by detecting transitions of the $\overline{\text{DRDY}}$ pin. See the *Auto-Sequencer and $\overline{\text{DRDY}}$ Behavior* section for details.

Table 7-19 provides an overview of the sequencer architecture.

Table 7-19. Auto-Sequencer Architecture

SPECIFICATION	VALUE	DESCRIPTION
Architecture	Register page based sequencer	1 Status and General Configuration Page (holds enable bits for all sequence steps). 1 Step Configuration Page per sequence step.
Maximum number of sequence steps	32	Up to 32 individual pages available to define sequence step configurations.
Maximum number of conversions per step	512	Up to 512 ADC conversions for each sequence step. Individually programmable for each step.

7.3.10.1 Auto-Sequencer: Basic Operation

Figure 7-21 shows a flow chart of the sequencer operation. Set the START bit in the CONVERSION_CTRL register to start the sequencer, and use the STOP bit to stop the sequencer. Alternatively, the START pin can be used to control sequencer start and stop. See the *Starting and Stopping the Sequencer* section for details on starting and stopping the sequencer.

Each sequence step corresponds to a finite number of ADC conversions (programmable 1 up to 512), which means up to 16,384 different measurements can be taken in one sequence run. Set the SEQ_MODE[1:0] bits to 10b or 11b to enable sequencer operation. See the *Sequencer Modes* section for a detailed description of the SEQ_MODE[1:0] bits. When a sequence is started, the sequencer steps through all enabled sequence steps, always starting with the step defined by the STEP_INIT[4:0] bits of the CONVERSION_CTRL register. The sequencer ignores sequence steps that are disabled. In each step, the sequencer configures the ADC as defined in the corresponding Step Configuration Page, and then adds a programmable delay before starting the conversions to allow for settling of the input signal. The STEPx_DELAY[15:0] bits on each step configuration page select the delay time for the respective sequencer step x. After the delay, the ADC acquires the pre-defined number of conversions (ranging from 1 up to 512 as defined by the STEPx_NUM_CONV[3:0] bits) before the sequencer configures the ADC for the next step in the sequence.

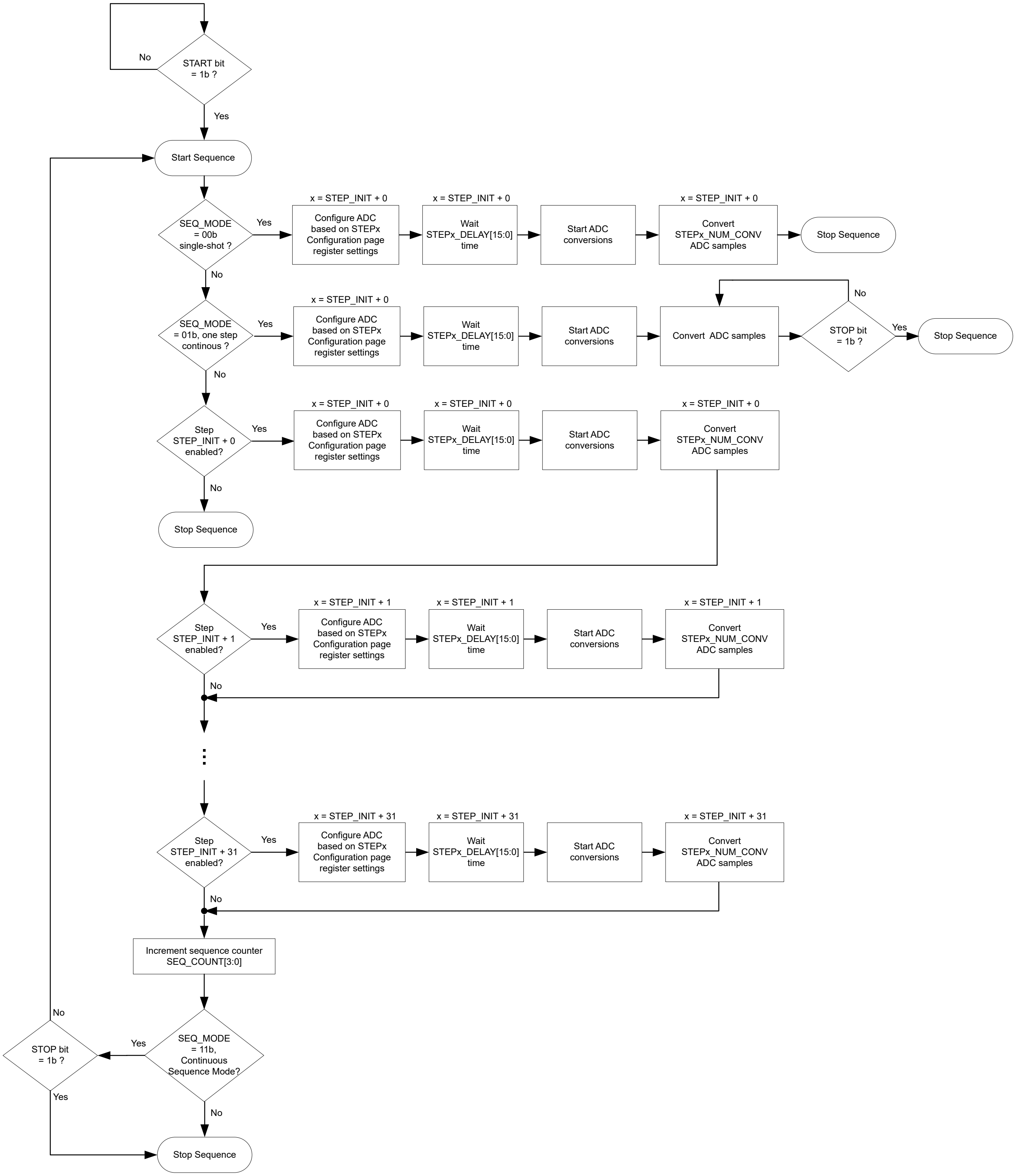

START bit = 1b ?
No
Yes
Start Sequence
SEQ_MODE = 00b single-shot ?
Yes
x = STEP_INIT + 0
Configure ADC based on STEPx Configuration page register settings
x = STEP_INIT + 0
Wait STEPx_DELAY[15:0] time
Start ADC conversions
x = STEP_INIT + 0
Convert STEPx_NUM_CONV ADC samples
Stop Sequence
No
SEQ_MODE = 01b, one step continous ?
Yes
x = STEP_INIT + 0
Configure ADC based on STEPx Configuration page register settings
Wait STEPx_DELAY[15:0] time
Start ADC conversions
Convert ADC samples
STOP bit = 1b ?
No
Yes
Stop Sequence
No
Step STEP_INIT + 0 enabled?
Yes
x = STEP_INIT + 0
Configure ADC based on STEPx Configuration page register settings
x = STEP_INIT + 0
Wait STEPx_DELAY[15:0] time
Start ADC conversions
x = STEP_INIT + 0
Convert STEPx_NUM_CONV ADC samples
No
Stop Sequence
Step STEP_INIT + 1 enabled?
Yes
x = STEP_INIT + 1
Configure ADC based on STEPx Configuration page register settings
x = STEP_INIT + 1
Wait STEPx_DELAY[15:0] time
Start ADC conversions
x = STEP_INIT + 1
Convert STEPx_NUM_CONV ADC samples
No
Step STEP_INIT + 31 enabled?
Yes
x = STEP_INIT + 31
Configure ADC based on STEPx Configuration page register settings
x = STEP_INIT + 31
Wait STEPx_DELAY[15:0] time
Start ADC conversions
x = STEP_INIT + 31
Convert STEPx_NUM_CONV ADC samples
No
Increment sequence counter SEQ_COUNT[3:0]
SEQ_MODE = 11b, Continuous Sequence Mode?
Yes
STOP bit = 1b ?
No
Yes
No
Stop Sequence

Figure 7-21. Sequencer Flow Chart

Several status indicators are available to monitor the progression of the sequencer. All status indicators reside on the General Configuration Page. While a sequence is ongoing, the SEQ_ACTIVE bit is set in the FIFO_SEQ_STATUS register. The STEP_INDICATOR[4:0] bits indicate the sequence step number for the latest completed conversion result, and the CONV_COUNT[3:0] bits indicate the conversion number for the latest completed conversion result. STEP_INDICATOR[4:0] bits and CONV_COUNT[3:0] bits are output in every SPI frame as part of the STATUS header, assuming the status header is enabled, or can be read from the STATUS_MSB and STATUS_LSB registers. The SEQ_COUNT[3:0] bits in the FIFO_SEQ_STATUS register indicate the number of completed sequences; the bits indicate the sequence number for the latest completed conversion result and update when the last conversion of last sequence step completed.

Do not change any register settings on the individual step configuration pages while the sequencer is active. Any write operation to any of the Step Configuration Pages triggers a sequencer reset, see the *Starting and Stopping the Sequencer* section for details.

Set the FIFO_EN bit in the FIFO_CFG register to 1b to enable the FIFO buffer. If the FIFO buffer is enabled, each ADC conversion result is stored in the FIFO immediately after the conversion completes. See the *FIFO Buffer* section for details on the FIFO operation. Use the Read FIFO Buffer Command to read conversion data from the FIFO. If the FIFO buffer is disabled, only the last completed conversion result can be read using the Read Conversion Data operation.

The ADS125P08 sequencer offers four sequence modes. See the *Sequencer Modes* section for details.

Table 7-20 provides a high-level overview of the auto-sequencer functionality.

Table 7-20. Auto-Sequencer Overview

BIT FIELD	SHORT DESCRIPTION	DESCRIPTION
CONFIGURATION AND CONTROL		
PAGE_POINTER[7:0] **PAGE_INDICATOR[7:0]**	Page Pointer, Page Indicator	Pointer and indicator for page addressing, see *Configuring the Auto-Sequencer section*.
SEQ_MODE[1:0]	Sequencer mode selection	Enables sequencer, and selects sequencer mode, continuous versus single sequence, see *Sequencer Modes* section.
SEQ_STEP_x_EN	Sequence step enable	32 sequence steps, x = 1 to 32, one enable bit for each step.
STEP_INIT[4:0]	Pointer to first sequence step	Selects the step number for starting the sequence.
START **STOP**	Start/stop control	Triggers start/stop of sequence, see *Starting and Stopping the Sequencer section*.
STEPx_NUM_CONV[3:0]	Conversion number selection per step	Up to 512 ADC conversions for each sequence step. Individually programmable for each step.
STOP_BEHAVIOR[1:0]	Stop mode selection	Selects operation of sequencer after STOP bit is set.
DRDY_CFG[1:0]	Selection of $\overline{\text{DRDY}}$ behavior	Selects whether DRDY pin indicates new data after every conversion/every sequence step/every sequence/reaching FIFO threshold, see *Auto-Sequencer and $\overline{\text{DRDY}}$ Behavior section.*
INDICATORS		
SEQ_ACTIVE	Sequencer active indicator	Indicates if sequencer is active.
SEQ_COUNT[3:0]	Sequence counter	Counter for completing full sequences.
STEP_INDICATOR[4:0]	Sequence step number indicator	Indicates the sequence step number for the latest completed conversion result.
CONV_COUNT[3:0]	Conversion number indicator	Indicates he conversion number for the latest completed conversion result.

7.3.10.2 Sequencer Modes

The ADS125P08 sequencer offers four sequence modes:

- Single-shot mode, no sequencing
- Single step continuous conversion mode, no sequencing
- Single sequence mode
- Continuous sequence mode

The SEQ_MODE[1:0] bits in the SEQUENCER_CFG register select the sequence mode for the ADS125P08. Table 7-21 and Table 7-22 provide an overview of the sequence modes based on the SEQ_MODE[1:0] bit settings. As shown in Table 7-21, the SEQ_MODE[1] bit enables or disables the sequencer. As shown in Table 7-22, the SEQ_MODE[0] bit further controls the flow mode of the sequencer, defining single versus continuous operation.

Table 7-21. Sequencer Enable

SEQ_MODE[1]	DESCRIPTION
0	Sequencer disabled: Step page enable bits are ignored.
1	Sequencer enabled: Enabled steps are executed.

Table 7-22. Sequencer Modes

SEQ_MODE[1:0]	DESCRIPTION
00	Single-shot mode, no sequencing. Step defined by STEP_INIT[4:0] bits is executed only once.
01	Single step continuous conversion mode, no sequencing. Step defined by STEP_INIT[4:0] is executed indefinitely.
10	Single sequence mode: Each enabled sequence step is executed once.
11	Continuous sequence mode: The sequence is repeated indefinitely.

7.3.10.2.1 Single-Shot Mode

Setting the SEQ_MODE[1:0] bits in the SEQUENCER_CFG register to 00b selects the single-shot mode (no sequencing). In this mode, the device only executes one sequence step, one time. The SEQ_STEP_x_EN bits are ignored in single-shot mode. The STEP_INIT[4:0] bits in the CONVERSION_CTRL register point to the sequence step x (x = 0 to 31) to be executed:

$$x = \text{STEP\_INIT}[4:0] \qquad (18)$$

Figure 7-21 shows a flow chart of the sequencer operation, including the behavior in single-shot mode (SEQ_MODE[1:0] = 00b).

The device configures the ADC based on the step configuration page x, adds the programmable delay time defined by the STEPx_DELAY[15:0] bits and then starts ADC conversions. The number of ADC conversions (up to 512) is set by the STEPx_NUM_CONV[3:0] bits in the STEPx_ADC_REF_CFG register.

After completion of the ADC conversions (1 up to 512) in single-shot mode, the device enters standby mode (STBY_MODE = 1b) or remains fully powered in idle mode (STBY_MODE = 0b).

7.3.10.2.2 Single Step Continuous Conversion Mode

Setting the SEQ_MODE[1:0] bits in the SEQUENCER_CFG register to 01b selects the single step continuous conversion mode (no sequencing). In this mode, the device only executes one sequence step, over and over again. The SEQ_STEP_x_EN bits are ignored in this mode. The STEP_INIT[4:0] bits in the CONVERSION_CTRL register point to the sequence step x (x = 0 to 31) to be executed :

$$x = \text{STEP\_INIT}[4:0] \qquad (19)$$

Figure 7-21 shows a flow chart of the sequencer operation, including the behavior in single step continuous conversion mode (SEQ_MODE[1:0] = 01b).

The device configures the ADC based on the step configuration page x, adds the programmable delay time defined by the STEPx_DELAY[15:0] bits and starts ADC conversions. The STEPx_NUM_CONV[3:0] bits in the STEPx_ADC_REF_CFG register are ignored in this operating mode. The ADC continuous conversions until the device is powered down, or the sequencer is stopped by any of the options described in the *Starting and Stopping the Sequencer* section.

In this mode, one option to discontinue the ADC conversions is setting the STOP bit to 1b. The ADC conversion discontinue immediately or continue until the sequence step is completed depending on the STOP_BEHAVIOR[1:0] bits in the SEQUENCER_CFG register. See the *Starting and Stopping the Sequencer* section for details. The device then enters standby mode (STBY_MODE = 1b) or remains fully powered in idle mode (STBY_MODE = 0b).

7.3.10.2.3 Single Sequence Mode

Setting the SEQ_MODE[1:0] bits in the SEQUENCER_CFG register to 10b selects the single sequence mode (sequencer enabled). In this mode, the device executes each enabled sequence step one time. The STEP_INIT[4:0] bits in the CONVERSION_CTRL register point to the first sequence step x_initial (x = 0 to 31) to be executed:

$$\text{x\_inital} = \text{STEP\_INIT[4:0]} \tag{20}$$

Figure 7-21 shows a flow chart of the sequencer operation, including the behavior in single sequence mode (SEQ_MODE[1:0] = 10b).

Enable sequence steps by setting the corresponding SEQ_STEP_x_EN bit (x = 0 to 31) in the SEQUENCE_STEP_EN registers to 1b. If the first sequence step as defined by STEP_INIT[4:0] is not enabled, sequencing is not started. If the first enabled sequence step x_inital is not equal to 0, then only steps from x_inital to 31 are executed in this mode, and steps smaller than x_inital are ignored even if enabled. For example, if x_inital = 20, and all steps are enabled, only steps 20 to 31 are executed.

For each enabled sequence step, the device configures the ADC based on the step configuration page x, adds the programmable delay time defined by the STEPx_DELAY[15:0] bits and starts ADC conversions. The number of ADC conversions for the step is set by the STEPx_NUM_CONV[3:0] bits in the STEPx_ADC_REF_CFG register. Up to 512 ADC conversion results can be sampled. Afterwards the device proceeds to the next enabled sequence step.

To stop the sequencer and discontinue the ADC conversions, set the STOP bit to 1b. See the *Starting and Stopping the Sequencer* section for details on stopping the sequencer, and details on the stop behavior defined by the STOP_BEHAVIOR[1:0] bits.

After the last conversion of the last sequence step completed, the device enters standby mode (STBY_MODE = 1b) or remains fully powered in idle mode (STBY_MODE = 0b).

7.3.10.2.4 Continuous Sequence Mode

Setting the SEQ_MODE[1:0] bits in the SEQUENCER_CFG register to 11b selects the continuous sequence mode (sequencer enabled). In this mode, the device runs through the configured sequence over and over again until stopped by the host.

The STEP_INIT[4:0] bits in the CONVERSION_CTRL register point to the first sequence step x_initial (x_initial = 0 to 31) to be executed:

$$\text{x\_inital} = \text{STEP\_INIT[4:0]} \tag{21}$$

Figure 7-21 shows a flow chart of the sequencer operation, including the behavior in continuous sequence mode (SEQ_MODE[1:0] = 11b).

Enable sequence steps by setting the corresponding SEQ_STEP_x_EN bit (x = 0 to 31) in the SEQUENCE_STEP_EN registers to 1b. If the first sequence step as defined by STEP_INIT[4:0] is not enabled (i.e. SEQ_STEP_x_initial_EN = 0b), sequencing is not started.

For each enabled sequence step, the device configures the ADC based on the step configuration page x, adds the programmable delay time defined by the STEPx_DELAY[15:0] bits and starts ADC conversions. The number of ADC conversions is set by the STEPx_NUM_CONV[3:0] bits in the STEPx_ADC_REF_CFG register. Up to 512 ADC conversion results can be sampled. Afterwards the device proceeds to the next enabled sequence step.

To stop the sequencer and discontinue the ADC conversions, set the STOP bit to 1b. See the *Starting and Stopping the Sequencer*section for details on stopping the sequencer, and details on the stop behavior defined by the STOP_BEHAVIOR[1:0] bits.

After the last conversion of the last sequence step completed, the device enters standby mode (STBY_MODE = 1b) or remains fully powered in idle mode (STBY_MODE = 0b).

7.3.10.3 Configuring the Auto-Sequencer

As explained in the *Channel Auto-Sequencer* section, the sequencer operation and configuration is controlled by registers in the General Configuration Page and the individual Step Configuration Pages.

Configure the auto-sequencer by writing to the sequencer related registers both in the Step Configuration Pages (pages 1 to 32) as well as the General Configuration Page (page 0). Access the general configuration registers, or the register page for each individual sequence step by writing the corresponding address value (0, or 1 to 32) to the PAGE_POINTER[7:0] register as shown in Table 7-23.

Table 7-23. Register Page Decoding and Sequencer Steps

PAGE_POINTER[7:0]	REGISTER PAGE CONTENT	STEP ENABLE BIT	DESCRIPTION
00h	Status and General Configuration Page	N/A	Register page for generic status and configuration data
01h	Step 0 Configuration Page	SEQ_STEP_0_EN	Configuration settings for sequence step 0
02h	Step 1 Configuration Page	SEQ_STEP_1_EN	Configuration settings for sequence step 1
03h	Step 2 Configuration Page	SEQ_STEP_2_EN	Configuration settings for sequence step 2
...	...	...	...
1Fh	Step 30 Configuration Page	SEQ_STEP_30_EN	Configuration settings for sequence step 30
20h	Step 31 Configuration Page	SEQ_STEP_31_EN	Configuration settings for sequence step 31
21h – FFh	Invalid pages	N/A	Invalid pages

Follow this procedure for a reliable configuration of the auto-sequencer:

1. Stop an ongoing sequence by setting the STOP bit in the CONVERSION_CTRL register to 1b, or alternatively put the device into power down mode.
2. Verify that the SEQ_ACTIVE bit reads 0b, to verify that the sequencer is not running.
3. Write 00h to the PAGE_POINTER[7:0] to access the General Configuration Page.
4. Configure the STEP_INIT[4:0], SEQ_MODE[1:0], STOP_BEHAVIOR[1:0] and DRDY_CFG[1:0] bits on the General Configuration Page. See Table 7-20 for an overview of those bits.
5. Enable the desired sequence steps by setting the corresponding SEQ_STEP_x_EN bits (x = 0 to 31) on the General Configuration Page.
6. Write the address for an enabled sequence step to the PAGE_POINTER[7:0], as defined in Table 7-23, and then configure each register on the selected Step Configuration Page. See Table 7-24 for an overview. Repeat this for all enabled sequence steps.
7. If the device is configured in power down mode, put the device back into active mode.
8. Start the sequencer by setting the START bit.

Each Step Configuration Page contains configuration settings for input multiplexer, voltage reference, ADC, digital filters, gain and offset calibration as well as GPIOs, as shown in Table 7-24.

When the sequencer is disabled (SEQ_MODE[1] = 0b), a write to the step registers updates the device configuration immediately. When the sequencer is enabled (SEQ_MODE[1] = 1b), a write to the step registers takes in effect at the next sequencer start.

Table 7-24. Configuration Settings per Sequence Step

REGISTER	BIT FIELD	DESCRIPTION
STEPx_AINP_CFG STEPx_AINN_CFG	STEPx_AINP[4:0] STEPx_AINN[4:0]	Selection of positive and negative analog input for ADC
STEPx_ADC_REF_CFG	STEPx_EXT_RNG	Selection of range for external reference
STEPx_ADC_REF_CFG	STEPx_REF_SEL	Selection of external vs internal voltage reference
STEPx_ADC_REF_CFG	CODING	Conversion data coding selection, unipolar vs bipolar
STEPx_ADC_REF_CFG	STEPx_NUM_CONV[3:0]	Number of ADC conversions to be executed during this sequence step
STEPx_FILT1_CFG	STEPx_FLTR_OSR[4:0]	OSR for digital filter
STEPx_FILT1_CFG	STEPx_FLTR_MODE	Selection of filter mode sinc3 vs sinc4
STEPx_DELAY_xSB_CFG	STEPx_DELAY_MSB[7:0]	Conversion-start delay time selection
STEPx_OFFSET_CAL_xSB	STEPx_OFFSET_CAL[23:0]	ADC offset calibration value
STEPx_GAIN_CAL_xSB	STEPx_GAIN_CAL[15:0]	ADC gain calibration value

7.3.10.4 Starting and Stopping the Sequencer

There are two ways to start or stop the sequencer operation:

- Writing to the START and/or STOP bit in the CONVERSION_CTRL register
- Use the START pin for start/stop control

Set the START bit in the CONVERSION_CTRL register to start the sequencer. Setting the START bit while the sequencer is running aborts the ongoing sequence run and restarts a new sequence run from the beginning. Setting the START bit while the ADC is in powerdown mode does not start a sequence.

The START bit clears after a sequence begins, and thus is reading back 0b.

Figure 7-22 shows an example of the sequencer operation including the START pin timing.

Use the STOP bit to stop the sequencer. After setting the STOP bit to stop the sequencer, the STOP bit reads back 1b until the sequencer is stopped. Putting the device into power-down mode aborts the sequence run immediately.

The last conversion result of a sequence run is still available for readout after the sequencer stopped. The conversion results of the sequencer are only cleared to 0h after a device reset, when the device is in power-down mode, or are overwritten when conversion results from a new sequence run become available.

After the sequencer stopped, the configuration settings are still active as defined on the last active step page before the stop. This applies to all bits on the step page, including GPIO outputs. The configuration from the last step page is maintained until the device is reset, power-cycled or the configuration is overwritten by a new sequence run.

As an alternative to using the START and STOP bit in the CONVERSION_CTRL register, the START pin can be used to control sequencer starts and stops. A rising edge on the START pin is equivalent to writing to the START bit. A falling edge on the START pin is equivalent to writing to the STOP bit. Configure the GPIO0/START pin to START pin operation using the GPIO0_CFG[1:0] bits (setting 11b). Put the device into the power down mode first before programming GPIO0. The GPIO0_CFG[1:0] bits default to 00b at power-up, thus the START pin operation is only available after setting the GPIO0_CFG[1:0] bits to 11b, which requires SPI communication. Therefore the START pin functionality is not available at power-up. The first step page executed is determined by the STEP_INIT[4:0] bits, just like with the START and STOP bits. If the START pin is high at power-up, a start operation begins after the power-up cycle completes.

The following register write operations aborts the ongoing sequence run and stop the sequencer:

- A write operation to any register on any Step Configuration Page (applies to both enabled and disabled pages)
- A write operation to any of the registers on the General Configuration Page except the following:

  - AGPIOx_CFG[1:0]
  - FIFO_EN
  - FIFO_TRESH_A[8:0]
  - FIFO_THRES_B[8:0]
  - FAULT_PIN_BEHAVIOR
  - REG_MAP_CRC_EN
  - STATUS_EN
  - SPI_CRC_EN
  - PFx_EN
  - PFx_BYPASS
  - CS_FWD_EN_CODE[7:0]
  - AGPIOCx_FWD_EN
  - GPIOx_FWD_EN

The STOP_BEHAVIOR[1:0] bits in the SEQUENCER_CFG register define the operation of the sequencer after a stop is initiated, depending on the sequencer mode, as shown in Table 7-25.

Table 7-25. STOP BEHAVIOR MODES

SEQ_MODE[1:0]	STOP_BEHAVIOR[1:0]			
	00	**01**	**10**	**11**
00	Stop immediately	Stop after current conversion completes	Stop after current step completes	
01		Stop after current conversion completes		
10		Stop after current conversion completes	Stop after current step completes	Stop after current sequence completes
11				

If a change in the sequencer configuration is desired, stop the sequencer and re-configure the settings while the sequencer is not running. To avoid false sequencer starts, follow the procedure given in the *Configuring the Auto-Sequencer* section to configure and re-start the sequencer.

7.3.10.5 Auto-Sequencer and $\overline{\text{DRDY}}$ Behavior

Configure the behavior of the $\overline{\text{DRDY}}$ pin during sequencer operation using the DRDY_CFG[1:0] bits in the SEQUENCER_CFG register.

There are four modes available for the $\overline{\text{DRDY}}$ behavior: Depending on the mode, $\overline{\text{DRDY}}$ drives low

- every time a new conversion result is available: DRDY_CFG[1:0] = 00b
- every time a sequence step is completed: DRDY_CFG[1:0] = 01b
- every time a sequence is completed: DRDY_CFG[1:0] = 10b
- when a pre-defined threshold in the FIFO buffer is reached: DRDY_CFG[1:0] = 11b

Figure 7-22 shows the $\overline{\text{DRDY}}$ operation for driving $\overline{\text{DRDY}}$ low every time a new conversion result is available (DRDY_CFG[1:0] = 00b). Two sequence steps are shown in this example with four ADC conversions executed in the first sequence step, and three ADC conversions in the second sequence step. $\overline{\text{DRDY}}$ is driven low once every individual conversion result is available.

At the beginning of a new sequence step, the first conversion is fully settled data but incurs a delay (latency time) compared to the normal data period t_{DATA} = 1 / f_{DATA}. This latency is needed to account for full settling of the digital filter. The latency time depends on the data rate and the filter mode (see the *Digital Filter* section for filter latency details). The time t_{SETTLE} is the time from the start of the sequence step (last $\overline{\text{DRDY}}$ falling edge of the previous step) to the first $\overline{\text{DRDY}}$ falling edge within the new sequence step. The time t_{SETTLE} also includes the programmable delay defined by the STEPx_DELAY_MSB[7:0] bits and the STEPx_DELAY_LSB[7:0] bits in the Step Configuration Page of this sequence step. As a result, t_{SETTLE} is the sum of programmable delay t_{STEPx_DELAY} and filter latency $t_{STEPx\_FLTR\_LATENCY}$:

$$t_{SETTLEx} = t_{STEPx\_DELAY} + t_{STEPx\_FLTR\_LATENCY} \tag{22}$$

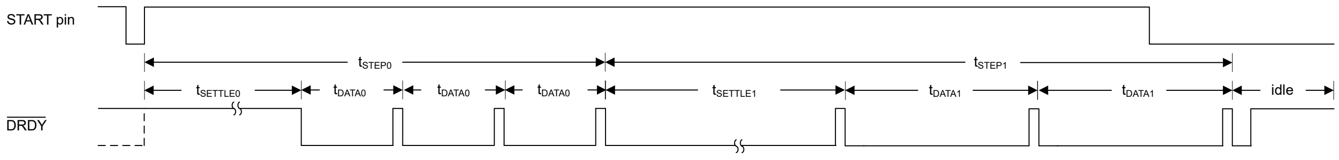


Figure 7-22. $\overline{\text{DRDY}}$ Pulse After Every Conversion

In Figure 7-22, no data are read from the ADC, thus $\overline{\text{DRDY}}$ remains low and pulses high shortly before the next $\overline{\text{DRDY}}$ falling edge. If data are read from the ADC after every conversion result completes, $\overline{\text{DRDY}}$ is forced high at the eighth SCLK edge during conversion data read operation. This is illustrated in Figure 7-23, where new data is read every time shortly after the falling edge of $\overline{\text{DRDY}}$ indicates that new data are available.

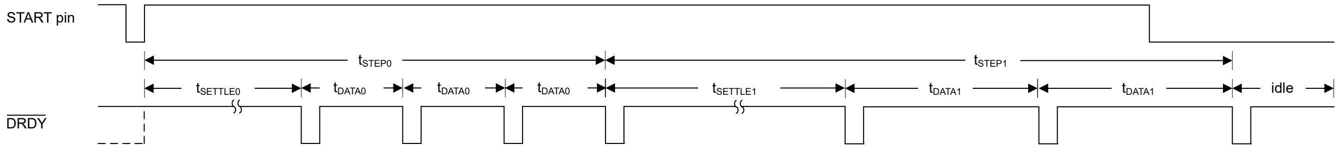


Figure 7-23. $\overline{\text{DRDY}}$ Pulse After Every Conversion, Data is Read From ADC

Figure 7-24 shows the $\overline{\text{DRDY}}$ operation for driving $\overline{\text{DRDY}}$ low every time a sequence step has completed (DRDY_CFG[1:0] = 01b). Two sequence steps are again shown in this example with four ADC conversions executed in the first sequence step, and three ADC conversions in the second sequence step. $\overline{\text{DRDY}}$ is driven low when the last conversion result in each sequence step is available.

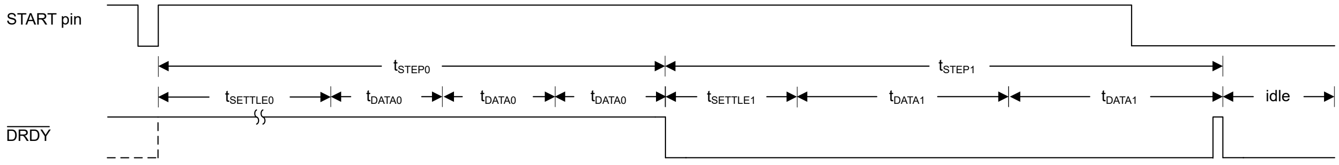


Figure 7-24. $\overline{\text{DRDY}}$ Pulse After Every Sequence Step

Figure 7-25 shows the $\overline{\text{DRDY}}$ operation for driving $\overline{\text{DRDY}}$ low every time a full sequence completed (DRDY_CFG[1:0] = 10b). In this particular example, the complete sequence only comprises of two sequence steps (STEP0 and STEP1). $\overline{\text{DRDY}}$ is driven low when the last conversion result of the final (last) sequence step is available.

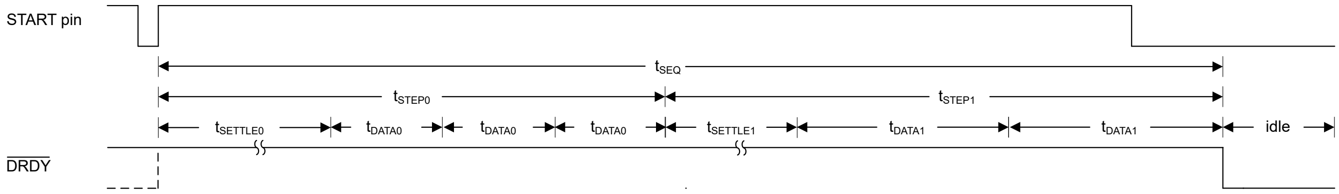


Figure 7-25. $\overline{\text{DRDY}}$ Pulse After Completion of Sequence

Table 7-26 summarizes the $\overline{\text{DRDY}}$ operation based on the FIFO thresholds (DRDY_CFG[1:0] = 11b). In this mode, the depth of the FIFO indicated by the FIFO_DEPTH[8:0] bits is monitored and compared against the two FIFO thresholds controlling the $\overline{\text{DRDY}}$ behavior, FIFO_THRES_A[8:0] and FIFO_THRES_B[8:0]. See the *FIFO Buffer* section for details on the FIFO buffer operation and the FIFO_DEPTH[8:0] bits. When the depth of the FIFO exceeds the larger of either thresholds (FIFO_THRES_A or FIFO_THRES_B), then $\overline{\text{DRDY}}$ transitions from high to low. This condition can be described as FIFO_DEPTH > FIFO_THRES_A (assuming FIFO_THRES_A is larger or equal to FIFO_THRES_B). When the depth of the FIFO is equal or smaller than the lowest of either threshold, then $\overline{\text{DRDY}}$ transitions from low to high. This condition can be described as FIFO_DEPTH < FIFO_THRES_B (assuming FIFO_THRES_B is smaller or equal to FIFO_THRES_A).

Table 7-26. $\overline{\text{DRDY}}$ Behavior Based on FIFO Thresholds (DRDY_CFG[1:0] = 11b)

THRESHOLD SETTING	TRIGGER CONDITION	$\overline{\text{DRDY}}$ TRANSITION
FIFO_THRES_A ≥ FIFO_THRES_B	FIFO_DEPTH > FIFO_THRES_A	$\overline{\text{DRDY}}$ falling edge
	FIFO_DEPTH ≤ FIFO_THRES_B	$\overline{\text{DRDY}}$ rising edge
FIFO_THRES_A < FIFO_THRES_B	FIFO_DEPTH > FIFO_THRES_B	$\overline{\text{DRDY}}$ falling edge
	FIFO_DEPTH ≤ FIFO_THRES_A	$\overline{\text{DRDY}}$ rising edge

In the example shown in Figure 7-26, six samples are acquired by the ADC and then two conversion results are read from the ADC. The FIFO thresholds controlling $\overline{\text{DRDY}}$ behavior have been set to a value of five samples for the FIFO_THRES_A thresholds (FIFO_THRES_A[8:0] = 000000101b) and four samples for the FIFO_THRES_B threshold (FIFO_THRES_B[8:0] = 000000100b). In this case, FIFO_THRES_A ≥ FIFO_THRES_B. Therefore, as soon as six samples have been acquired (starting from an empty FIFO), the condition that FIFO_DEPTH > FIFO_THRES_A is met (6 > 5), and $\overline{\text{DRDY}}$ transitions from high to low. After two conversion results are read from the ADC, and no additional sample is converted in the meantime, the condition that FIFO_DEPTH ≤ FIFO_THRES_B is met (4 ≤ 4), and $\overline{\text{DRDY}}$ transitions from low to high.

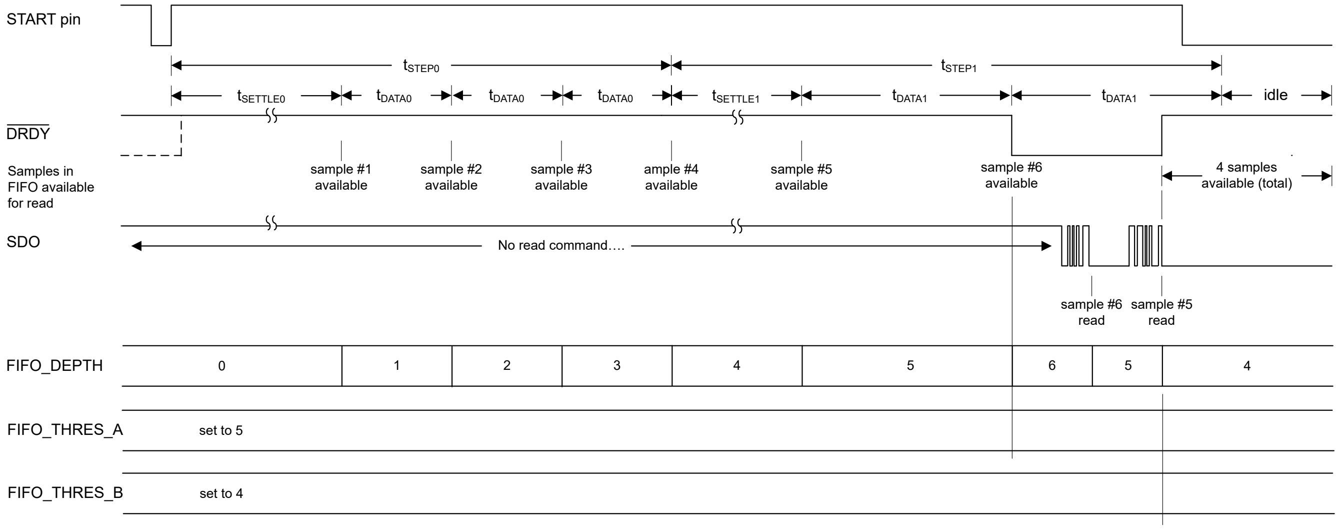


Figure 7-26. $\overline{\text{DRDY}}$ Behavior Based on FIFO Thresholds

7.3.11 Offset and Gain Calibration

The ADS125P08 provides the ability to calibrate offset and gain errors by using user-programmable offset and gain correction registers. As shown in Figure 7-27, the 24 bit offset correction value is subtracted from the conversion data before being multiplied by the 16 bit gain correction value. Output data are rounded to the final resolution (24 bit) and clipped to +FS and –FS code values after the scaling operation. The offset and gain calibration coefficients must be stored in external nonvolatile memory and programmed into the offset and gain calibration registers each time the device powers up or resets because the ADS125P08 registers are volatile.

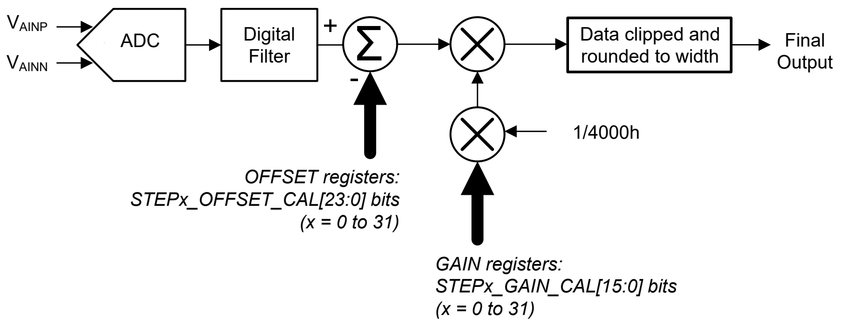

V_{AINP}
V_{AINN}
ADC
Digital Filter
+
-
Σ
Data clipped and rounded to width
Final Output
1/4000h
OFFSET registers: STEPx_OFFSET_CAL[23:0] bits (x = 0 to 31)
GAIN registers: STEPx_GAIN_CAL[15:0] bits (x = 0 to 31)

Figure 7-27. Calibration Logic Block Diagram

Equation 23 shows how conversion data are calibrated:

$$\text{Final Output Data} = (\text{Data} - \text{OFFSET}) \times \text{GAIN} / 4000\text{h} \qquad (23)$$

The 24-bit offset calibration value is provided in two's-complement format and programmed into the STEPx_OFFSET_CAL[23:0] bit field that spans across the STEPx_OFFSET_CAL_MSB, STEPx_OFFSET_CAL_LSB and STEPx_OFFSET_CAL_MSB registers. Table 7-27 shows example offset calibration values. The LSB size of the offset calibration value is calculated using Equation 24.

$$\text{LSB size} = (2 \times V_{REF}) / (2^{24}) \qquad (24)$$

Table 7-27. Offset Calibration Value Examples

STEPx_OFFSET_CAL[23:0] VALUE	APPLIED OFFSET CORRECTION
000010h	–16LSB
000001h	–1LSB
FFFFFFh	1LSB
FFFFF0h	16LSB

The 16-bit gain calibration value is provided in straight binary format normalized to unity gain at 400000h. The gain calibration value is programmed into the STEPx_GAIN_CAL[15:0] bit field. One LSB of the gain calibration value equals a gain correction factor of $1/2^{16}$ = 0.000015. For example, to correct a gain error greater than 1, the calculated gain calibration value is less than 400000h.

Table 7-28 shows example gain calibration values.

Table 7-28. Gain Calibration Value Examples

STEPx_GAIN_CAL[15:0] VALUE	APPLIED GAIN CORRECTION
FFFFh	3.999939
4333h	1.05
4001h	1.000015
4000h	1
3FFFh	0.999985
3CCCh	0.95

The recommended calibration procedure is as follows:

1. Preset the offset and gain calibration registers to STEPx_OFFSET_CAL[23:0] = 000000h and STEPx_GAIN_CAL[15:0] = 4000h, respectively.
2. Perform an offset calibration by shorting the ADC inputs internally using the respective input and system monitor multiplexer setting (STEPx_SYS_MON[3:0] = 0001b), or short the inputs externally at the system level to include the offset error of the external filter stages. Acquire multiple conversion data and write the average value of the data into the offset calibration registers. Averaging the data reduces conversion noise to improve calibration accuracy.
3. Perform a gain calibration by applying a precision calibration signal to the ADC inputs or at the system level to include the gain error of the external filter stages. Select the calibration voltage to be less than the full-scale input range to avoid clipping the output code. Clipped output codes result in inaccurate calibration. For example, use a 3.9V calibration signal with V_{REF} = 4.096V. When operating in the extended range mode, the calibration signal can be equal to V_{REF} without causing clipped output codes. Use Equation 25 to calculate the gain calibration value.

$$\text{Gain Calibration Value} = (\text{expected output code} / \text{actual output code}) \times 4000\text{h} \qquad (25)$$

For example, the expected output code for a 3.9V calibration voltage using a 4.096V reference voltage is: (3.9V / LSB size) = 79E000h, where LSB size = 2 × 4.096V / (2^{24}) = 4.096V / 7FFFFFh. If the actual measured output code is 741249h for example, then the gain calibration factor calculates to 1.05. The resulting gain calibration value to write into the STEPx_GAIN_CAL[15:0] bit field is: (1.05 – 1) / (1 / 2^{16}) = 4333h.

7.3.12 General Purpose IOs (GPIOs)

The ADS125P08 provides four eight analog inputs (AIN8 to AIN15) that can be configured as general purpose inputs and outputs (GPIOs). The logic levels of those GPIOs are referenced to the AVDD supply (therefore denoted AGPIOx). Use the AGPIOx_CFG[1:0] (x = 0 to 7) bits to configure the pins as either analog inputs, digital inputs or digital outputs with either push-pull or open-drain characteristic.

The ADS125P08 provides four digital pins (GPIO0 to GPIO3) that can be configured as general purpose inputs and outputs (GPIOs). The logic levels of those GPIOs are referenced to the IOVDD supply. Use the GPIOn_CFG[1:0] (n = 0 to 3) bits to configure the pins as either analog inputs, digital inputs or digital outputs, or for a special function.

Set the digital output levels of the (A)GPIOs using the STEPx_(A)GPIOn_DAT_OUT bits. The STEPx_(A)GPIOn_DAT_OUT bit setting has no effect when (A)GPIOx is configured as an analog or digital input.

The (A)GPIOx_DAT_IN bits indicate the readback values at the (A)GPIOx pins irrespective if the pins are configured as digital inputs or outputs. The (A)GPIOx_DAT_IN bits read back 0b when (A)GPIOx is configured as an analog input.

In addition, the following special functions are available:

- GPIO0/START can be configured as a START input. Set GPIO0_CFG[1:0] bits to 11b to configure the pin as a START input. Put the device into the power down mode first before programming GPIO0. See the *Starting and Stopping the Sequencer* section for details.
- $\overline{\text{DRDY}}$/GPIO1 can be configured as a dedicated $\overline{\text{DRDY}}$ output.
- GPIO2/CLKIN can be configured as an external clock input. Set GPIO2_CFG[1:0] bits to 11b to configure the pin as a clock input. See the *Clock Operation* section for details.
- GPIO3/$\overline{\text{FAULT}}$ can be configured as a $\overline{\text{FAULT}}$ indication output.

7.3.12.1 $\overline{\text{DRDY}}$ Output

By default, $\overline{\text{DRDY}}$/GPIO1 is configured as a dedicated $\overline{\text{DRDY}}$ output (i.e. GPIO1_CFG[1:0] = 11b). A falling edge on the $\overline{\text{DRDY}}$ pin indicates the completion of new conversion data. The $\overline{\text{DRDY}}$ output is always driven even when $\overline{\text{CS}}$ is high. See the *Data Ready ($\overline{\text{DRDY}}$) Pin* section and the *$\overline{\text{DRDY}}$ Pin Behavior* section for more details.

7.3.12.2 $\overline{\text{FAULT}}$ Output

Configure GPIO3/$\overline{\text{FAULT}}$ as a $\overline{\text{FAULT}}$ output by setting GPIO3_CFG = 11b. The $\overline{\text{FAULT}}$ pin is low when any of the ADC_REF_FAULTn, FIFO_FAULTn, INTERNAL_FAULTn, REG_WRITE_FAULTn or SPI_CRC_FAULTn status bits are 0b to indicate a fault. Connect a pulldown resistor from GPIO3/$\overline{\text{FAULT}}$ to GND to also detect potential device resets because the pin reverts back to a high-Z analog input during and after reset.

Use the FAULT_PIN_BEHAVIOR bit to select from the following $\overline{\text{FAULT}}$ output behaviors:

- Static output. The $\overline{\text{FAULT}}$ output is low when a fault occurred, otherwise the output is high.
- Heart beat output. The $\overline{\text{FAULT}}$ output is low when a fault occurred, otherwise the output is a 50% duty-cycle signal with a frequency of f_{MOD}/256. The heart beat signal frequency can be monitored by the host to detect potential device clock faults.

7.3.13 Burn-Out Current Sources

To help detect potential sensor faults, the ADS125P08 provides a pair of programmable burn-out current sources (BOCS). Use the STEPx_BOCS[1:0] bits on the step configuration page x to enable the current sources and to set the values to 2.5μA, 10μA, and 100μA.

The BOCS connect to the input buffer inputs after the internal multiplexer. When enabled, one BOCS sources current from AVDD to the selected positive analog input (AINP) and the other BOCS sinks current from the selected negative analog input (AINN) to GND.

In case of an open-circuit in the external sensor or a sensor wire disconnection, these BOCS pull the positive input toward AVDD and the negative input toward GND, resulting in a full-scale reading. A full-scale reading can also indicate that the sensor is overloaded or that the reference voltage is absent. A near-zero reading can indicate a shorted sensor. Distinguishing a shorted sensor condition from a normal reading can be difficult, especially if an RC filter is used at the inputs. The voltage drop across the external filter resistance and the residual resistance of the internal multiplexer created by the current sources can cause the ADC output to read a value larger than zero.

The ADC readings of a functional sensor can be corrupted when the burn-out current sources are enabled. Therefore, disable the burn-out current sources when performing precision measurements, and only enable the BOCS to test for sensor fault conditions during a dedicated diagnostic measurement.

The burn-out current sources remain active in idle mode, but turn off in standby and power-down mode.

7.3.14 Open Wire Detection with ADC 0-code output

Some applications require that the ADC provides a 0-code output when the inputs are in an open wire condition. This is an alternative method to detect an open wire in the system.

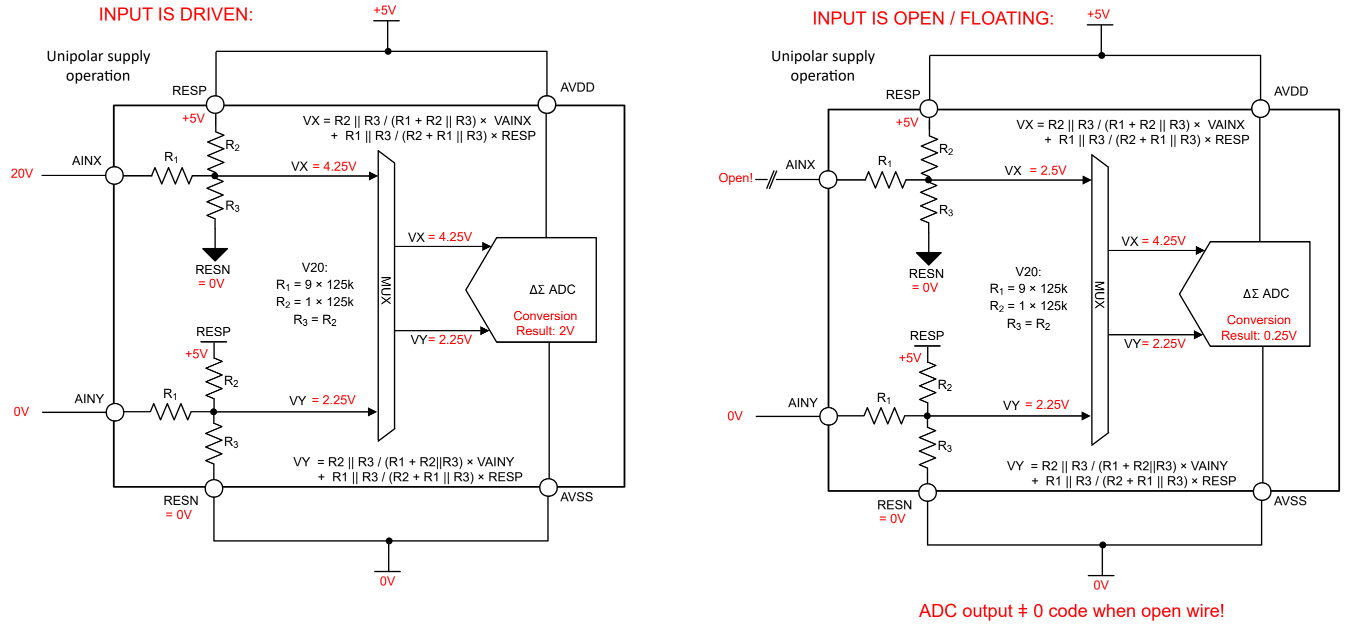

INPUT IS DRIVEN:
+5V
Unipolar supply operation
RESP
AVDD
+5V
VX = R2 || R3 / (R1 + R2 || R3) × VAINX + R1 || R3 / (R2 + R1 || R3) × RESP
R2
AINX
R1
20V
VX = 4.25V
R3
VX = 4.25V
RESN = 0V
V20:
R1 = 9 × 125k
R2 = 1 × 125k
R3 = R2
MUX
ΔΣ ADC
Conversion Result: 2V
RESP
+5V
VY= 2.25V
R2
AINY
R1
0V
VY = 2.25V
R3
VY = R2 || R3 / (R1 + R2||R3) × VAINY + R1 || R3 / (R2 + R1 || R3) × RESP
RESN = 0V
AVSS
0V
INPUT IS OPEN / FLOATING:
+5V
Unipolar supply operation
RESP
AVDD
+5V
VX = R2 || R3 / (R1 + R2 || R3) × VAINX + R1 || R3 / (R2 + R1 || R3) × RESP
R2
AINX
R1
Open!
VX = 2.5V
R3
VX = 4.25V
RESN = 0V
V20:
R1 = 9 × 125k
R2 = 1 × 125k
R3 = R2
MUX
ΔΣ ADC
Conversion Result: 0.25V
RESP
+5V
VY= 2.25V
R2
AINY
R1
0V
VY = 2.25V
R3
VY = R2 || R3 / (R1 + R2||R3) × VAINY + R1 || R3 / (R2 + R1 || R3) × RESP
RESN = 0V
AVSS
0V
ADC output ≠ 0 code when open wire!

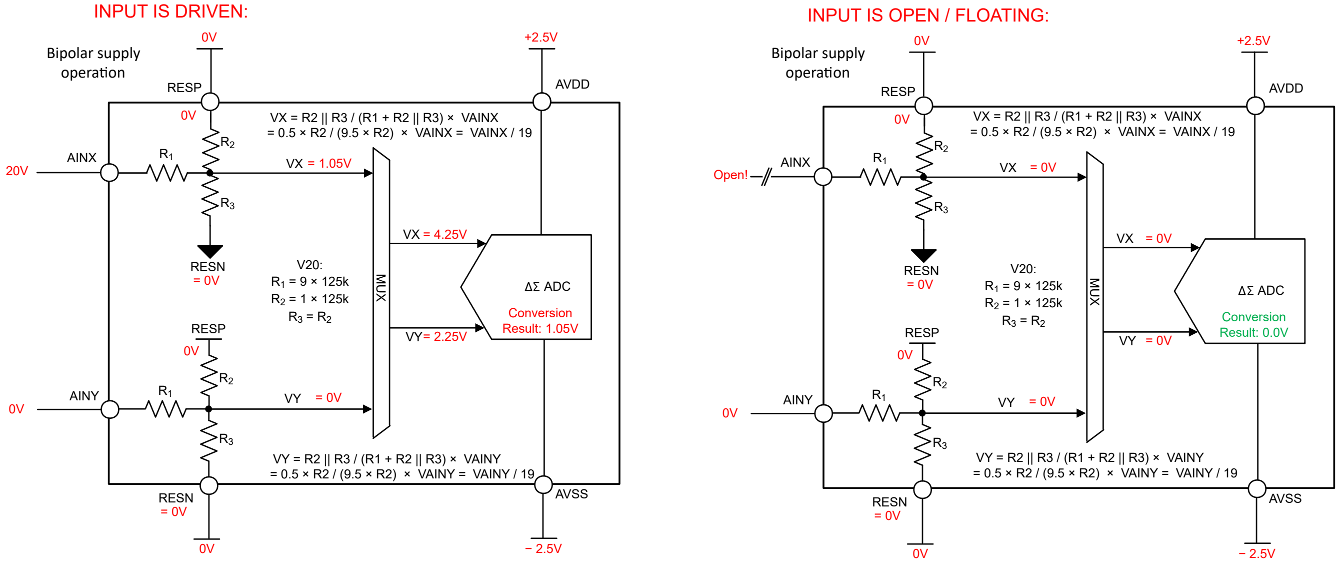
Figure 7-28. ADC Output for Open Wire Condition, RESP tied to AVDD

Assume the device is in unipolar supply operation, as shown in Figure 7-28. Due to the internal biasing from the resistor network divider at each input, an open wire (floating input) generates a non-zero input signal at the ADC input as shown on the right side in Figure 7-28. Therefore, the ADC provides non-zero code output for an open wire input.

The ADS125P08 has the capability to generate a zero-code output when inputs are floating (due to open wire). This mode is available because RESN and RESP are independent pins on the ADS125P08. Connect both

RESN and RESP pins to GND or DGND (i.e. 0V) for this feature, as shown in Figure 7-29. Operate the device with bipolar supplies as shown in Figure 7-29, if support of negative input voltages is required in this mode.

Figure 7-29. ADC Output for Open Wire Condition, RESP and RESN tied to GND



ADC output = 0 code when open wire!

The diagram on the right side in Figure 7-29 demonstrates the floating wire condition when both RESN and RESP pins are connected to GND (0V), and the device is operated with bipolar supply. The differential voltage at the ADC input is 0V, and the ADC outputs a conversion result close to 0-code in the open wire condition, as desired.

7.3.15 System Monitors

The ADS125P08 offer a set of system monitoring functions which can be routed to the buffer inputs internally for measurement through the input multiplexer. Use the STEPx_SYS_MON[3:0] bits (x = 0 to 31) to select from one of the following system monitors:

- The inputs of the high-impedance buffer can be shorted together to AVSS, to measure and calibrate the input offset of the internal signal chain.
- An integrated temperature sensor that provides an output signal proportional to the device temperature.
- The attenuated external reference voltage, (V_{REFP} – V_{REFN}) / 3.
- The attenuated analog and digital supplies, (AVDD / 3) and (DVDD / 3), respectively.
- The internal subregulated supplies, (CAPA – AVSS) / 1 and CAPD – DGND) / 1.

The AINP[4:0] and AINN[4:0] bits have no effect when one of the system monitors is selected. The analog inputs as well as the TDAC muxed signals are disconnected from the input buffer when a system monitor is selected. The internal 2.5V diagnostic reference is used for all settings.

Enable the high impedance input buffer using the BUF_MODE[1:0] bits to get improved accuracy for all system monitors. See the *High-Impedance Input Buffers* section for details on how to enable the input buffer.

7.3.15.1 Internal Short (Offset Calibration)

The system monitor offers the option to short both high-impedance buffer inputs (AINP and AINN) to AVSS. This option can be used to measure and calibrate the device offset by storing the result of the shorted input voltage reading in a microcontroller and consequently subtracting the result from each following reading. Take multiple readings with the inputs shorted and average the result to reduce the effect of noise.

7.3.15.2 Internal Temperature Sensor

The ADS125P08 provides an integrated temperature sensors (TS) to measure the die temperature. The temperature sensor outputs a voltage that is proportional to the die temperature. The output voltage characteristic (TS_{Offset}, TS_{TC}) of the temperature sensor is specified in the *Specifications* section.

Equation 26 shows how to convert the measured temperature sensor output voltage to die temperature:

$$\text{Die temperature [°C]} = 25\text{°C} + (\text{Measured voltage} - TS_{Offset}) / TS_{TC} \quad (26)$$

Make sure that the maximum temperature sensor output voltage that can occur in the application is smaller than V_{REF} = 2.5V.

7.3.15.3 External Reference Voltage Readback

The system monitor allows to monitor the external voltage reference connected between the REFP and REFN pins. For this purpose, select the attenuated external reference voltage, ($V_{REFP} - V_{REFN}$) / 3 for measurement.

7.3.15.4 Power-Supply Readback

The system monitor allows to monitor both the analog and digital supplies, as well as internal subregulated supplies. For this purpose, select either the attenuated analog supply (AVDD / 3) or the attenuated digital supply (DVDD / 3) for measurement. Alternatively, select the internal subregulated supplies, (CAPA – AVSS) / 1 or (CAPD – DGND) / 1.

7.3.16 Monitor Flags, Indicators and Counters

The ADS125P08 provides a set of monitors with according status flags to detect and indicate specific device or system faults to the host. Table 7-29 provides an overview of the available monitors. If a monitor detects a fault, the according low-active fault flag is set to 0b. Most monitors can be enabled or disabled as required using a dedicated monitor enable bit. If a monitor detects a fault, the according low-active fault flag is set to 0b. Except for the communication-related monitor fault flags, the fault flags must be cleared to 1b by the host after the fault condition is removed. The communication-related SPI CRC fault flag resets to 1b automatically in the SPI frame following a frame where no communication fault occurred.

Table 7-29. Monitor Overview

MONITOR NAME	MONITOR ENABLE BIT	MONITOR FAULT FLAG	FAULT FLAG REGISTER LOCATION	FAULT FLAG RESET MECHANISM
SUPPLY, ADC AND REFERENCE MONITORS				
Reset	N/A	RESETn	STATUS_MSB	Write 1b to clear bit to 1b
AVDD undervoltage	N/A	AVDD_UVn	ADC_REF_STATUS	Write 1b to clear bit to 1b
Reference undervoltage	REF_UV_EN	REF_UVn	ADC_REF_STATUS	Write 1b to clear bit to 1b
Modulator Overrange	N/A	MOD_OVR_FAULTn	ADC_REF_STATUS	Write 1b to clear bit to 1b
DIGITAL MONITORS				
Register map CRC	REG_MAP_CRC_EN	REG_MAP_CRC_FAULTn	DIGITAL_STATUS	Write 1b to clear bit to 1b
Memory map CRC	N/A	MEM_INTERNAL_FAULTn	DIGITAL_STATUS	Reset or power-cycle the device
Register map CRC fault page indicator	REG_MAP_CRC_EN	CRC_FAULT_PAGE[5:0]	DIGITAL_STATUS	Cleared to default value (111111b = no fault) when the REG_MAP_CRC_FAULTn flag is cleared
FIFO overflow flag	N/A	FIFO_OFn	FIFO_SEQ_STATUS	Write 1b to clear bit to 1b.
FIFO underflow flag	N/A	FIFO_UFn	FIFO_SEQ_STATUS	Write 1b to clear bit to 1b.
FIFO CRC fault flag	N/A	FIFO_CRC_FAULTn	FIFO_SEQ_STATUS	Write 1b to clear bit to 1b.
AGPIO readback	N/A	N/A	AGPIO_DATA_INPUT	N/A
GPIO readback	N/A	N/A	GPIO_DATA_INPUT	N/A

Table 7-29. Monitor Overview (continued)

MONITOR NAME	MONITOR ENABLE BIT	MONITOR FAULT FLAG	FAULT FLAG REGISTER LOCATION	FAULT FLAG RESET MECHANISM
COMMUNICATION MONITORS				
SPI CRC	N/A	SPI_CRC_FAULTn	STATUS_LSB	Updates in every new SPI frame based on the CRC result of the previous SPI frame
Register access	N/A	REG_WRITE_FAULTn	STATUS_LSB	Updates with the next register write command

Every monitor in the ADS125P08 has a corresponding fault flag (see Table 7-29), that sets to 0b when the respective monitor detects a fault condition. As shown in Table 7-29, the RESETn and communication related fault flags are located in the STATUS_MSB or STATUS_LSB register. However the ADC, reference, FIFO and memory related fault flags are grouped together in detailed status registers as shown in Table 7-30. Instead of reading the STATUS_MSB or STATUS_LSB registers on demand using a register read command, the ADS125P08 can output a STATUS header as the first two bytes of every frame on SDO as described in the *Section 7.5.3.2* section. Enable the STATUS header transmission using the STATUS_EN bit. The 16-bit STATUS header is a concatenation of the STATUS_MSB[7:0] and STATUS_LSB[7:0] register bits. To allow immediate indication of any of the ADC, reference, FIFO or memory related faults as part of the STATUS header, every detailed status register has a corresponding combined fault flag in the STATUS_MSB or STATUS_LSB register. That is, if any of the fault flags in the detailed status register set to 0b, then the combined fault flag sets to 0b as well.

Table 7-30. Detailed Status Registers and Corresponding Combined Fault Flags

DETAILED FAULT FLAGS FOR:	DETAILED FAULT FLAGS	DETAILED FLAG REGISTER LOCATION	COMBINED FAULT FLAG IN STATUS_MSB OR STATUS_LSB REGISTER
ADC and reference	AVDD_UVn REF_UVn MOD_OVR_FAULTn	ADC_REF_STATUS	ADC_REF_FAULTn
FIFO	FIFO_OFn FIFO_UFn FIFO_CRC_FAULTn	FIFO_SEQ_STATUS	FIFO_FAULTn
Memory / internal errors	MEM_INTERNAL_FAULTn REG_MAP_CRC_FAULTn	DIGITAL_STATUS	INTERNAL_FAULTn

To clear a set combined fault flag to 1b, the host must clear all set fault flags in the corresponding detailed status register. Only after all fault flags in the detailed status register are cleared to 1b can the host clear the combined fault flag by writing 1b.

In addition to the monitors that detect faults in the device, the ADS125P08 also provides the indicators and counters shown in Table 7-31, which provide feedback about the device state or behavior.

Table 7-31. Indicator and Counter Overview

INDICATOR OR COUNTER NAME	INDICATOR OR COUNTER BITS	INDICATOR OR COUNTER REGISTER LOCATION
Data ready indicator	DRDY	FIFO_SEQ_STATUS
Sequencer active	SEQ_ACTIVE	FIFO_SEQ_STATUS
Sequence step indicator	STEP_INDICATOR[4:0]	STATUS_MSB
ADC conversion counter	CONV_COUNT[3:0]	STATUS_LSB
FIFO depth indicator	FIFO_DEPTH[8:0]	FIFO_DEPTH_MSB, FIFO_DEPTH_LSB
Completed sequence counter	SEQ_COUNT[3:0]	FIFO_SEQ_STATUS

7.3.16.1 Reset (RESETn flag)

The RESETn flag indicates if a device reset happened since the last time the bit is cleared to 1b. Write 1b to clear the RESETn bit to 1b.

7.3.16.2 AVDD Undervoltage Monitor (AVDD_UVn flag)

The AVDD undervoltage monitor detects if the analog supply dropped below the AVDD undervoltage threshold (TH_{AVDD_UV}). Write 1b to clear the AVDD_UVn bit to 1b.

The AVDD undervoltage monitor is always active, except in power-down mode. AVDD_UVn sets to 0b when entering power-down mode even when the AVDD supply did not drop below the AVDD undervoltage threshold.

The device does not reset when the analog supply drops below the AVDD threshold as long as the DVDD supply is still present.

7.3.16.3 Reference Undervoltage Monitor (REV_UVn flag)

The reference undervoltage monitor detects if the reference voltage selected by the REF_SEL[1:0] bits dropped below the reference undervoltage threshold (TH_{REF_UV}). Write 1b to clear the REF_UVn bit to 1b. Enable the reference undervoltage monitor using the REF_UV_EN bit. However, the reference undervoltage monitor stops in standby and power-down mode.

7.3.16.4 Modulator Overrange Monitor (MOD_OVR_FAULTn flag)

The Modulator Overrange flag (MOD_OVR_FAULTn) indicates modulator saturation occurred during the conversion cycle. The flag is valid at the end of the conversion cycle. See the *Modulator* sections for details on the modulator and the modulator saturation flag.

7.3.16.5 Register Map CRC (REG_MAP_CRC_FAULTn flag)

The REG_MAP_CRC_FAULTn flag indicates if a register map CRC fault occurred. Enable the register map CRC using the REG_MAP_CRC_EN bit. However the register map CRC stops in standby and power-down mode. Write 1b to clear the REG_MAP_CRC_FAULTn bit to 1b.

The register map CRC is used to check the register map contents for unintended changes. Each register page has a separate CRC register: The CRC register for the Status and General Configuration Page is the REG_MAP_CRC register, and the CRC register for each sequence step page is the corresponding STEPx_REG_MAP_CRC register (x = 0 to 31). Write a new register map CRC code to the REG_MAP_CRC or STEPx_REG_MAP_CRC register whenever the registers are changed. For each page, the CRC code is calculated over the register addresses shown in Table 7-32. The ADC continuously compares the CRC code written to the CRC register to an internally calculated value. If the values do not match, the REG_MAP_CRC_FAULTn bit in the DIGITAL_STATUS register is set. If set, correct the register values or update the CRC code then write 1b to the REG_MAP_CRC_FAULTn bit to clear the error flag. The CRC_FAULT_PAGE[5:0] bits in the DIGITAL_STATUS register indicate which register page showed a CRC error when the REG_MAP_CRC_FAULTn indicates a CRC fault. This field is cleared to default value (111111b = no fault) when the REG_MAP_CRC_FAULTn flag is cleared.

Table 7-32. Registers used in CRC

Register Page	Registers used for CRC Calculation[1]
Status and General Configuration Page	0x12 to 0x18 0x20 to 0x2D 0x30 to 0x32
Step Configuration Page (0 to 31)	0x00 to 0x10

[1] All reserved (unused) bits set to 0b for CRC calculation.

The register map CRC code computation is the same as the one shown in the *SPI CRC* section.

Set the REG_MAP_CRC_EN bit to 1b (enabled) when calculating the CRC code on the Status and General Configuration Page because this bit must be set to enable the CRC check.

Use the following procedure to change register bits without accidentally causing a REG_MAP_CRC_FAULTn indication:

- Disable the register map by setting REG_MAP_CRC_EN = 0b
- Wait the fault response time $t_{p(REG\_MAP\_CRC)}$
- If the REG_MAP_CRC_FAULTn flag is set to 0b, clear the fault flag by writing 1b to the REG_MAP_CRC_FAULTn bit
- Optional: Verify the REG_MAP_CRC_FAULTn fault flag is cleared to 1b
- Change the device register bits as needed
- Update the REG_MAP_CRC[7:0] bits based on the new register map settings
- Enable the register map CRC by setting REG_MAP_CRC_EN = 1b

Register bits can also be changed while the register map CRC is enabled, as discussed in the following procedure, but can cause unintended REG_MAP_CRC_FAULTn indications.

- Change the register bits as needed while the register map CRC is enabled
- Update the REG_MAP_CRC[7:0] bits based on the new register map settings
- Wait the fault response time $t_{p(REG\_MAP\_CRC)}$
- If the REG_MAP_CRC_FAULTn flag is set to 0b, clear the fault flag by writing 1b to the REG_MAP_CRC_FAULTn bit
- Optional: Verify the REG_MAP_CRC_FAULTn fault flag is cleared to 1b

7.3.16.6 Memory Map CRC (MEM_INTERNAL_FAULTn flag)

The MEM_INTERNAL_FAULTn flag indicates if a memory map CRC fault occurred, or if a wrong register page is selected internally.

Similar to the register map CRC, the device uses a memory map CRC to check the internal memory for random bit changes. Changes to the internal memory bits can cause undetermined device behavior or degraded device performance. The memory map CRC is always enabled, except in standby and power-down mode,and constantly calculates the CRC value across the internal memory map. The device compares the calculation result against a memory map CRC value that is stored in the memory map in production. If the internal calculation result and the stored memory map CRC value do not match, the MEM_INTERNAL_FAULTn flag is set to 0b. No other action is taken by the device in the event of a memory map CRC fault.

The CRC calculation is implemented serially, one memory map word per internal clock oscillator period. Therefore random bit changes are not indicated immediately in the MEM_INTERNAL_FAULTn flag, but can take up to $t_{p(MEM\_MAP\_CRC)}$.

The MEM_INTERNAL_FAULTn flag is also set to 0b if a wrong page is selected internally (the PAGE_INDICATOR does not match the PAGE_POINTER).

In case of a memory map CRC fault or wrong page selected fault, which both trigger the MEM_INTERNAL_FAULTn flag, write 1b to the MEM_INTERNAL_FAULTn bit to clear the fault flag to 1b. Perform a power cycle or reset the device if the fault flag continues to set to 0b.

7.3.16.7 FIFO Overflow (FIFO_OFn flag) and FIFO Underflow (FIFO_UFn flag)

The FIFO overflow flag (FIFO_OFn) detects an overflow condition in the FIFO buffer, and the FIFO underflow flag (FIFO_UFn) detects an underflow condition in the FIFO buffer. Both flags are only available when the FIFO is enabled. See the *FIFO Buffer* section for details about the FIFO overflow flag and FIFO underflow flag implementation.

7.3.16.8 FIFO CRC Fault (FIFO_CRC_FAULTn flag)

The FIFO CRC fault flag (FIFO_CRC_FAULTn flag) detects a CRC error during FIFO read. This flag is only available if the FIFO is enabled. See the *FIFO Buffer* section for details about the FIFO CRC implementation.

7.3.16.9 GPIO Readback

All available GPIOs in ADS125P08 provide an independent readback path when the respective GPIO is configured as digital output. That means an input receiver circuit independent from the output driver circuit detects the voltage level driven on the GPIO pin. The result of the readback is displayed in the according STEPx_GPIOn_DAT_IN bits (x = 0 to 31, n = 0 to 3) and STEPx_AGPIOn_DAT_IN bits (x = 0 to 31, n = 0 to 7).

7.3.16.10 SPI CRC Fault (SPI_CRC_FAULTn flag)

The SPI_CRC_FAULTn flag indicates if a SPI CRC fault occurred on SDI in the previous SPI frame. The execution of the command in the frame where the SPI CRC fault occurred is blocked. Instead, a no operation command is executed. Commands in following frames are not blocked. The SPI_CRC_FAULTn bit updates in every new SPI frame based on the CRC result of the previous SPI frame. Enable the SPI CRC using the SPI_CRC_EN bit. In addition, enable the transmission of the STATUS word using the STATUS_EN bit to get notified about any SPI CRC faults. See the *SPI CRC* section for details about the SPI CRC implementation.

7.3.16.11 Register Write Fault (REG_WRITE_FAULTn flag)

The REG_WRITE_FAULTn flag indicates if a write access to an invalid register address occurred. This flag sets when an invalid register address is written to, and updates at the next register write command. Reading from an invalid register address does not set the flag, but can be detected from the address indication inside the SPI frame of the read command.

7.3.16.12 DRDY Indicator (DRDY bit)

The DRDY bit is the inverse of the $\overline{\text{DRDY}}$ pin. Enable the transmission of the STATUS header using the STATUS_EN bit to leverage the DRDY bit indication. The DRDY bit indicates if the conversion data read within the current SPI frame are new or are repeated data from the last read operation. Polling the DRDY bit using the register read command is not reliable, because the DRDY bit reads 0b in most cases.

7.3.16.13 Sequencer Active Indicator (SEQ_ACTIVE bit)

The sequencer active indicator bit (SEQ_ACTIVE bit) indicates if conversions are currently ongoing or if conversions stopped and the device is in idle, standby or powerdown mode.

This bit sets when the sequencer is enabled and a start condition is initiated on the START pin or setting the START bit. The bit resets automatically when the device enters standby mode or idle mode after the sequence has finished (by either stop bit or reaching the end of the sequence).

See the *Channel Auto-Sequencer* section for details on the sequencer operation, including the available sequencer modes.

7.3.16.14 Sequence Step Indicator (STEP_INDICATOR[4:0])

The sequence step indicator (STEP_INDICATOR[4:0]) indicates the step page configuration that is used for the latest conversion result, which is currently available for readout. The step indicator resets to 00h after a device reset, in powerdown mode, or when writing to the SEQUENCER_CFG register. At the same time the conversion counter (CONV_COUNT[3:0]) resets to Fh, the sequence counter (SEQ_COUNT[3:0]) resets to 0h, and the conversion data clears.

See the *Channel Auto-Sequencer* section for details on the sequencer operation, including the available sequencer modes, and for details on the sequencer indicators.

7.3.16.15 ADC Conversion Counter (CONV_COUNT[3:0])

The conversion counter CONV_COUNT[3:0]) increments every time a new conversion completes. After reaching a counter value of Fh, the counter rolls over to 0h with the completion of the next conversion.

The counter only resets to Fh in powerdown mode, after a device reset, or when writing to the SEQUENCER_CFG register. At the same time the sequence step indicator (STEP_INDICATOR[4:0]) resets to 00h, the sequence counter (SEQ_COUNT[3:0]) resets to 0h, and the conversion data clears. At the completion of the first conversion after reset, powerdown, or after writing to the SEQUENCER_CFG register, the counter reads 0h. When the sequencer is enabled (SEQ_MODE[1:0] = 10b or 11b), the counter always reads 0h for the first conversion of a step. When the sequencer is disabled (SEQ_MODE[1:0] = 00b or 01b), the counter value does not return to 0h if conversions with a new step page configuration complete.

When the conversion counter is output as part of the STATUS header (STATUS_EN = 1b), the device makes sure that the conversion counter value always matches to the ADC conversion result that is output in the same SPI frame.

Reset the counter to Fh by writing to the SEQUENCER_CFG register before starting a conversion with a new step page configuration if desired.

7.3.16.16 FIFO Depth Indicator (FIFO_DEPTH[8:0])

The FIFO depth indicator (FIFO_DEPTH[8:0] bits) reports the depth of the FIFO, which is the difference between the FIFO write pointer and read pointer. The indicator is only available when the FIFO is enabled. See the *FIFO Buffer* section for details about the FIFO depth indicator.

7.3.16.17 Completed Sequence Counter (SEQ_COUNT[3:0])

The sequence counter (SEQ_COUNT[3:0]) indicates the progression trough multiple sequence runs. When SEQ_MODE[1:0] = 11b, the sequence counter indicates which sequence run the latest conversion result belongs to, which is currently available for readout. The sequence counter increments with the completion of the first conversion of a new sequence run. At the completion of the first conversion of the first sequence run, the counter reads 0h. At the completion of the first conversion of the second sequence run, the counterreads 1h. After reaching a counter value of Fh, the counter rolls over to 0h with the completion of the first conversion of the next sequence run.

The counter resets to 0h at the completion of the first conversion after setting the START bit to 1b or at the rising edge of the START pin. When writing to the SEQUENCER_CFG register, in powerdown mode, or after a device reset, the counter resets to 0h immediately. At the same time the sequence step indicator (STEP_INDICATOR[4:0]) resets to 00h, the conversion counter (CONV_COUNT[3:0]) resets to Fh, and the conversion data clears. The sequence counter always reads 0h when SEQ_MODE[1:0] = 00b, 01b, or 10b.

See the *Channel Auto-Sequencer* section for details on the sequencer operation, including the available sequencer modes.

7.3.17 Test DAC (TDAC)

The ADS125P08 includes a test voltage digital-to-analog converter (TDAC) intended for ADC self-testing and verification. The 5-bit TDAC is capable of providing single-ended test voltages. The minimum TDAC output is 1/32 of the auxiliary voltage reference. The maximum TDAC output is equal to the auxiliary voltage reference.

As shown in the *Functional Block Diagram*, the unbuffered (and more accurate) TDAC output can also be directly connected to the high-impedance buffer inputs AINP and AINN. Set the STEPx_TDAC_SEL[4:0] to 00001b (positive input) or 00010b (negative input) to use this mode. The other input is connected to AVSS in each case. In this mode, the input multiplexer is forced open. If the user also selects a System Monitor using STEPx_SYS_MON[3:0] at the same time, the System Monitor has priority and the unbuffered TDAC is not used.

When using the buffered TDAC, a dedicated multiplexer routes the TDAC to the selected input as defined by the STEPx_TDAC_SEL[4:0] bits. The input multiplexer selection is unaffected. Set the STEPx_TDAC_SEL[4:0] to 10011b to route the buffered TDAC output to the REFP/TDACOUT pin.

Table 7-33 shows all available configurations of the TDAC multiplexer.

Table 7-33. TDAC Multiplexer Configurations

STEPx_TDAC_SEL[4:0] (x = 0 to 31)	DESCRIPTION
00000b	Mux open, TDAC not connected
00001b	TDAC unbuffered to positive input; negative input is connected to AVSS
00010b	TDAC unbuffered to negative input, positive input is connected to AVSS
00011b	Buffered TDAC to AIN0

Table 7-33. TDAC Multiplexer Configurations (continued)

STEPx_TDAC_SEL[4:0] (x = 0 to 31)	DESCRIPTION
00100b	Buffered TDAC to AIN1
00101b	Buffered TDAC to AIN2
00110b	Buffered TDAC to AIN3
00111b	Buffered TDAC to AIN4
01000b	Buffered TDAC to AIN5
01001b	Buffered TDAC to AIN6
01010b	Buffered TDAC to AIN7
01011b	Buffered TDAC to AIN8
01100b	Buffered TDAC to AIN9
01101b	Buffered TDAC to AIN10
01110b	Buffered TDAC to AIN11
01111b	Buffered TDAC to AIN12
10000b	Buffered TDAC to AIN13
10001b	Buffered TDAC to AIN14
10010b	Buffered TDAC to AIN15
10011b	Buffered TDAC to REFP/TDACOUT pin
all other codes	Mux open, TDAC not connected

Set the STEPx_TDAC_VAL[4:0] bits to select the TDAC output value. The auxiliary reference value is the same as the value selected for the ADC reference by the REF_VAL bit on the general configuration page (01b = 2.5V, and 1b = 4.096V). Table 7-34 shows the TDAC output voltages depending on the STEPx_TDAC_VAL[4:0] bits and REF_VAL values.

Table 7-34. TDAC Output Voltage

STEPx_TDAC_VAL[4:0] (x = 0 to 31)	Value on TDAC	Value for REF_VAL = 0b (2.5V)	Value for REF_VAL = 1b (4.096V)
00000b	(1 / 32) × Auxiliary Vref	0.078V	0.128V
00001b	(2 / 32) × Auxiliary Vref	0.156V	0.256V
...	...	...	...
nnnnn	((nnnnn + 1) / 32) × Auxiliary Vref)	...	...
11110b	(31 / 32) × Auxiliary Vref	2.422V	3.968V
11111b	(32 / 32) × Auxiliary Vref	2.5V	4.096V

7.3.18 Parallel Post Filters

The ADS125P08 integrates parallel post filters to enable filter notches at targeted frequencies without reducing the data rate.

As shown in Figure 7-30, up to 8 post filters are available to use in parallel. These filters are cascaded moving average filters. The filter order and number of averages are user programmable. The post filters are similar to traditional sinc filters with the difference that there is no decimation within the parallel filters. The input data rate of each post filter equals the output data rate once the filter is settled.

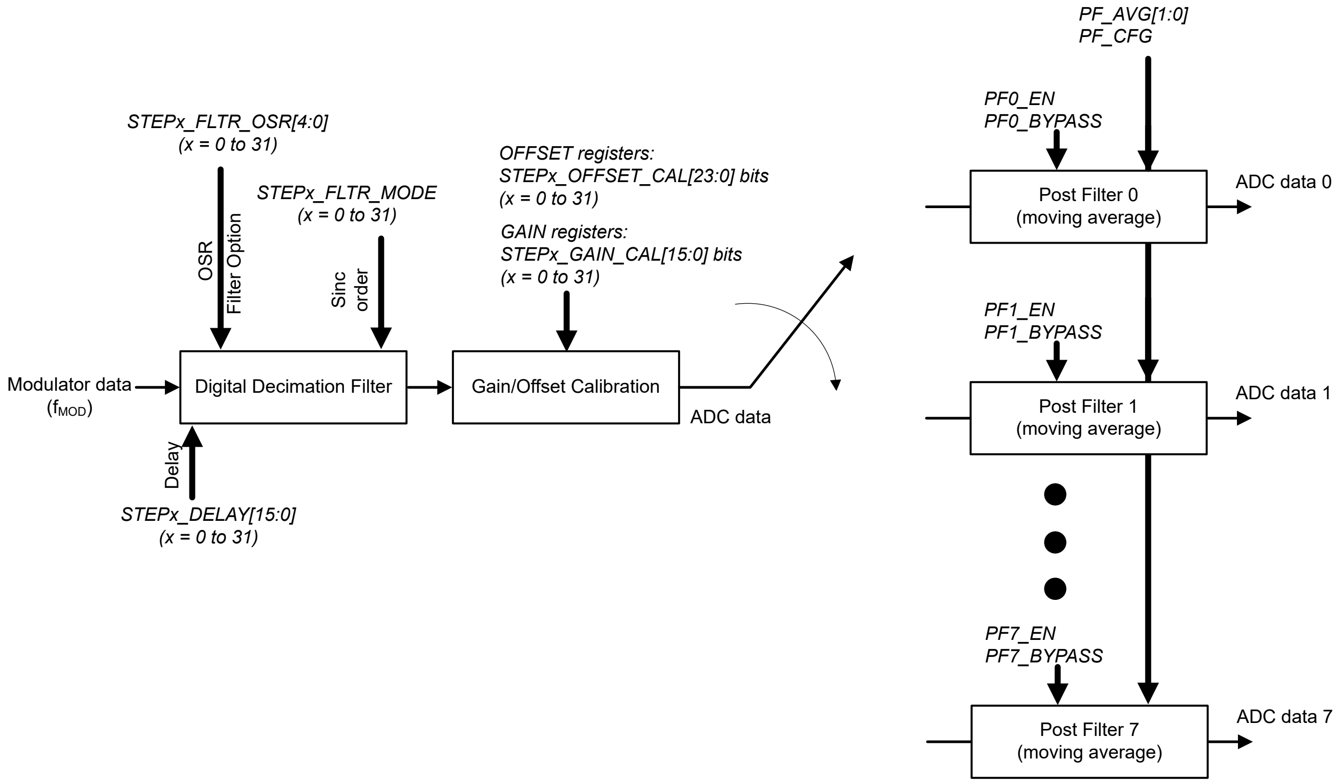


Figure 7-30. Parallel Post Filter Block Diagram

The sequencer must be enabled to use the post filters. The parallel filters are reset whenever the sequencer starts or restarts.

7.3.18.1 Configuring the Parallel Post Filters

The post filters 0 to 7 are assigned to corresponding sequence steps 0 to 7. Any other step enabled in a sequence cannot be post filtered and is ignored. When a post filter is used on a sequence step, the number of conversions for that step is forced to 1. The main filter OSR and conversion-start delay time are as configured in the active step page: For example, the STEP0_FLTR_CFG register sets the filter configuration for the step 0 and the post filter 0, and the STEP0_DELAY_MSB_CFG and STEP0_DELAY_LSB_CFG registers set the delay time which is inserted before the first ADC conversion of each sequence step.

The post filters are individually enabled by the PFn_EN (n = 0 to 7) register bits. Each enable corresponds to the respective step number. For example, PF3_EN enables the post filter for sequence step 3. Disabling a post filter saves power. Note the associated sequencer step must be enabled to use the post filter. If the sequencer is disabled, the PFn_EN (n = 0 to 7) bits are ignored.

Figure 7-31 depicts the timing sequence for the example of all 8 post filters enabled, and sinc4 filter option is selected in the digital decimation filter. Each sequence step corresponds to one post filter, meaning one channel. Cue to the number of conversions forced to 1, each sequence step is comprised of the programmable delay and only one fully settled conversion result before switching to the next step/channel.

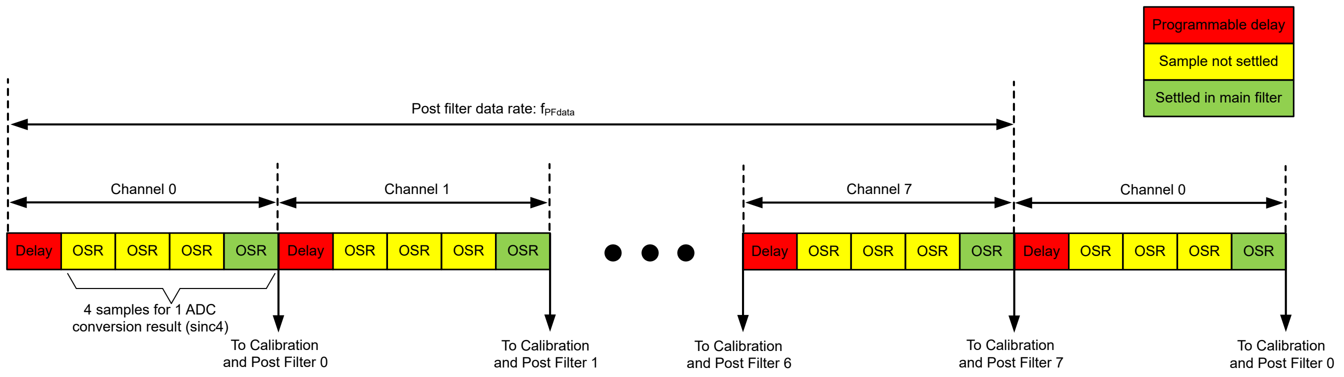

Programmable delay
Sample not settled
Settled in main filter
Post filter data rate: f_PFdata
Channel 0
Channel 1
Channel 7
Channel 0
Delay
OSR
OSR
OSR
OSR
4 samples for 1 ADC conversion result (sinc4)
To Calibration and Post Filter 0
To Calibration and Post Filter 1
To Calibration and Post Filter 6
To Calibration and Post Filter 7
To Calibration and Post Filter 0

Figure 7-31. Parallel Post Filter Timing Diagram

Equation 27 describes the transfer function of each post filter:

$$H(z) = \left(\frac{1 + z^{-1} + \ldots + z^{-(N-1)}}{\mathrm{Navr}}\right)^{\mathrm{Mord}} \tag{27}$$

where:

- Navr is the number of averages set by the PF_AV[1:0] bits
- Mord is the filter order set by the PF_CFG bit

The number of averages within the post filters and the filter order settings apply to all 8 post filters. Table 7-35 shows the available options for the number of averages, and Table 7-36 shows the available options for the filter order. The post filters are reset whenever PF_AV[1:0] or PF_CFG are changed or whenever the main filter is reset.

Table 7-35. Parallel Post Filter Average Control

PF_AV[1:0]	NUMBER OF AVERAGES
00	4
01	8
10	16
11	16

Table 7-36. Parallel Post Filter Order Control

PF_CFG	POST FILTER ORDER
0	1 (similar to sinc 1)
1	3(similar to sinc 3)

A post filter bypass control exists for each post filter, using the PFn_BYPASS (n = 0 to 7) bits. This control allows the user to select data before or after the post filter. The user can for example monitor the faster main filter output while the post filter is processing the data and then at a later time retrieve the averaged value. If PFn_BYPASS = 0, the corresponding post filter is enabled. If PFn_BYPASS = 1 , the post filter is bypassed. Post filter bypass can be changed as the sequencer is running, allowing the user to dynamically read data before or after the post filter. PFn_BYPASS does not switch the output data right away. PFn_BYPASS just tells the sequencer which data is put into the output data buffer when new data is generated, either the main filter data or the post filter data. All sequence steps share the same output data buffer. The output data buffer is updated only when new data is generated, either the main filter data or the post filter data depending on PFx_BYPASS setting of the active step. If the sequencer is disabled, the PFn_BYPASS (n = 0 to 7) bits are ignored.

If all post filters are disabled (when not using the post filters), when PFn_EN = 0 (n = 0 to 7), make sure that all PFn_BYPASS (n = 0 to 7) bits are set to the default value PFn_BYPASS = 1 (n = 0 to 7). Setting any of the

PFn_BYPASS (n = 0 to 7) bits to 0 can enable the post filters unintentionally, even if all enable bits PFn_EN = 0 (n = 0 to 7) are set to 0.

7.3.18.2 Frequency Response of the Parallel Post Filters

The data rate of the post filter results from the data rate of the main filter (meaning the digital decimation filter shown in Figure 7-30) and the number of channels. Equation 28 defines the data rate of the main filter assuming sinc4 decimation:

$$f_{\mathrm{ADCdata}} = \frac{f_{\mathrm{MOD}}}{\mathrm{Delay} + 4 \times \mathrm{OSR}} \tag{28}$$

where:

- f MOD is the modulator frequency
- OSR is the oversampling ratio set by the STEPx_FLTR_OSR[4:0] bits
- Delay is the delay time which is inserted before the first ADC conversion of each sequence step, set by the STEPx_DELAY[15:0] bits

Equation 29 describes the post filter data rate, which is also the scan rate of the sequencer (that is, the frequency at which the sequencer repeats operation):

$$f_{\mathrm{PFdata}} = \frac{f_{\mathrm{ADCdata}}}{\mathrm{channel\_num}} = \frac{f_{\mathrm{MOD}}}{(\mathrm{Delay} + 4 \times \mathrm{OSR}) \times \mathrm{channel\_num}} \tag{29}$$

where:

channel_num is the number of channels with post filters enabled (typically 2, 4, or maximum of 8)

A straightforward way to confirm the scan rate by measurement is to configure $\overline{\mathrm{DRDY}}$ to toggle every time the sequencer repeats, meaning at the end of each completed sequence. This way the user can measure the sequence rate directly.

The frequency response of all enabled post filters is the same, and is set by the filter configuration (number of averages N and filter order M) and the post filter data rate (which is the scan rate).

Equation 30 defines the notch frequency for each parallel post filter:

$$f_{\mathrm{PFnotch}} = \frac{f_{\mathrm{PFdata}}}{\mathrm{Navr}} = \frac{f_{\mathrm{MOD}}}{(\mathrm{Delay} + 4 \times \mathrm{OSR}) \times \mathrm{channel\_num} \times \mathrm{Navr}} \tag{30}$$

where:

Navr is the number of averages set by the PF_AV[1:0] bits

For example, assuming the modulator frequency is 12.8MHz, the sinc4 OSR is set to 128, and the delay is set to 488, with 8 channels enabled, the post filter data rate/scan rate is 1.6kHz and the notch frequency with Navr = 4 is 400Hz. Figure 7-32 shows the frequency response for each postfilter in this configuration.

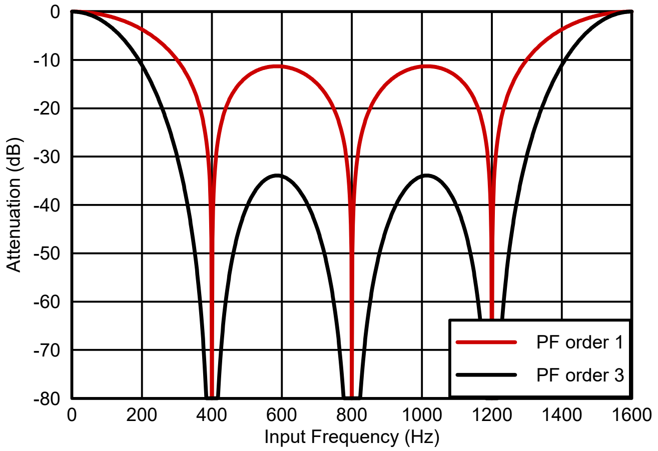


Figure 7-32. Parallel Post Filter Frequency Response, 4 averages and scan rate of 1.6kHz

In another example, if modulator frequency is 12.8MHz, sinc4 OSR is 512, and the delay set to 1285, with 8 channels enabled, the post filter data rate/scan rate is 480Hz and the notch frequency with Navr = 8 is 60Hz. Figure 7-33 shows the frequency response for each postfilter in this configuration.

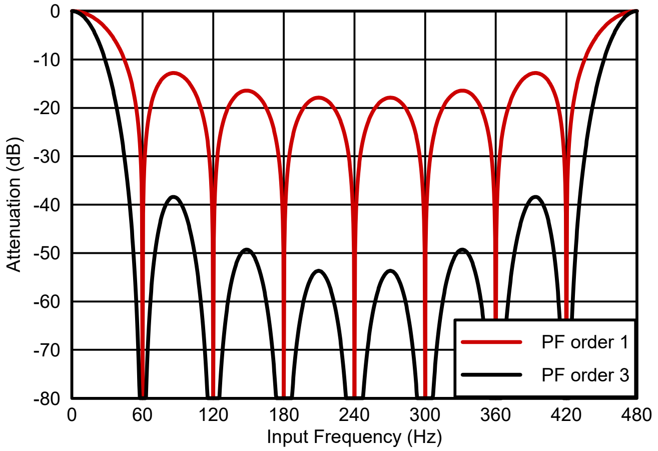


Figure 7-33. Parallel Post Filter Frequency Response, 8 averages and scan rate of 480Hz

Using the filter order Mord = 3 (similar to a sinc3) results in a wider notch than filter order Mord = 1 (similar to a sinc1) with the cost of longer latency, see Figure 7-32 and Figure 7-33. Note that total frequency response for a given step using a post filter is the composite of the main filter and the post filter responses.

7.3.18.3 Settling Times and $\overline{\text{DRDY}}$ Behavior When Using the Post Filters

While the input and output data rates are the same (when settled), there is skew between these as the parallel filter needs time to process the inputs. Table 7-37 shows additional delays for $\overline{\text{DRDY}}$ transition due to the internal processing in the post filters, and Table 7-38 shows the settling time for different post filter configurations.

Table 7-37. Parallel Post Filter Processing Time ($\overline{\text{DRDY}}$ delay)

NUMBER OF AVERAGES	$\overline{\text{DRDY}}$ DELAY TIME IN MODCLKs	
	PF ORDER 1	PF ORDER 3
4	5	15
8	9	27
16	17	51

Table 7-38. Parallel Post Filter Settling Time

NUMBER OF AVERAGES	PF ORDER 1	PF ORDER 3
4	4th data is settled	10th data is settled
8	8th data is settled	22nd data is settled
16	16th data is settled	46th data is settled

The behavior of the $\overline{\text{DRDY}}$ pin when using the post filters is set by the DRDY_CFG[1:0] bits, to transition either after each completed conversion, or after each sequence step or after a sequence is complete, see the *Auto-Sequencer and $\overline{\text{DRDY}}$ Behavior* section for details. However after a sequencer start, when the parallel post filters are enabled and the post filter output is selected as the data output (PFn_BYPASS = 0), $\overline{\text{DRDY}}$ transitions for the first time only if the post filter has settled as defined in Table 7-38. Thereafter, $\overline{\text{DRDY}}$ transitions as defined by the DRDY_CFG[1:0] bits, but with an additional delay specified in Table 7-37. The additional delay as shown in Table 7-37 adds on to the inherent delay of the calibration engine, which is inserted after the main filter is settled (i.e. after every 4th ADC conversion result in the sinc4 case), as a constant delay 6.5 modulator clock cycles. For example in the case of 4 averages and post filter order 1, the total delay for each $\overline{\text{DRDY}}$ transition with respect to the completion of ADC settled conversion result is 6.5 + 5 = 11.5 modulator clock periods. Figure 7-34 illustrates this behavior for the configuration where number of averages = 4, PF order = 1, and channel_num = 8. The first $\overline{\text{DRDY}}$ transition occurs after the 25th ADC data in this case, see Figure 7-34.

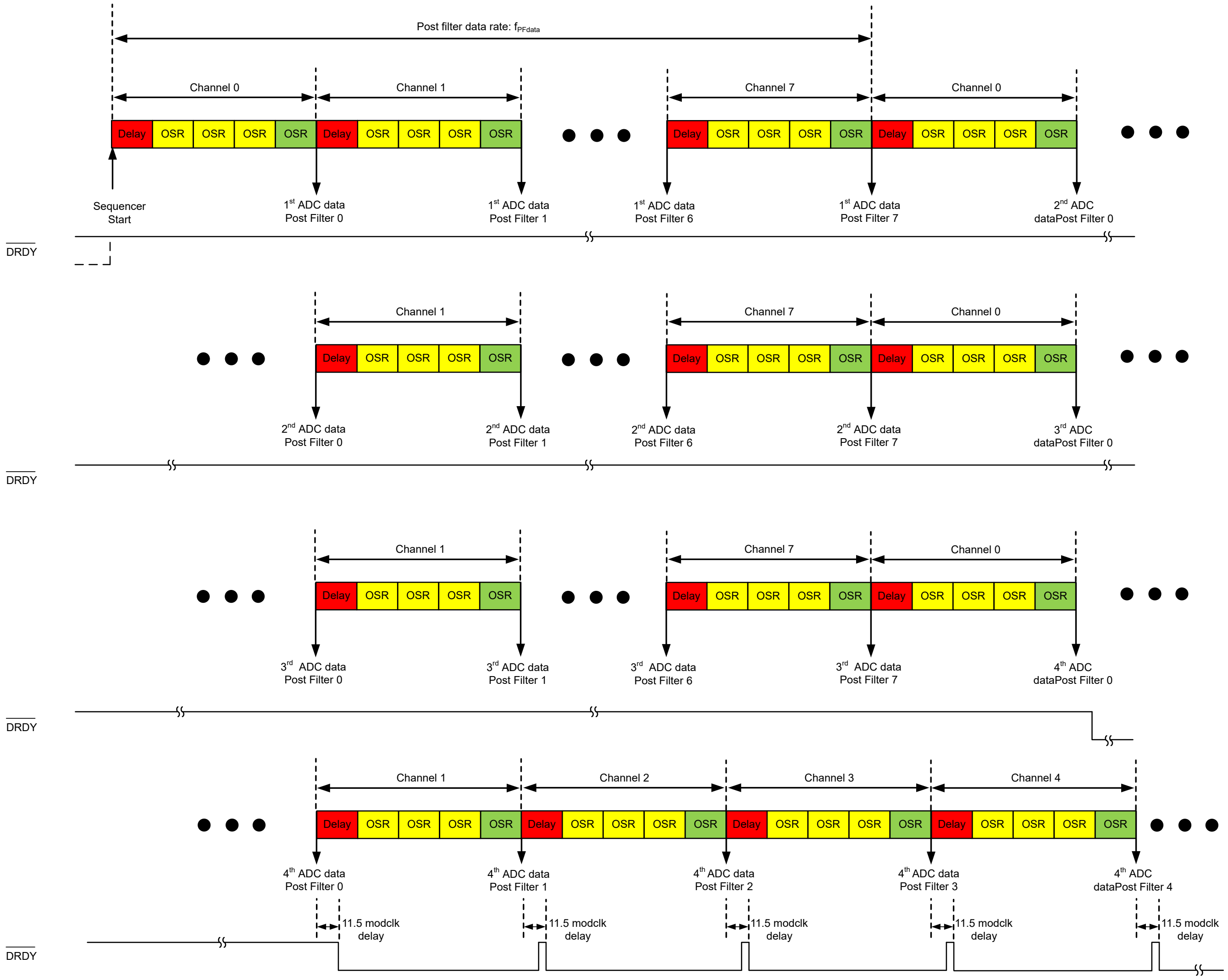

Post filter data rate: f_PFdata
Channel 0
Channel 1
Channel 7
Channel 0
Delay
OSR
Sequencer Start
1st ADC data Post Filter 0
1st ADC data Post Filter 1
1st ADC data Post Filter 6
1st ADC data Post Filter 7
2nd ADC dataPost Filter 0
DRDY
2nd ADC data Post Filter 0
2nd ADC data Post Filter 1
2nd ADC data Post Filter 6
2nd ADC data Post Filter 7
3rd ADC dataPost Filter 0
DRDY
3rd ADC data Post Filter 0
3rd ADC data Post Filter 1
3rd ADC data Post Filter 6
3rd ADC data Post Filter 7
4th ADC dataPost Filter 0
DRDY
Channel 1
Channel 2
Channel 3
Channel 4
4th ADC data Post Filter 0
4th ADC data Post Filter 1
4th ADC data Post Filter 2
4th ADC data Post Filter 3
4th ADC dataPost Filter 4
11.5 modclk delay
11.5 modclk delay
11.5 modclk delay
11.5 modclk delay
11.5 modclk delay
DRDY

Figure 7-34. $\overline{\text{DRDY}}$ Behavior When Using the Post Filters

7.3.18.4 Examples of Recommended Post Filter Settings

Table 7-39 shows examples for some commonly used notch-frequencies and how to implement them with specific post-filter settings. The top 5 rows show the recommended configuration settings for the postfilters (how many post filters, meaning channels are enabled, number of averages, OSR, and delay with a given modulator frequency), and the bottom 3 rows show the calculated performance in terms of ADC data rate, scan rate and post filter notch frequency.

Table 7-39. Post Filter Configurations for Common Notch-frequencies

# of Ch	**8**	**8**	**8**	**8**	**8**	**4**	**4**	**4**	**4**	**4**	**2**	**2**	**2**	**2**	**2**
Navr	**4**	**4**	**4**	**4**	**4**	**4**	**4**	**4**	**4**	**4**	**4**	**4**	**4**	**4**	**4**
SINC4 OSR	**128**	**1024**	**1024**	**4000**	**8000**	**256**	**2048**	**4000**	**8000**	**16000**	**512**	**4000**	**8000**	**16000**	**32000**
DELAY	**488**	**2570**	**3904**	**7952**	**8000**	**976**	**5141**	**0**	**15904**	**16000**	**1952**	**10666**	**0**	**31808**	**32000**
f_MOD (MHz)	**12.8**	**12.8**	**12.8**	**12.8**	**12.8**	**12.8**	**12.8**	**12.8**	**12.8**	**12.8**	**12.8**	**12.8**	**12.8**	**12.8**	**12.8**
ADC data rate (kHz)	12.800	1.920	1.600	0.534	0.320	6.400	0.960	0.800	0.267	0.160	3.200	0.480	0.400	0.134	0.080
Scan rate (kHz)	1.600	0.240	0.200	0.067	0.040	1.600	0.240	0.200	0.067	0.040	1.600	0.240	0.200	0.067	0.040
Notch freque ncy f_notch (Hz)	400.0	60.0	50.0	16.7	10.0	400.0	60.0	50.0	16.7	10.0	400.0	60.0	50.0	16.7	10.0

7.3.19 Chip Select Forwarding

The ADS125P08 offers a chip select forwarding (CS-FWD) function which can be used to pass SPI control to other devices. When this feature is enabled, a new command frame frequently called User Datagram Protocol (UDP) is used to communicate with the ADS125P08 as well as other devices. The CS-FWD function allows controlling the CS of other devices through the ADS125P08. If the user does not enable the CS-FWD function as described in the *Configuring the CS forward feature* section, then normal SPI communication is assumed by the ADS125P08.

The CS-FWD feature allows to reduce the number of isolation channels in the system and enables seamless communication between the microcontroller on one side of the isolation barrier and multiple chips (including the ADS125P08) on the other side. In contrast to the alternative method of Daisy-Chain connection, the CS-FWD function does not increase the length of the SPI frame proportionally with the number of the target devices. Thus, the CS-FWD has the benefit of avoiding inefficient long SPI frames.

In general, there are three User Datagram Protocol (UDP) host controller modes of operation: UDP Phase0-CS, UDP Phase1-CS and UDP Secondary-CS. The ADS125P08 only supports the UDP Phase0-CS mode (which uses the same phase of $\overline{\text{CS}}$, active low).

An example of the phase0-CS mode connectivity is shown in Figure 7-35. The host device (denoted MCU host - for example, a microcontroller) provides SDI, SDO, SCLK and the CS. In this mode, the CS of the host device is transferred. Phase0-CS mode uses the same phase of CS, meaning active low, to transfer subsequent CS pulses to the secondary device (SPI Dest1 to Dest4 in Figure 7-35). The multi-purpose GPIO pins are used in that mode to transfer the CS signal to the destination device (see the *Configuring the CS forward feature* section for details). The red lines in Figure 7-35 indicate a situation where destination device 1 is selected as the target device for communication.

Add pullup resistors (as shown in Figure 7-35) to the GPIO pins which are intended to be used for CS forwarding. This is not mandatory, but can keep the node high by default, for example, in a power-up condition, avoiding unintentional communication to a target device which is not selected.

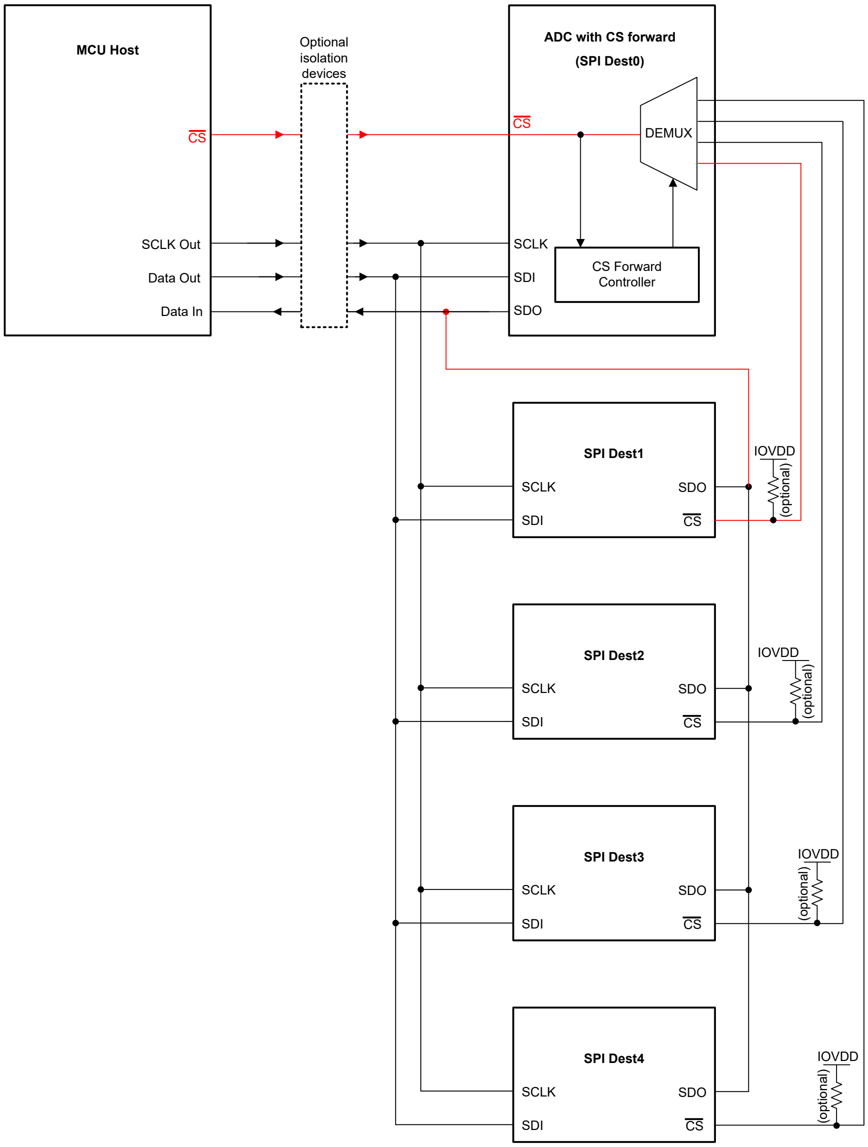


Figure 7-35. CS Forward Block Diagram

7.3.19.1 Configuring the CS forward feature

Write 010111b to the CS_FWD_EN_CODE[5:0] register to enable the CS-FWD function. Set the GPIOn_FWD_EN bits (n = 0 to 3) and AGPIOn_FWD_EN bits (n = 0 to 7) to configure each corresponding GPIO pin and AGPIO pin as a $\overline{\text{CS}}$ output on the ADS125P08. Note that for each pin having the (A)GPIOn_FWD_EN bit set, the corresponding (A)GPIOn_CFG bits must be programmed to digital output functionality (bit setting 10b).

Table 7-40 shows an overview of the register bits used to configure the CS-FWD feature.

Configure the TIMEOUT_SEL[1:0] bits as described in the *CS Forward Timeout* section.

Table 7-40. CS Forward Configuration Bits

REGISTER BITS	DESCRIPTION
CS_FWD_EN_CODE[5:0]	CS Forward feature enable. Write 010111b to enable the CS Forward feature.
TIMEOUT_SEL[1:0]	Timeout enable and duration selection.
GPIOn_FWD_EN	Enable bit to configure GPIO pin as a CS forward output
GPIOn_CFG	GPIO configuration bits - set to 10b (digital output) if pin is used for CS-FWD
AGPIOn_FWD_EN	Enable bit to configure AGPIO pin as a CS forward output
AGPIOn_CFG	AGPIO configuration bits - set to 10b (digital output) if pin is used for CS-FWD

The digital output mode of the (A)GPIO pin can be used to keep the CS input of any destination device high when the CS-FWD mode is not active: Set the corresponding (A)GPIO data output to 1b using the STEPx_(A)GPIOn_DAT_OUT bit of the step configuration page which is currently pointed to by STEP_INIT.

In summary, a recommended sequence for starting operation of the CS-FWD mode is as follows (n is the index of all GPIOs which is used for CS-FWD):

- Set (A)GPIOn output state high on the step page x pointed to by STEP_INIT, by setting STEPx_(A)GPIOn_DAT_OUT to 1b.
- Configure (A)GPIOn as digital output using the (A)GPIOn_CFG bits, setting 10b.
- Select (A)GPIO to be CS-FWD using the (A)GPIOn_FWD_EN bits.
- Send the CS-FWD enable register (010111b) and set the timeout.
- Send first header frame to the CS-FWD controller.

7.3.19.2 CS Forward Timeout

The ADS125P08 offers a timeout feature. Use the TIMEOUT_SEL[1:0] bits to enable the timeout feature and select the timeout duration.

Table 7-41 shows the options available for the timeout duration.

Table 7-41. Timeout enable and duration options

TIMEOUT_SEL[1:0]	DESCRIPTION
00b	Timeout disable
01b	Timeout enable with the short timeout, 256 CLKIN cycles
10b	Timeout enable with the medium length timeout, 2048 CLKIN cycles
11b	Timeout enable with the long timeout, 16384 CLKIN cycles

When enabled, the timeout checks that a rising edge of $\overline{CS}$ happens within the selected number of CLKIN cycles after a $\overline{CS}$ falling edge. If no rising edge is detected, timeout happens. When a timeout occurs, the rest of the SPI frame on SDI is ignored before the rising edge of $\overline{CS}$. A new SPI transaction starts at the next $\overline{CS}$ falling edge, see Figure 7-36.

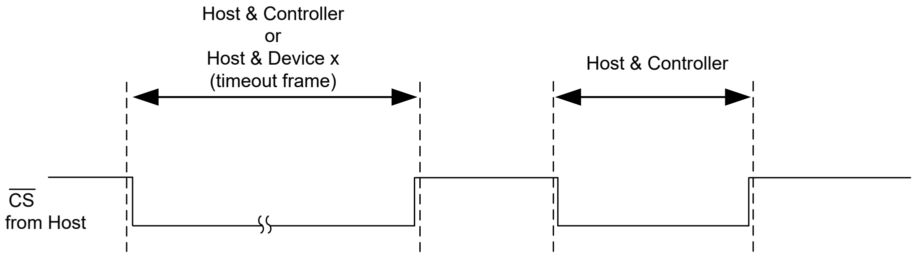


Figure 7-36. Timeout Diagram

The timeout feature improves the communication error tolerance: without a timeout capability, the communication can be lost if there are communication errors. Note that the timeout feature is only available in the CS-FWD mode.

7.3.19.3 CS Forward Header, Frame, and State Diagram

After the CS-FWD mode is enabled, the UDP communication is started by sending the CS Forward Header frame as shown in Table 7-42. The first byte on SDI includes the CS forward index or SPI index as 4 MSBs. This is the destination device selection, 0h00 to 0h0B where 0h00 is destination device 1, 0h01 is destination device 2 and so on. The first byte on SDI also includes the number of frames to be transferred as 4LSBs. The second byte on SDI includes the CRC for the preceding data, if CRC is enabled. The third byte (do not care) is always required: When in CS-FWD mode, three bytes per frame are always required for the header frame to communicate with the CS-FWD controller.

Table 7-42. CS Forward Header Frame

SIGNAL	1 st BYTE	2 nd BYTE	3 rd BYTE
SDI	SPI index (4 MSBs) and frame# bits (4 LSBs)	CRC (if CRC enabled)	Don’t care
SDO	FFh	**1 st byte** received on SDI	CRC check result (if CRC enabled)

The SPI index (4 MSBs) of the first byte in the header indicates which destination device is selected, Table 7-43 describes the exact mapping of these bits to a specific destination device.

Table 7-43. SPI Index Selection

SPI INDEX	DESTINATION DEVICE SELECTED
0000b	The device, ADS125P08
0001b	Destination device 1, connected to GPIO0
0010b	Destination device 2, connected to GPIO1
0011b	Destination device 3, connected to GPIO2
0100b	Destination device 4, connected to GPIO3
0101b	Destination device 5, connected to AGPIO0
0110b	Destination device 6, connected to AGPIO1
0111b	Destination device 7, connected to AGPIO2
1000b	Destination device 8, connected to AGPIO3
1001b	Destination device 9, connected to AGPIO4
1010b	Destination device 10, connected to AGPIO5
1011b	Destination device 11, connected to AGPIO6
1100b	Destination device 12, connected to AGPIO7
all other codes	Reserved

The frame number bits (4LSBs) of the first byte in the header in Table 7-42 sent from the host indicate the number of frames sent to the selected destination device as shown in Table 7-44. After the specified number of frames are sent to the selected destination device, the CS-FWD controller expects another header frame with new information regarding the next destination device selection and number of frames to be sent. The CS-FWD controller stays in CS-FWD mode until the user explicitly chooses to exit this mode, see the *Disabling the CS-FWD mode* section for details on exiting the mode.

Table 7-44. CS-FWD Frame Number Indication

FRAME NUMBER LSBS	NUMBER OF CS-FWD FRAMES SENT
0000b	1
0001b	2
0010b	3
0011b	4
...	...
1110b	15
1111b	16

During an UDP sequence, a header response frame is sent from the device on SDO as shown in Table 7-42, this response is denoted as Ack in the following, as shown in Figure 7-37. This consists of bits[23:16] = FFh to represent the device writing to the host, bits[15:8] repeat the data sent to initiate the UDP (first byte on SDI), and Bits[7:0] are a correct CRC for the preceding data using the polynomial x^8 $+x^2$ + x + 1. This byte of CRC is required only when CRC is enabled for the SPI, otherwise the byte is 0hFF.

The host can distinguish whether the host is communicating with the CS forward controller or any other destination device based on the SDO received. If talking to the controller, the host receives the SDO as FFh and the mirrored bytes received by the controller. So the host can check if the bytes received by the controller are correct or not and then decide how to respond: keep timeout and re-send the commands to the controller if the bytes are incorrect.

Figure 7-37 shows a timing diagram for a typical CS-FWD communication sequence. In the first header frame shown, the SPI index is set to target device M. Therefore, in the subsequent frames, the $\overline{\text{CS}}$ signal is only forwarded to device M (j is the number of frames for communication to device M as specified in the header frame). In the 2nd header frame, device N is specified and communication to device N is performed in subsequent frames (k is the number of frames for communication to device N).

Figure 7-37 also indicates the responses sent from the target device to SDO. Simultaneously with each header frame, the Ack frame of the target device is visible on SDO. Thereafter, the target device sends command or data bytes to SDO depending on the commands used, for simplicity those are indicated by Ack(M) for device M and Ack(N) for device N in Figure 7-37.

The condition for a qualified forward frame is to have a minimum of at least 8 SCLK cycles when CSn is low. However, the frames can be longer than 8 SCLK cycles, and subsequent frames can differ in length. There is no maximum limit on the frame length.

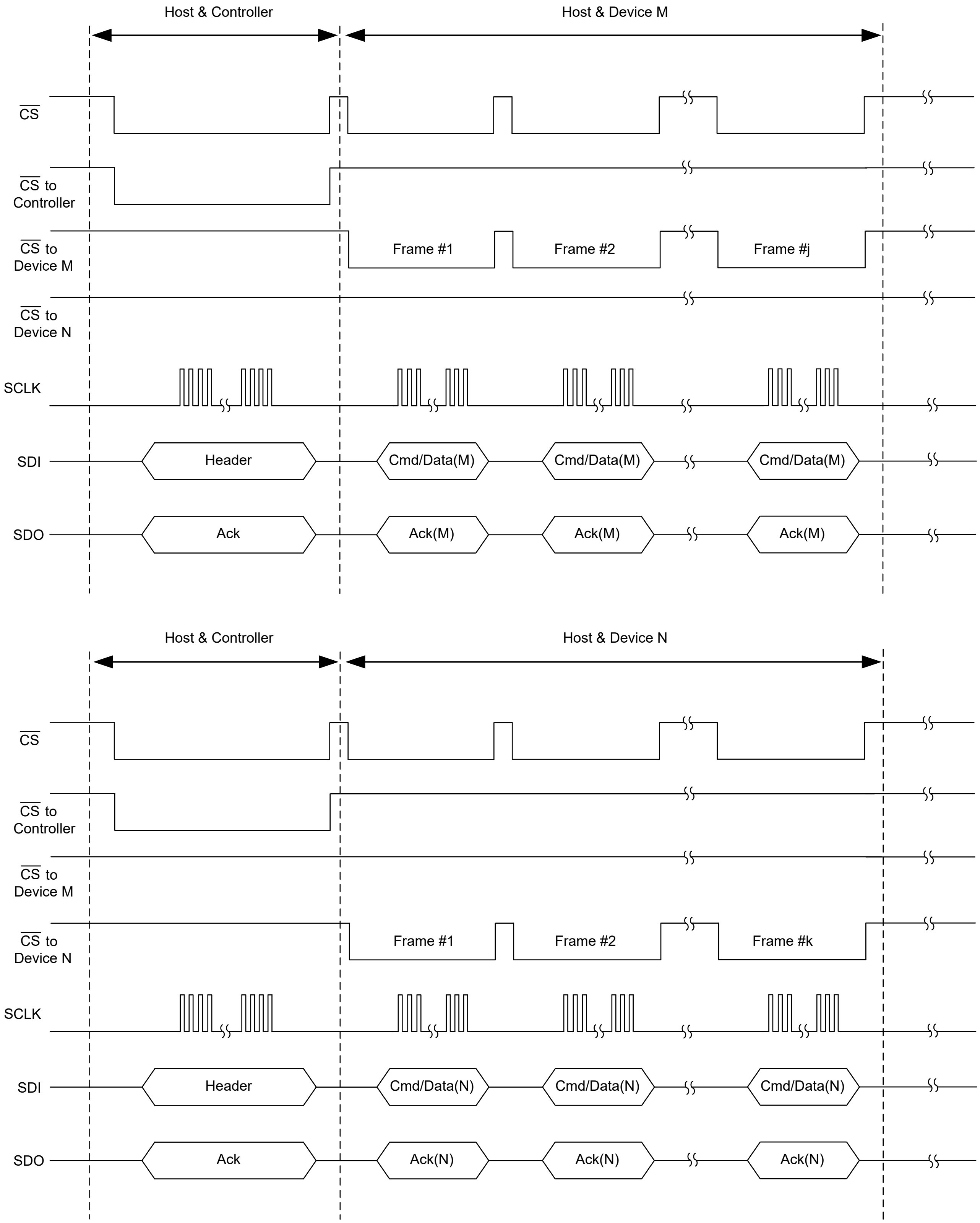


Figure 7-37. CS Forward Timing Diagram

The following error handling is recommended to provide robustness of the communication with and without CRC enabled:

- CRC is enabled: If the host detects a CRC error when receiving data from a device, the host must repeat the command to the controller until the correct CRC is received by the host. Similarly, if the CRC error is detected by the SPI controller device in the header, the controller must keep receiving the header from the host until the CRC check is correct.
- CRC is disabled: If the 2nd byte on SDO does not match the 1st byte sent to the device, the host must cause the timeout without toggling SCLK in the immediate next frame.

Figure 7-38 depicts the state diagram for the CS-FWD controller. As soon as the CS forward is enabled, the controller is expecting to receive a valid header frame. Once a valid header frame is received, the communication with the selected destination device is started, and continues until the pre-defined number of frames are completed, or a timeout occurs, or the CS forward mode is disabled in the SPI controller device.

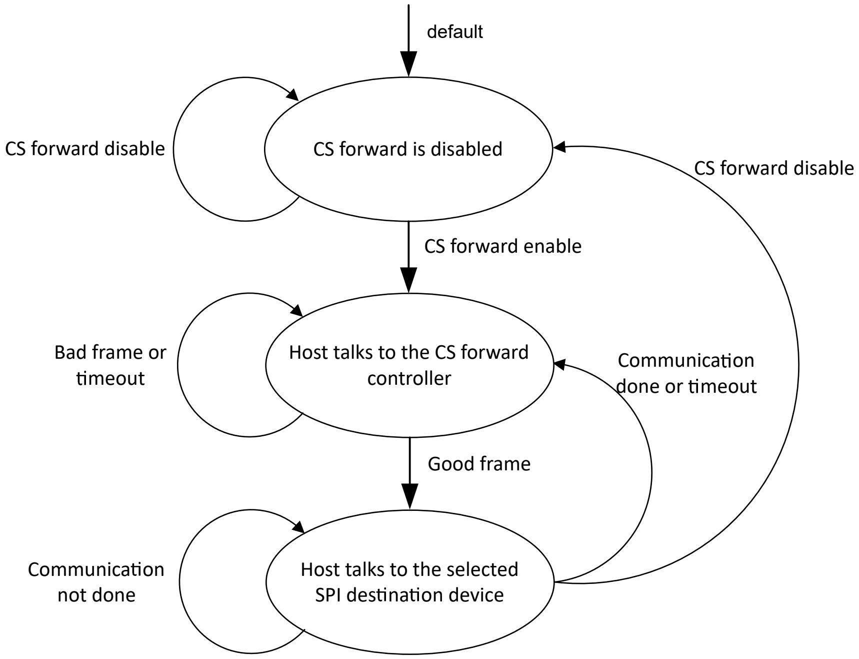


Figure 7-38. CS Forward State Diagram

7.3.19.4 Disabling the CS-FWD mode

Write 000000b to the CS_FWD_EN_CODE[5:0] register to disable the CS-FWD controller and end the CS_FWD mode operation. This SPI write operation has to be sent in a 3-byte CS-FWD frame – see the *CS Forward Header, Frame, and State Diagram* section – with the destination device 0 selected (the ADS125P08 device). Once the CS-FWD is deactivated, the GPIO pins automatically revert to the generic GPIO functionality, and the GPIOx_FWD_EN bits (x = 0 to 3) and AGPIOx_FWD_EN bits (x = 0 to 7) are ignored. After this, the SPI communication to the ADS125P08 is back to the normal SPI mode, and if a GPIO bit is set to a GPIO output mode, the data for the output is determined by the active step configuration page again as expected in a regular GPIO output mode.

Another option to end the CS-FWD mode operation is to toggle the $\overline{\text{RESET}}$ pin. Alternatively, sending the reset pattern to the device deactivates the CS_FWD mode. However note that the reset pattern needs to be sent in a frame where the host is communicating with the destination device 0 (the ADS125P08 device) by using the SPI index 0000b as shown in Table 7-43. Using the $\overline{\text{RESET}}$ pin or the reset pattern to exit CS-FWD mode is less preferred as these actions reset all user registers of ADS125P08.

7.4 Device Functional Modes

7.4.1 Power-Scalable Speed Modes

The ADC offers four speed modes with corresponding clock signal frequencies. Mode selection is based on the desired data rate, resolution, and device power consumption. The highest speed mode offers the maximum data rate and signal bandwidth, and the lowest speed mode minimizes power consumption for applications not requiring large signal bandwidths. Do not exceed the specified value of ADC clock frequency of any speed mode. See the *Clock Operation* section for the clock frequencies and clock divider options. The speed mode is programmed by the SPEED_MODE[1:0] bits.

7.4.2 Sequencer Functional Modes

ADS125P08 offers multiple functional modes controlled by the sequencer, the *Sequencer Modes* section for details.

7.4.3 Idle Mode and Standby Mode

When conversions are stopped by the user, program the ADC to engage idle mode or low-power standby mode. Idle mode (default) or standby mode is programmed by the STBY_MODE bit.

In idle mode, the analog circuit is fully biased and operational, including sampling of the signal and voltage reference inputs. Only the digital filter is idle. When conversions are started, the digital filter is enabled to begin the conversion process.

In standby mode, sampling of the signal and reference voltage stop when conversions stop to conserve power. When conversions are restarted, sampling of the signal and reference voltages resume. Exiting standby mode requires 24 clock cycles added to the normal conversion latency time.

7.4.4 Power-Down Mode

Power-down mode is engaged by setting the PWDN bit. In power-down mode, the analog and digital sections are powered off, except for a small bias current required to maintain SPI operation needed to exit power-down mode by clearing the register bit. The digital LDO also remains active to maintain user register settings. The sampling of the signal and voltage reference are stopped during power-down mode. Exit power-down mode by writing 0b to the PWDN bit or by resetting the device.

Setting the PWDN bit stops conversions immediately, and stops an ongoing sequencer run. Register settings are maintained except the analog GPIO settings which are reset to default values. Upon exit from power-down mode, the sequencer continues with the next ADC conversion as defined by the sequence run.

7.4.5 Reset

The ADC performs an automatic reset at power-on. Manual reset is through the RESET pin or through SPI operation. The control logic, digital filter, SPI and data port operation, and user registers reset to the default values. See Figure 5-3 for details when the ADC is available for operation after reset.

7.4.5.1 $\overline{\text{RESET}}$ Pin

The $\overline{\text{RESET}}$ pin is an active low input. The ADC is reset by taking $\overline{\text{RESET}}$ low then back high. Because the $\overline{\text{RESET}}$ pin has an internal 20kΩ pullup resistor, $\overline{\text{RESET}}$ can be left unconnected if not used. The $\overline{\text{RESET}}$ pin is a Schmitt-triggered input designed to reduce noise sensitivity. See Figure 5-3 for $\overline{\text{RESET}}$ pin timing and for the start of SPI communications after reset. Because the ADC performs an automatic reset at power-on, a reset is not required after device power-on.

7.4.5.2 Reset by SPI Register Write

The device is reset through SPI operation by writing 01011010b to the RESET_CODE[7:0] bits. Writing any other value to this bit field does not result in reset. In 4-wire SPI mode, reset takes effect at the end of the frame at the time $\overline{\text{CS}}$ is taken high. In 3-wire SPI mode, reset takes effect on the last falling edge of SCLK of the register write operation. Reset in 3-wire SPI mode requires that the SPI is synchronized to the SPI host. If SPI synchronization is not assured, use the pattern described in the *Reset by SPI Input Pattern* section to reset the device. Reset can be validated by checking the RESETn flag of the STATUS_MSB register.

7.4.5.3 Reset by SPI Input Pattern

The device is also reset through SPI operation by inputting a special bit pattern. The input pattern does not follow the input command format. There are two input patterns in which to reset the ADC. Pattern 1 consists of a *minimum* 1023 consecutive ones followed by one zero. The device resets on the falling edge of SCLK when the final zero is shifted in. This pattern can be used for either 3- or 4-wire SPI modes. Figure 7-39 shows a pattern 1 reset example.

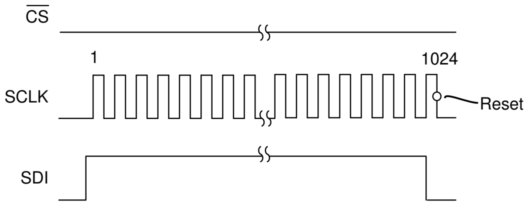


Figure 7-39. Reset Pattern 1 (3-Wire or 4-Wire SPI Mode)

Reset pattern 2 is only for use with the 4-wire SPI mode. To reset, input a *minimum* of 1024 consecutive ones (no ending zero value), followed by taking $\overline{CS}$ high at which time reset occurs. Use pattern 2 when the devices are connected in daisy-chain mode. Figure 7-40 shows a pattern 2 reset example.

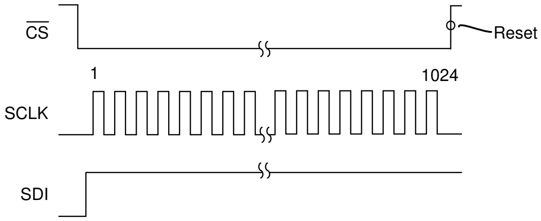


Figure 7-40. Reset Pattern 2 (4-Wire SPI Mode)

7.4.6 Synchronization

Conversions are synchronized and controlled by the START pin or, optionally, through SPI operation. If controlling conversions through SPI operation, keep the START pin low to avoid contention with the pin.

A conversion starts when writing 1b to the START bit. Likewise, a conversion stops when writing a 1b to the STOP bit. The stop behavior is configurable using the STOP_BEHAVIOR[1:0] bits. See the *Starting and Stopping the Sequencer* section for details on the stop behavior. Writing a 1b to the START and STOP bits at the same time has no effect on conversions.

The Start bit clears after a sequence begins so the bit reads back 0b. The Stop bit clears after the sequencer stops. The user can read back 1b if the sequencer hasn't completed a sequence after writing Stop bit.

Configure the GPIO0/START pin to START pin operation using the GPIO0_CFG[1:0] bits. A rising edge on the START pin is equivalent to writing to the START bit. A falling edge on the START pin is equivalent to writing to the STOP bit. The step page is determined by the STEP_INIT[4:0] bits, just as with the START and STOP bits. If the START pin is high at power-up, a start operation begins after the power-up cycle completes. See the *Starting and Stopping the Sequencer* section for details.

If a start is issued during a conversion, the conversion stops and restarts. If a start is issued while a stop request is pending (per STOP_BEHAVIOR[1:0]), the stop request is cleared and a new start initiates.

When a start event occurs, either from the START pin or writing the START bit, a new conversion begins as soon as the modulator is ready. The wait time depends on whether the ADC is in idle or standby mode, and the data rate, the filter mode and are a combination of internal and user-set values.

After the ADC is synchronized, the first conversion is fully settled data but incurs a delay (latency time) compared to the normal data period. This latency is needed to account for full settling of the digital filter. The latency time depends on the data rate and the filter mode (see the *Digital Filter* section for filter latency details).

7.4.7 Conversion-Start Delay Time

A programmable delay time is provided to delay the start of the conversion cycle after the START pin or START bit is asserted, and at the beginning of each sequence step. This delay time allows for settling of external components, such as the voltage reference after exiting standby mode, or for additional settling time when switching the signal through the multiplexer. After the initial delay time, subsequent conversions are not delayed. The programmed value of this delay time adds to the value of the conversion latency time of the digital filter. Use the STEPx_DELAY_MSB[7:0] and STEPx_DELAY_LSB[7:0] bits (x = 0 to 31) to configure the delay time per sequence step.

7.5 Programming

7.5.1 Serial Interface (SPI)

The serial interface is used to read conversion data, configure device registers, and control ADC conversions.

The serial interface consists of four lines: $\overline{\text{CS}}$, SCLK, SDI, and SDO/$\overline{\text{DRDY}}$. A dedicated $\overline{\text{DRDY}}$ pin is also available. The interface operates in the peripheral mode (passive) where SCLK is driven by the host. The interface is compatible to SPI mode 1 (CPOL = 0 and CPHA = 1). In SPI mode 1, SCLK idles low, and data update on SCLK rising edges and are latched on SCLK falling edges.

The interface supports full-duplex operation, meaning input data and output data can be transmitted simultaneously. The interface also supports daisy-chain connection of multiple ADCs to simplify the SPI connection.

7.5.2 Serial Interface Signals

7.5.2.1 Chip Select ($\overline{\text{CS}}$)

$\overline{\text{CS}}$ is an active-low input that enables the interface for communication. A communication frame starts by taking $\overline{\text{CS}}$ low and ends by taking $\overline{\text{CS}}$ high. When $\overline{\text{CS}}$ is taken high, the device ends the frame by interpreting the last 16 bits of input data (24 bits in CRC mode) regardless of the total number of bits shifted in. When $\overline{\text{CS}}$ is high, the SPI resets, commands are blocked, and SDO/$\overline{\text{DRDY}}$ enters a high-impedance state. The dedicated $\overline{\text{DRDY}}$ pin is an active output regardless of the state of $\overline{\text{CS}}$. $\overline{\text{CS}}$ can be tied low to operate the interface in 3-wire SPI mode.

7.5.2.2 Serial Clock (SCLK)

SCLK is the serial clock input used to shift data into and out of the ADC. Output data update on the rising edge of SCLK and input data are latched on the falling edge of SCLK. SCLK is a Schmitt-triggered input designed to increase noise immunity. Even though SCLK is noise resistant, keep SCLK as noise-free as possible to avoid unintentional SCLK transitions. Avoid ringing and overshoot on the SCLK input. A series termination resistor at the SCLK driver can reduce ringing.

7.5.2.3 Serial Data Input (SDI)

SDI is the serial interface data input. SDI is used to input data to the device. Input data are latched on the falling edge of SCLK.

7.5.2.4 Serial Data Output/Data Ready (SDO/$\overline{\text{DRDY}}$)

SDO/$\overline{\text{DRDY}}$ is a dual-function output pin. This pin is programmable to provide output data only, or to provide output data and the data-ready indication. The SDO_MODE bit of the CLK_DIGITAL_CFG register programs the mode. The dual-function mode multiplexes output data and data-ready operations on a single pin. This mode can replace the function of the dedicated $\overline{\text{DRDY}}$ pin to reduce the number of SPI I/O lines required to interface to the host.

Output data update on the rising edge of SCLK. The SDO/$\overline{\text{DRDY}}$ pin is in a high-impedance state when $\overline{\text{CS}}$ is high.

When programmed to dual-function mode (SDO_MODE bit = 1b) and when $\overline{\text{CS}}$ is low, SDO/$\overline{\text{DRDY}}$ mirrors $\overline{\text{DRDY}}$ until the first rising edge of SCLK, at which time the pin changes mode to provide data output. When the data read operation is complete (24th falling edge of SCLK, or 48th edge if the CRC byte and STATUS header are included), the pin reverts back to mirroring $\overline{\text{DRDY}}$. Figure 7-41 illustrates the operation of SDO/$\overline{\text{DRDY}}$.

When using the dual-function mode of ADS125P08, the device switches from $\overline{\text{DRDY}}$ to SDO mode at the first rising SCLK. This transition takes $t_{p(DRDO)}$ which must be accounted for either by lengthening the first SCLK high-period, lengthening all SCLK high-periods, or latching output data after SCLK falling edge.

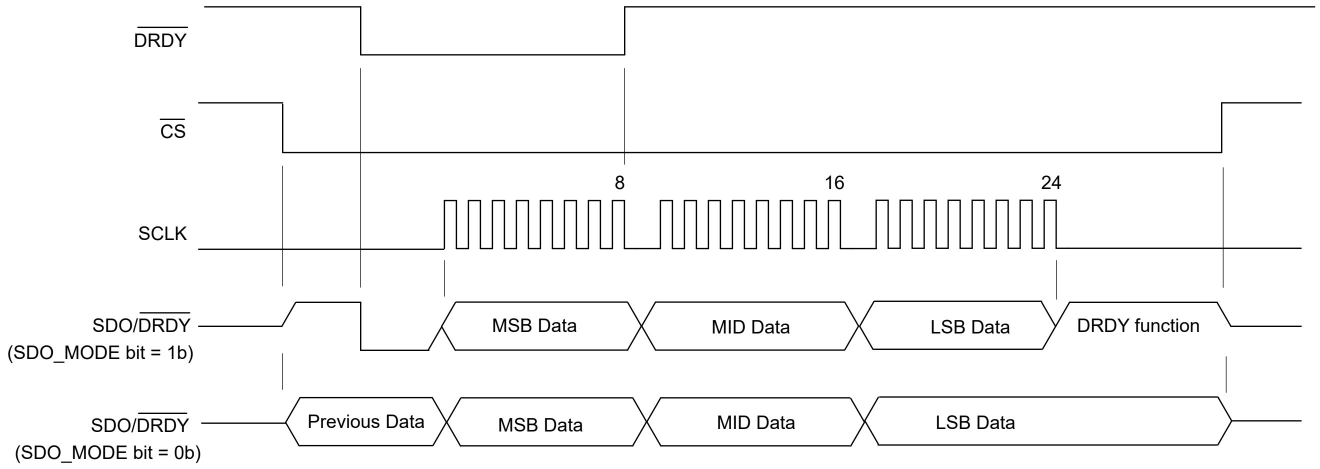


Figure 7-41. SDO/DRDY and DRDY Function

For the output data only mode SDO_MODE = 0b, SDO stays at the level of the last bit sent if the host does not send any extra SCLK pulses after the last data is shifted out on SDO, as shown in Figure 7-41. If the host sends additional SCLK pulses after the last data is shifted out, then SDO drives low. Figure 7-42 shows a timing diagram of the SDO behavior with additional SCLK pulses.

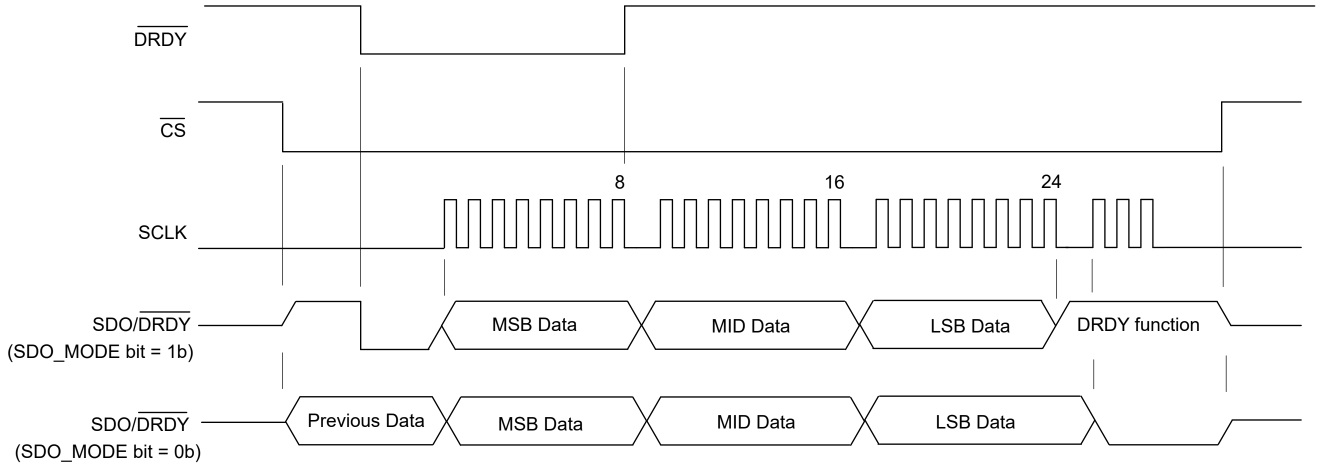


Figure 7-42. SDO/DRDY and DRDY Function With Additional SCLK Pulses

7.5.2.5 Data Ready (DRDY) Pin

DRDY is the data-ready output signal pin. On the ADS125P08, this pin is a dual-function output pin denoted DRDY/GPIO1 on the *Functional Block Diagram* and in the *Pin Configuration and Functions* section. This pin is programmable to operate as a general purpose input/output, or to provide the data-ready indication. By default, the pin operates as DRDY signal. To simplify notation, the pin is referred to as DRDY pin instead of DRDY/GPIO1 pin for the remainder of this document.

DRDY drives high when conversions are started or resynchronized, and drives low when conversion data are ready. DRDY is driven back high at the eighth falling edge of SCLK during conversion data read as shown in Figure 7-41. If conversion data are not read, DRDY pulses high just prior to the next falling edge. Whenever the ADC is programmed to enter standby mode (STBY_MODE bit = 1b), DRDY is driven back high four f_{CLK} cycles after transitioning low. DRDY is an active output whether CS is high or low.

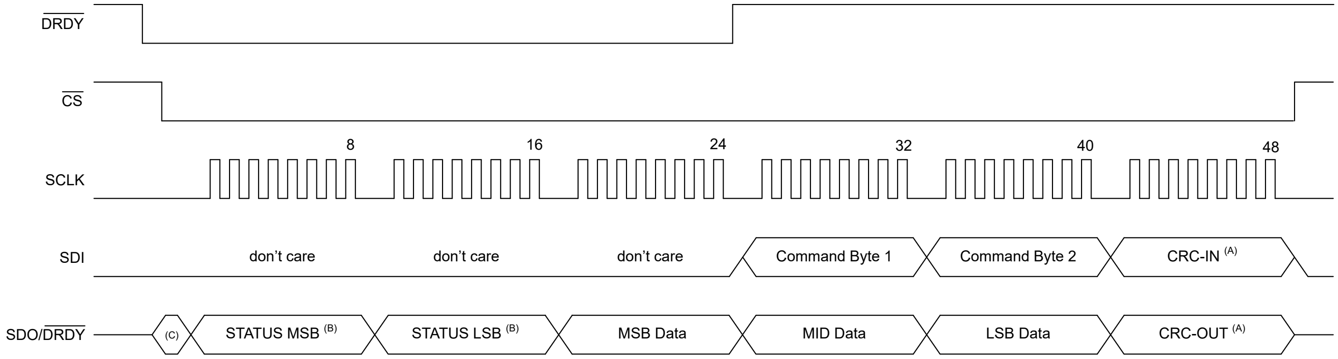
See the *DRDY Pin Behavior* section for further details on the DRDY operation.

7.5.3 Serial Interface Communication Structure

7.5.3.1 SPI Frame

Communication through the serial interface is based on the concept of frames. A frame consists of a prescribed number of SCLKs required to shift in or shift out data. A frame starts by taking $\overline{\text{CS}}$ low and ends by taking $\overline{\text{CS}}$ high. When $\overline{\text{CS}}$ is taken high, the device interprets the last 16 bits (or 24 bits in CRC mode) of input data regardless of the amount of data shifted into the device.

The interface is full duplex, meaning that the interface is capable of transmitting data on SDO while simultaneously receiving data on SDI. Typically, the input frame is sized to match the output frame by padding the frame with leading zeros if needed. However, if not transmitting and receiving data in full-duplex mode, the input data frame can be the minimum size of 16 bits (or 24 bits in CRC mode). Figure 7-43 shows a typical communication frame structure. In this example, conversion data are shifted out on the SDO pin.



A. Optional CRC byte. If the CRC is disabled, the frame shortens by one byte.
B. Optional STATUS header. If STATUS is disabled, the frame shortens by two bytes.
C. If the SDO_MODE bit = 0b, the previous state of SDO/$\overline{\text{DRDY}}$ remains until the first SCLK rising edge. Otherwise, SDO/$\overline{\text{DRDY}}$ follows $\overline{\text{DRDY}}$.

Figure 7-43. Typical Communication Frame

The output frame size, as given in Table 7-45, depends on the optional STATUS header (which is 2 bytes) and CRC byte. After the ADC is powered up or reset, the default output frame size is 24 bits. In 3-wire SPI mode, the input frame must match the size of the output frame for the SPI to remain synchronized.

Table 7-45. Output Frame Size

RESOLUTION	STATUS HEADER	CRC BYTE	FRAME SIZE
24 bit	No	No	24 bit
24 bit	No	Yes	32 bit
24 bit	Yes	No	40 bit
24 bit	Yes	Yes	48 bit

A continuous-read mode is available, where an arbitrary number of register data or FIFO data can be retrieved without any transitions of $\overline{\text{CS}}$, and the frame extends to accommodate the additional data. See the *Continuous Read Mode* section for details. In continuous-read mode, the output frame size is unlimited.

7.5.3.2 STATUS Header

The ADS125P08 outputs an optional STATUS header as the first two bytes in every frame on SDO. The STATUS header is enabled by setting the STATUS_EN bit of the DIAG_MONITOR_CFG register. The 16 bit STATUS header is a concatenation of the STATUS_MSB[7:0] and STATUS_LSB[7:0] register bits. Fault flags, status flags, the conversion counter, and the sequence step indicator, are all part of these bits. See the respective register bit descriptions in the *Register Map* section for details.

- Communication-related fault flags, such as the SPI_CRC_FAULTn and REG_WRITE_FAULTn flags, always indicate faults that occurred in the previous SPI frame. These fault flags clear automatically for the next SPI frame.
- All other device-related fault and status flags indicate the state of the device at the instance of the start of the current SPI frame.
- Some flags represent logic-or combinations of flags from other status registers, such as the ADC_REF_FAULTn, FIFO_FAULTn and INTERNAL_FAULTn flags. Those bits update when any bit of the respective lower level status registers is set. For those bits to clear, all bits in the lower-level status registers must be cleared.
- All fault flags, except for the communication related fault flags mentioned above, are latching. That means, these fault flags do not reset automatically to 1b when the fault condition is removed and must be cleared by the host.
- The conversion and sequence step counters indicate the count of the data which is output in the current SPI frame: The conversion counter increments with every new conversion, and the sequence step counter indicates which sequence step is currently executed.

7.5.3.3 SPI CRC

The SPI cyclic redundancy check (CRC) is a check code used to detect transmission errors to and from the host controller. A CRC-IN byte is transmitted with the ADC input data by the host on SDI and a CRC-OUT byte is transmitted with the output data by the device on SDO. Use the SPI_CRC_EN bit to enable the SPI CRC. In addition, enable the transmission of the STATUS header using the STATUS_EN bit to get notified about any SPI input CRC faults.

The CRC-IN code is calculated by the host over the two command bytes. Any input bytes padded to the start of the frame are not included in the CRC-IN calculation. The ADC checks the input command CRC-IN code against an internal code calculated over the two received input command bytes. If the CRC-IN codes do not match, the command is not executed and the SPI_CRC_FAULTn bit is set to 0b in the STATUS_LSB byte.

The SPI_CRC_FAULTn bit is output as part of the STATUS header to provide immediate indication that a CRC error occurred in the previous frame. The SPI_CRC_FAULTn bit clears automatically in the next SPI frame, assuming there is no SPI CRC error in the current frame.

The number of bytes used to calculate the output CRC code depends on the amount of data bytes transmitted in the frame on SDO. Table 7-46 shows the number of bytes used for the output CRC calculation.

Table 7-46. Data Covered by Output CRC

ACTION	STATUS HEADER ENABLED	BYTE COUNT	BIT COUNT AND DESCRIPTION
Conversion data read	No	3	24 bits of conversion data
Register data read	No	3	8 bits of register data + 8 bits address byte + 8 bits of 00h padding
Conversion data read	Yes	5	16 bits STATUS header + 24 bits of conversion data
Register data read	Yes	5	16 bits STATUS header + 8 bits of register data + 8 bits address word + 8 bits of 00h padding

The CRC code calculation is the 8-bit remainder of the bitwise exclusive-OR (XOR) operation of the variable length argument with the CRC polynomial. The CRC is based on the CRC-8-ATM (HEC) polynomial: $X^8 + X^2 + X^1 + 1$. The nine coefficients of the polynomial are: 100000111. The CRC calculation is initialized to all 1s to detect errors in the event that SDI and SDO/$\overline{\text{DRDY}}$ are either stuck high or low.

Figure 7-44 shows a visual representation of the CRC calculation. The following procedure calculates the CRC value:

- Preload the 8-bit shift register, which has XOR blocks located at positions that correspond to the CRC polynomial (07h), with the seed value of FFh.
- Shift in all data bits starting with the most-significant bit (MSB) and re-compute the shift-register value after each bit.

- The resulting shift-register value after all data bits have been shifted in is the computed CRC value.

The example C code available for download *here* includes a potential CRC implementation.

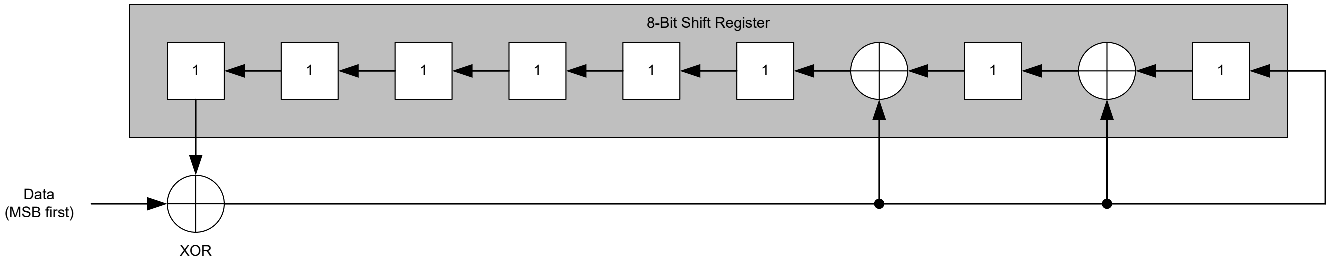


Figure 7-44. Visual Representation of CRC Calculation

7.5.4 Device Commands

Commands are used to read and write register data, or to read from the FIFO buffer. The *Register Map* consists of a series of one-byte (8 bit) registers, accessible by read and write operations. The minimum frame length of the input command sequence is 16 bit (24 bit in CRC mode). If desired, the input command sequence can be padded with leading zeros to match the length of the output data frame. In CRC mode, the device interprets the two bytes preceding the CRC-IN byte at the end of the frame as the command bytes. Table 7-47 shows the ADS125P08 commands.

Table 7-47. SPI Commands

DESCRIPTION	BYTE 1	BYTE 2	BYTE 3 (Optional CRC-IN Byte)
No operation (read conversion data)	00h	00h	D7h
Read register command	40h + address [5:0]	don't care	CRC-IN of byte 1 and byte 2
Write register command	80h + address [5:0]	register data	CRC -INof byte 1 and byte 2
Read FIFO buffer command	0Fh	don't care	CRC-IN of byte 1 and byte 2

The device supports special extended-length bit patterns that are longer than the standard command length. These patterns are used to reset the ADC and to reset the frame in 3-wire SPI mode. The extended bit patterns are explained in the *Reset by SPI Input Pattern* and *3-Wire SPI Mode* sections.

7.5.4.1 No-Operation

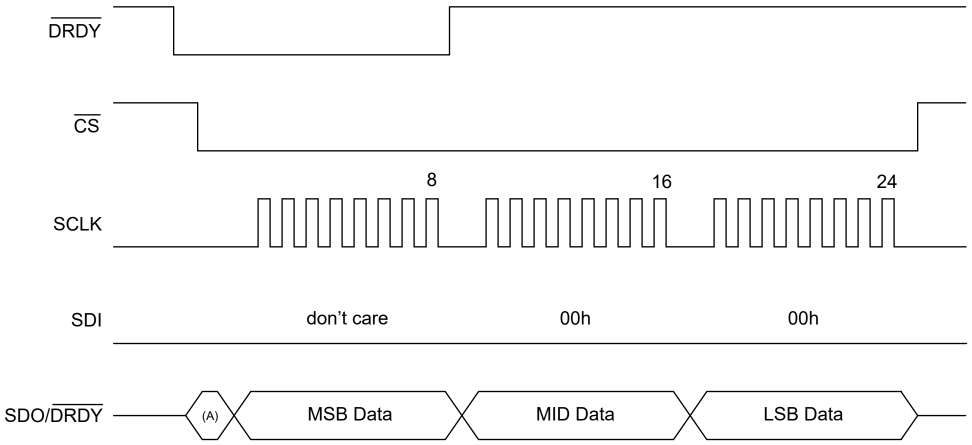
The no-operation command bytes are 00h and 00h. Use this command if no input command is desired. If the SPI CRC check is enabled, the CRC byte is required (byte 3), which is always D7h for bytes 00h and 00h. SDI can be held low during data readback, but in CRC mode the SPI_CRC_FAULTn bit of the STATUS_LSB flag is set to 0b. The SPI_CRC_FAULTn flag can be ignored while reading conversion data, and clears automatically to 1b in every new SPI frame.

7.5.4.2 Read Conversion Data

Conversion data are read by taking $\overline{CS}$ low and by applying SCLK to shift out the data directly (no command is used). Conversion data are buffered, which allows data to be read up to one f_{MOD} clock cycle before the next $\overline{DRDY}$ falling edge. Conversion data can read multiple times until the next conversion data are ready, and are never corrupted. If the register read command is sent in the previous frame then register data replace the conversion data.

$\overline{DRDY}$ is driven back high at the eighth SCLK falling edge during conversion data read (i.e. when the transmission of the conversion data MSB byte is complete).

Figure 7-45 shows an example of reading 24 bit conversion data with the STATUS and CRC bytes disabled.

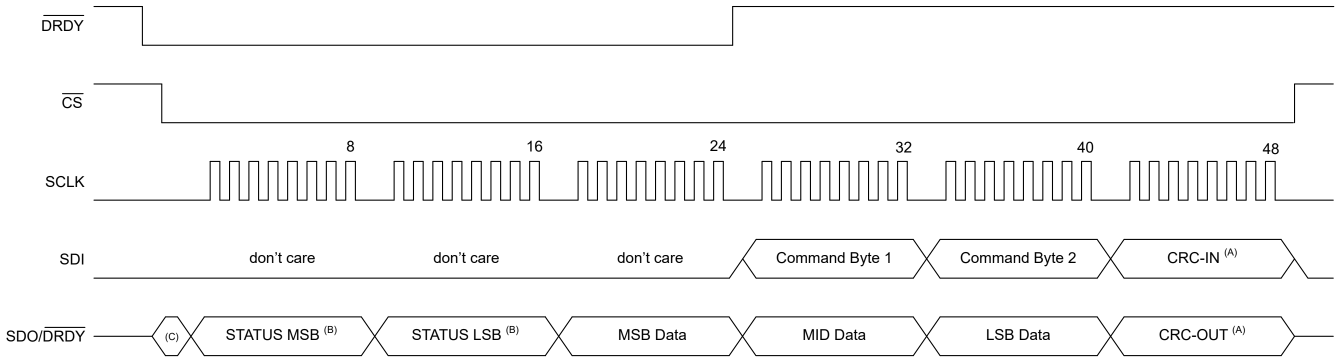


A. If the SDO_MODE bit = 0b, the previous state of SDO/$\overline{\text{DRDY}}$ remains until the first SCLK rising edge. Otherwise, SDO/$\overline{\text{DRDY}}$ follows $\overline{\text{DRDY}}$.

Figure 7-45. Read Conversion Data, 24 Bit Frame Size

Figure 7-46 is an example of the read conversion data operation when the STATUS header and the CRC byte are included. This example also shows the optional use of a full-duplex transmission when a register command is input at the same time the conversion data are output. If no input command is desired, the input bytes are 00h, 00h, and D7h. The output CRC (CRC-OUT) code computation includes the STATUS header.

$\overline{\text{DRDY}}$ is driven back high at the at the 24th SCLK falling edge, when the transmission of the conversion data MSB byte is complete. This is also true if the data is not completely read, for example, if the read operation is stopped any time after the transmission of the conversion data MSB byte, but before the end of the frame.



A. Optional CRC byte. If the CRC is disabled, the frame shortens by one byte.

B. Optional STATUS header. If STATUS is disabled, the frame shortens by two bytes.

C. If the SDO_MODE bit = 0b, the previous state of SDO/$\overline{\text{DRDY}}$ remains until the first SCLK rising edge. Otherwise, SDO/$\overline{\text{DRDY}}$ follows $\overline{\text{DRDY}}$.

Figure 7-46. Read Conversion Data, 48 Bit Frame Size

Conversion data can be read asynchronous to $\overline{\text{DRDY}}$. However, when conversion data are read close to the $\overline{\text{DRDY}}$ falling edge, there is uncertainty whether previous data or new data are output. If the SCLK shift operation starts at least one f_{MOD} clock cycle before the $\overline{\text{DRDY}}$ falling edge, then old data are provided. If the shift operation starts at least one f_{MOD} clock cycle *after* $\overline{\text{DRDY}}$, then new data are output. In either case, data is not corrupted. The DRDY bit of the STATUS_MSB header indicates if the data are old (previously read data, DRDY = 0b) or new (DRDY = 1b).

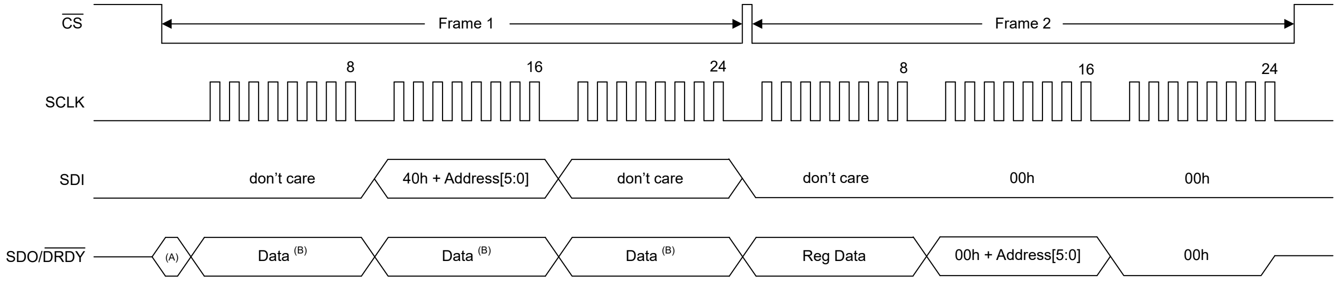
7.5.4.3 Read Register Command

The read register command is used to read register data. The command follows a two-frame protocol in which the read command is sent in one frame and the ADC responds with register data in the next frame. The first byte of the command is the base command value (40h) added to the 6-bit register address. The value of the second command byte is arbitrary, but is used together with the first byte for the CRC. The device outputs 00h as the register data for a read from an register outside the valid address range. The register data format is most-significant-bit first.

Figure 7-47 shows an example of reading register data using the 24-bit output frame size. Frame 1 is the command frame and frame 2 is the data response frame. The frames are delimited by taking $\overline{CS}$ high. The data response frame returns the requested register data byte, followed by the register address indication byte, and a 00h padded byte to complete the 24-bit frame. The 6-bit register address is right-aligned within the register address indication byte (padded with 00b at the MSB position). If desired, the data response frame can be shortened after the data byte by taking $\overline{CS}$ high.

Reading from a register address outside the valid address range returns the address FFh in the register address indication byte to indicate an error.

When reading multiple registers, full-duplex operation can be used to double the throughput of the read register operations by inputting the next read register command during the data response frame of the previous register.

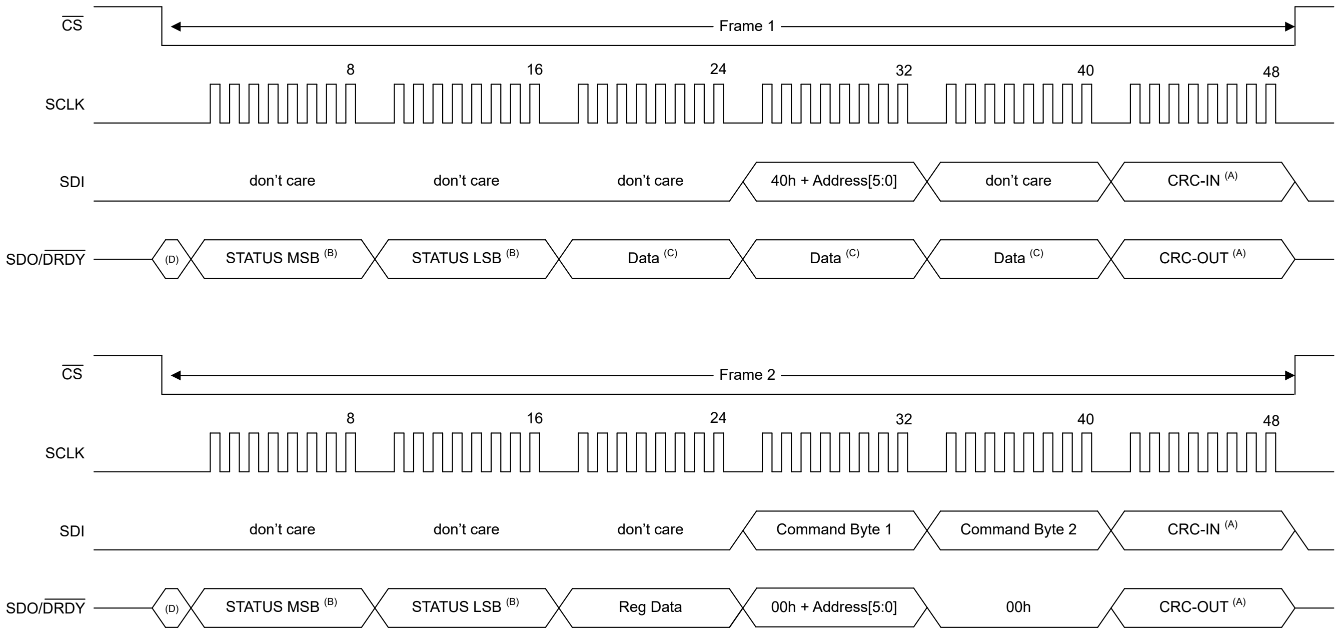


A. Previous state of SDO/$\overline{DRDY}$ before first SCLK.

B. Data are either 24 bits of conversion data, or if the read register command is sent in the prior frame, the data field is the register data byte + address indication byte + 00h padding.

Figure 7-47. Read Register Data, 24-Bit Frame Size

Figure 7-48 shows an example of the read register operation using the 48-bit frame size in full-duplex operation. In frame 1, conversion data are output at the same time as the read register command is input (if the previous frame is not a read register command). The input command is padded with three don't care bytes to match the length of the output data frame. The padded input bytes are excluded from the CRC-IN code calculation. Frame 2 shows the input of the next command concurrent with the output of the previous register data. The CRC-OUT code includes all preceding bytes within the data output frame. The SPI_CRC_FAULTn bit of the STATUS_LSB header indicates if an SPI CRC error occurred and whether the read register command is accepted.



A. Optional CRC byte. If CRC is disabled, the frame shortens by one byte.
B. Optional STATUS header. If STATUS is disabled, the frame shortens by two bytes.
C. Depending on the previous operation, the data field is either conversion data or the register data byte + address indication byte + 00h padding.
D. Previous state of SDO/$\overline{\text{DRDY}}$ before the first SCLK.

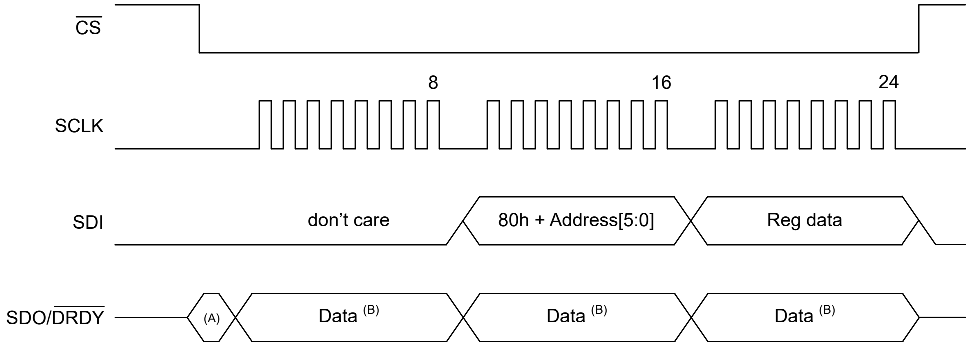
Figure 7-48. Read Register Data, 48-Bit Frame Size

7.5.4.4 Write Register Command

The write register command is used to write register data. The write register operation is performed in a single frame. The first byte of the command is the base value (80h) added to the 6-bit register address. The second byte of the command is the register data.

Writing to registers outside the valid address range is ignored and the REG_WRITE_FAULTn bit of the STATUS_LSB byte is set low to indicate an error.

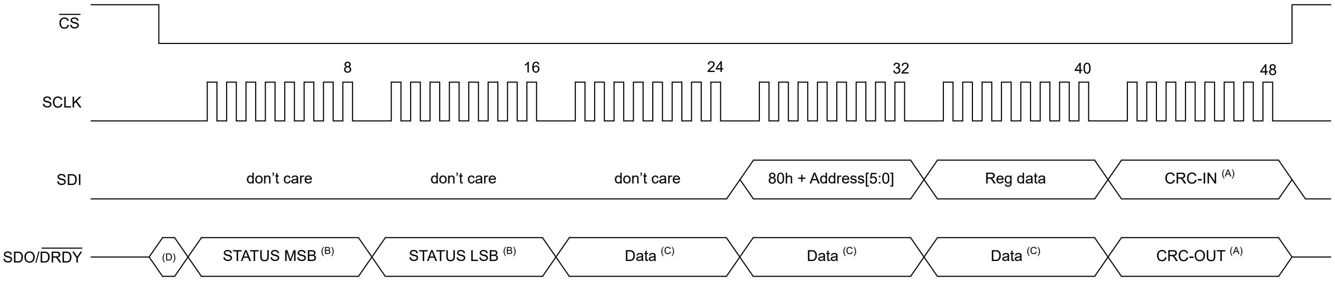
Figure 7-49 shows an example of a register write operation using the 24-bit frame size. When configuring a series of registers (when conversion data can be ignored), the minimum 16-bit frame size can be used to improve throughput.



A. Previous state of SDO/$\overline{\text{DRDY}}$ before the first SCLK.
B. Data are either the conversion data, or if the read register command is sent in a prior frame, the data field is register data byte + address indication byte + 00h padding.

Figure 7-49. Write Register Data, 24-Bit Frame Size

Figure 7-50 shows an example of a write register operation using the 48-bit frame size. Full-duplex operation is also illustrated to show simultaneous input of the command and output of conversion data. The input frame is prefixed with two *don't care* bytes to match the output frame so all conversion data bytes are transmitted. Successful write operations are verified by reading back the register data, or by checking the SPI_CRC_FAULTn bit of the STATUS_LSB byte for input byte CRC errors. If an SPI CRC input error occurred, SPI_CRC_FAULTn is set low.



A. Optional CRC byte. If CRC is disabled, the frame shortens by one byte.
B. Optional STATUS header. If STATUS is disabled, the frame shortens by two bytes.
C. The data field is either conversion data, or if the read register command is sent in the prior frame, register data byte + address indication byte + 00h padding.
D. Previous state of SDO/$\overline{DRDY}$ before the first SCLK.

Figure 7-50. Write Register Data, 48-Bit Frame Size

7.5.4.5 Read FIFO Buffer Command

The read FIFO buffer command is used to read FIFO buffer data. The command follows a two-frame protocol in which the read command is sent in one frame and the ADC responds with FIFO buffer data in the next frame. The first byte of the command is the FIFO buffer read command value (0Fh). The value of the second command byte is arbitrary, but is used together with the first byte for the CRC.

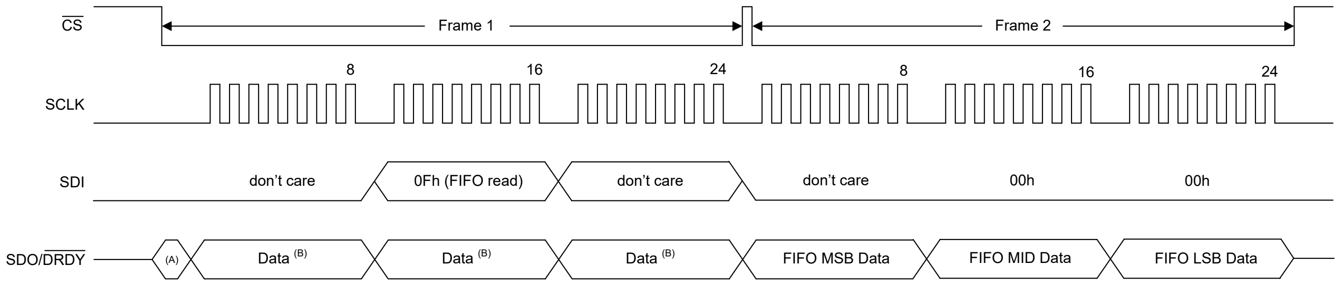
As explained in the *FIFO Buffer Read and Write* section, the FIFO buffer read pointer defines the memory location inside the FIFO for the data word to be read. The FIFO_DEPTH[8:0] bits on the Status and General Configuration Page indicate the depth of the FIFO buffer, for example, the number of conversion results stored and available for read out. See the *FIFO Buffer Read and Write* section for a detailed description of the FIFO buffer operation.

When the FIFO is disabled, each of the conversion data bytes retrieved from the FIFO read 00h.

Figure 7-51 shows an example of reading FIFO data using the 24-bit frame size. Frame 1 is the command frame and frame 2 is the data response frame. The frames are delimited by taking $\overline{CS}$ high. The data response frame returns the FIFO data.

When reading multiple FIFO buffer addresses, full-duplex operation can be used to double the throughput of the read FIFO operations by inputting the next read FIFO command during the data response frame of the previous read FIFO command. Alternatively, continuous read mode is available for reading multiple FIFO addresses efficiently. See the *Continuous Read Mode* section for details.

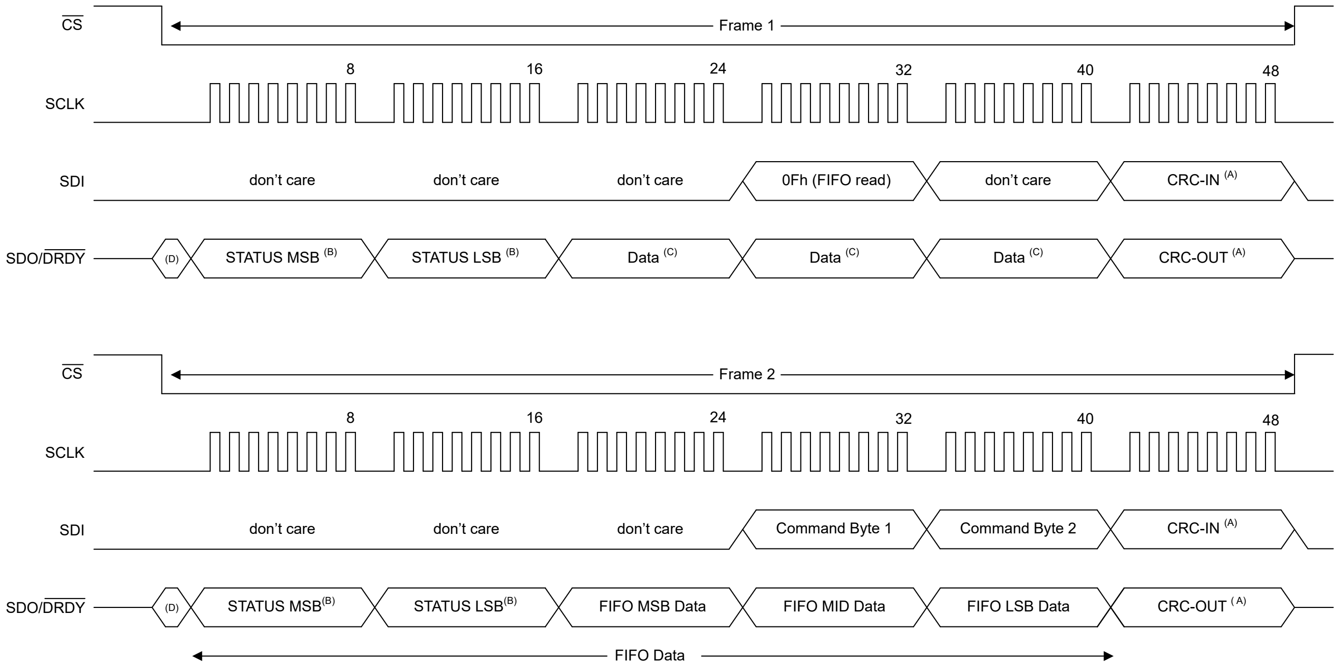
When repeatedly reading data from the FIFO, wait for the device to prepare the next data. When performing sequential Read FIFO transactions or sequential Continuous Read FIFO transactions, the transactions must be started $t_{d(FIFORD)}$ after each other.



A. Previous state of SDO/$\overline{\text{DRDY}}$ before first SCLK.
B. Data are either 24 bits of conversion data, or if the read register command is sent in the prior frame, the data field is the register data byte + address byte + 00h padding.

Figure 7-51. Read FIFO Buffer Data, 24-Bit Frame Size

Figure 7-52 shows an example of the read FIFO buffer operation using the 48-bit frame size in full-duplex operation. In frame 1, conversion data are output at the same time as the input of the read FIFO command (if the previous frame is not a read register command). The input command is padded with three don't care bytes to match the length of the output data frame. The padded input bytes are excluded from the CRC-IN code calculation. Frame 2 shows the input of the next command concurrent with the output of the FIFO data. The CRC-OUT code includes all preceding bytes within the data output frame. The SPI_CRC_FAULTn bit of the STATUS_LSB header indicates if an SPI CRC error occurred and whether the read FIFO command is accepted.



A. Optional CRC byte. If CRC is disabled, the frame shortens by one byte.
B. Optional STATUS header. If STATUS is disabled, the frame shortens by two bytes.
C. Depending on the previous operation, the data field is either conversion data or the register data byte + address byte + 00h padding.
D. Previous state of SDO/$\overline{\text{DRDY}}$ before the first SCLK.

Figure 7-52. Read FIFO Buffer Data, 48-Bit Frame Size

The STATUS header returned during a FIFO read command is a combination of STATUS bits stored in the FIFO at the time of the ADC conversion, and STATUS bits stored in the General Configuration Page STATUS registers at the time of the FIFO read operation. Table 7-48 defines which bit fields are retrieved from the FIFO versus the bit fields read from the STATUS register of the Status and General Configuration Page.

Table 7-48. STATUS Bits Retrieval During FIFO read

STATUS WORD	STATUS BIT (FIELD)	READ FROM
STATUS_MSB	STEP_INDICATOR[4:0]	FIFO
	ADC_REF_FAULTn	FIFO
	RESETn	STATUS register
	DRDY	STATUS register
STATUS_LSB	CONV_COUNT[3:0]	FIFO
	FIFO_FAULTn	STATUS register
	INTERNAL_FAULTn	STATUS register
	REG_WRITE_FAULTn	STATUS register
	SPI_CRC_FAULTn	STATUS register

7.5.5 Continuous Read Mode

The ADS125P08 offers a continuous read mode. In continuous read mode, an arbitrary number of register data or FIFO data can be retrieved without any transitions of $\overline{\text{CS}}$, and the frame extends to accommodate the additional data. This simplifies the process of reading a large amount of data, and reduces the overhead on the microcontroller peripheral which control the $\overline{\text{CS}}$ line.

Setting the CONT_READ_EN bit in the CLK_DIGITAL_CFG register enables the continuous read mode. The SPI switches to continuous read mode in the next frame following the SPI frame which changed the CONT_READ_EN bit from 0b to 1b.

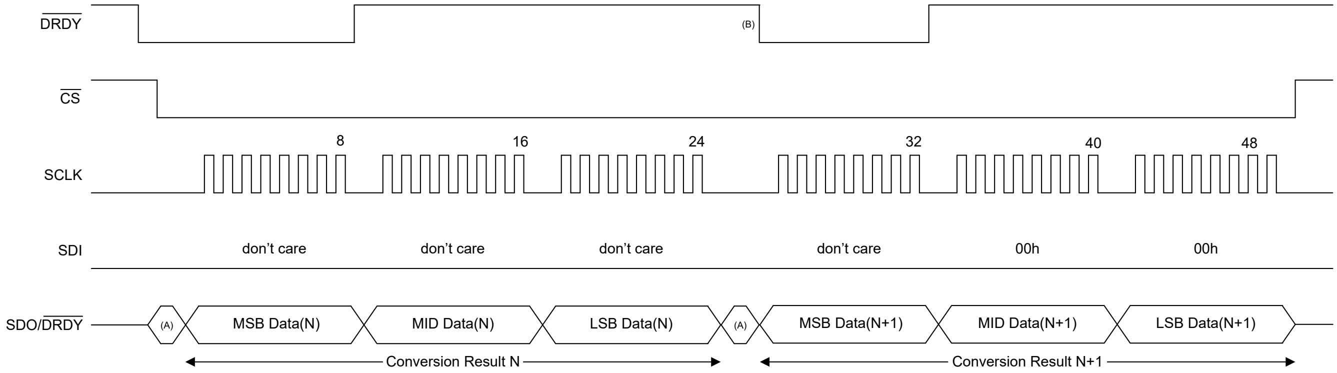
The SPI returns to the default/single read mode by resetting the CONT_READ_EN bit to 0b.

7.5.5.1 Read Conversion Data in Continuous Read Mode

Continuous read mode supports the same command format and clocking as the read conversion data operation shown in the *Read Conversion Data* section, except there is no $\overline{\text{CS}}$ toggling between conversion data reads, and therefore no wait time between conversion data read operations.

Figure 7-53 shows an example of reading two consecutive conversion results "N" and" N+1" with STATUS header and CRC byte disabled. In this example, a new conversion result N+1 completed while reading conversion result N (first 24 SCLK pulses), and this new conversion result is now available in time for the next read, as indicated by the $\overline{\text{DRDY}}$ signal transitioning to low before the 25th clock cycle. Thus, the second read operation returns conversion result N+1. Depending on the relative timing between the clock at CLKIN and the clock at SCLK, there can be a new conversion result available for the next read operation.

This example shows reading of two consecutive conversion results N and N+1 in continuous read mode, however an arbitrary number of conversion results can be read when holding $\overline{\text{CS}}$ low for additional clock periods.

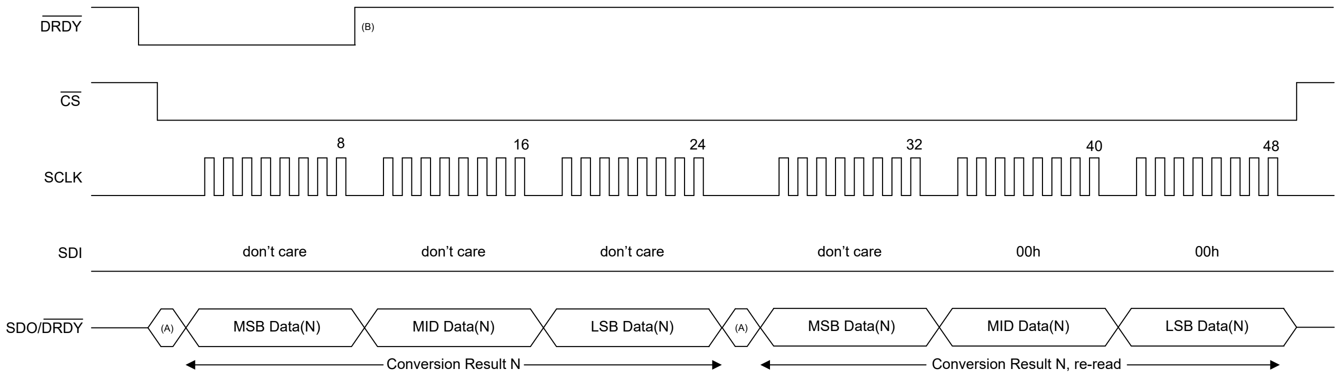


A. If the SDO_MODE bit = 0b, the previous state of SDO/$\overline{\text{DRDY}}$ remains until the first SCLK rising edge. Otherwise, SDO/$\overline{\text{DRDY}}$ follows $\overline{\text{DRDY}}$.

B. In this example, a new conversion result N+1 completed before the second conversion data read.

Figure 7-53. Conversion Data Read in Continuous Read Mode, STATUS and CRC disabled - New Conversion Result Available

Figure 7-54 shows an example of reading conversion data in continuous read mode with STATUS and CRC bytes disabled, but no new conversion data completed. In this case, the second conversion data read operation returns the same conversion result "N".



A. If the SDO_MODE bit = 0b, the previous state of SDO/DRDY remains until the first SCLK rising edge. Otherwise, SDO/DRDY follows DRDY.

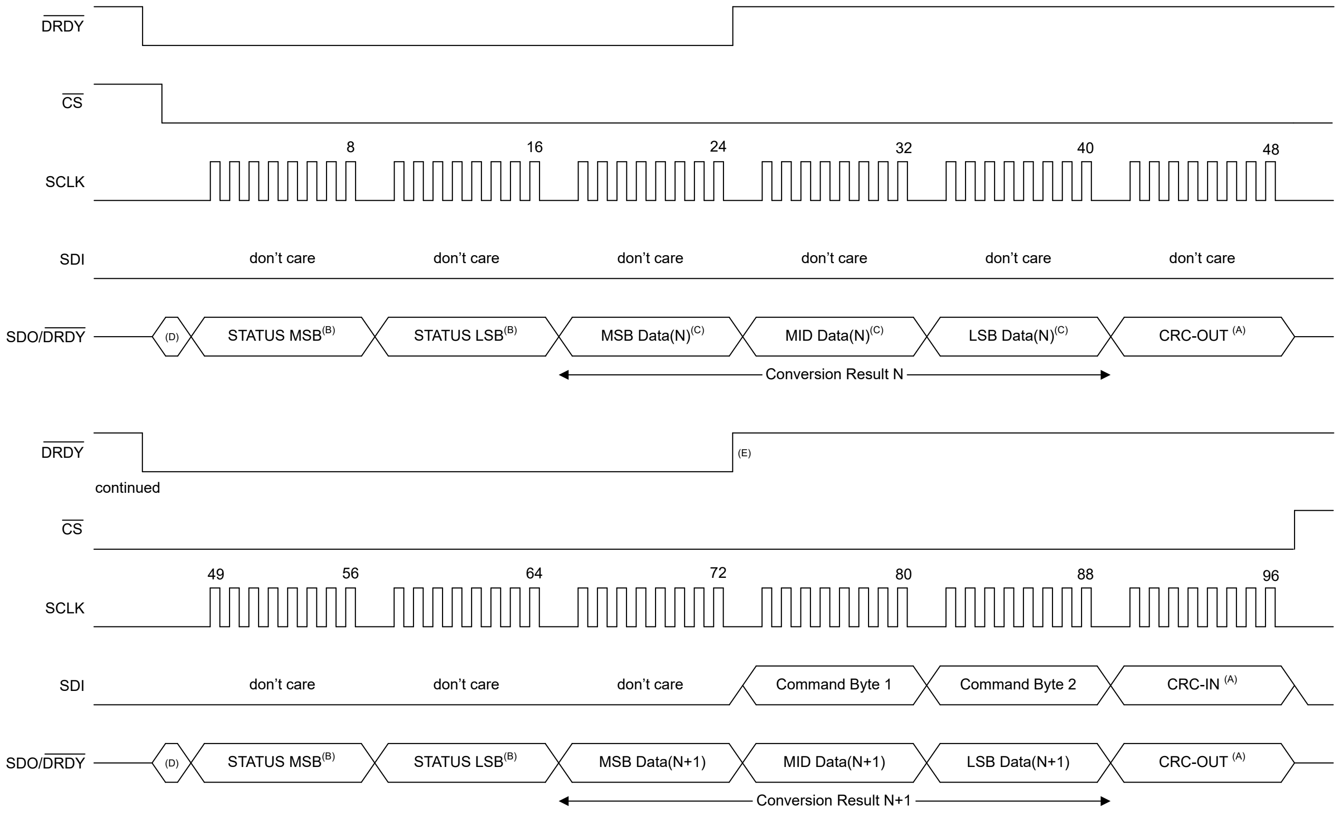
B. In this example, no new conversion result completed before the second conversion data read.

Figure 7-54. Conversion Data Read in Continuous Read Mode, STATUS and CRC disabled - No New Conversion Result Available

Figure 7-55 shows an example of the data read operation in continuous read mode including the STATUS header and the CRC byte. This example also shows the optional full-duplex operation when a command is input at the same time the conversion data is output.

In this example, a new conversion result "N+1" completed while reading conversion result "N", and this new conversion result is now available in time for the next read, as indicated by the DRDY signal transitioning to low before the 49st clock cycle. Thus, the second read operation returns conversion result N+1. Depending on the relative timing between the clock at CLKIN and the clock at SCLK, there can be a new conversion result available for the next read operation.

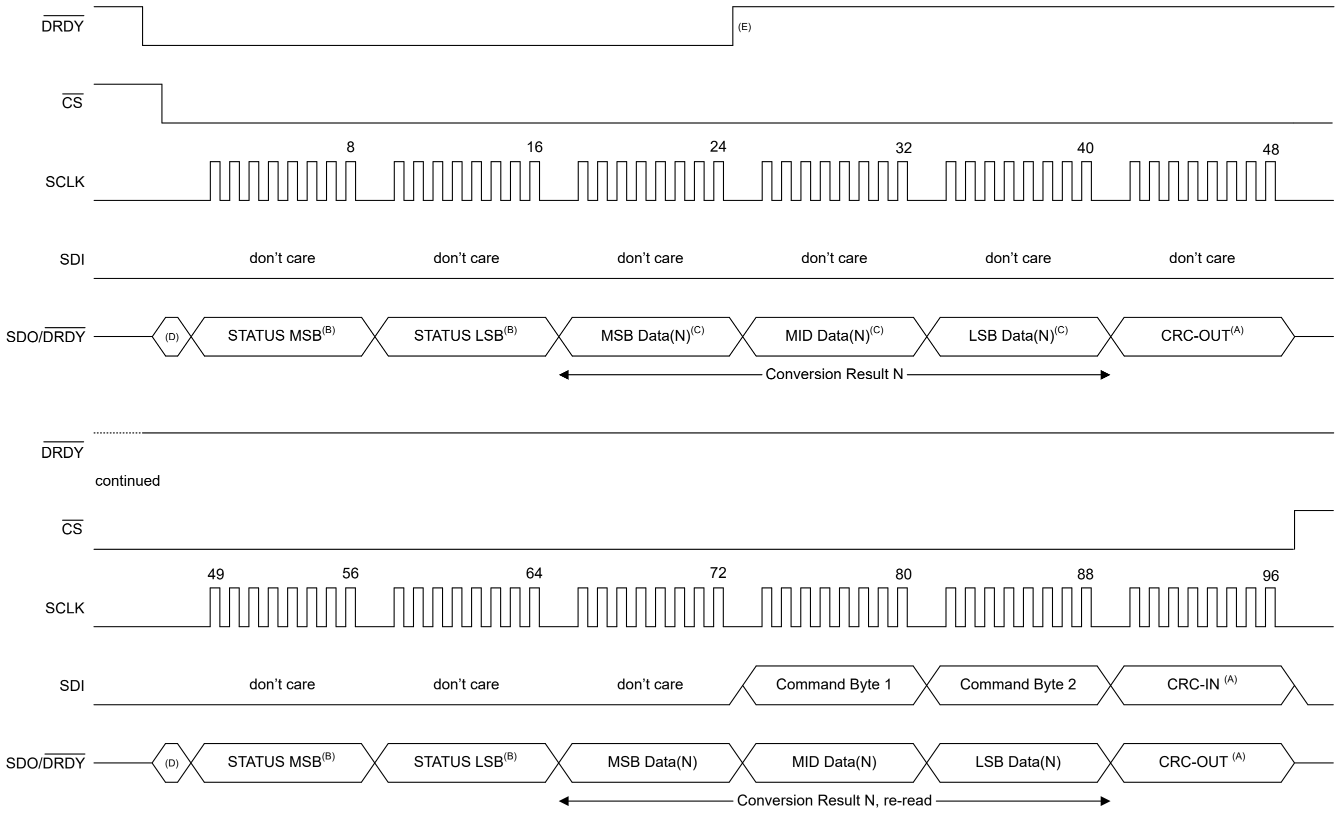
This example shows reading of two consecutive conversion results N and N+1 in continuous read mode, however an arbitrary number of conversion results can be read when holding CS low for additional clock periods.



A. Optional CRC byte. If CRC byte is disabled, the frame shortens accordingly.
B. Optional STATUS header. If STATUS header is disabled, the frame shortens accordingly.
C. Depending on the previous operation, the data field is either conversion data or the register data byte + address byte + 00h padding.
D. If the SDO_MODE bit = 0b, the previous state of SDO/DRDY remains until the first SCLK rising edge. Otherwise, SDO/DRDY follows DRDY.
E. In this example, a new conversion result N+1 completed before the second conversion data read.

Figure 7-55. Conversion Data Read in Continuous Read Mode, STATUS and CRC Enabled - New Conversion Result Available

Figure 7-56 shows an example of reading two conversion results in continuous read mode with STATUS header and CRC byte enabled, but no new conversion data completed during the first read. In this case, the second conversion data read operation returns the same conversion result "N".



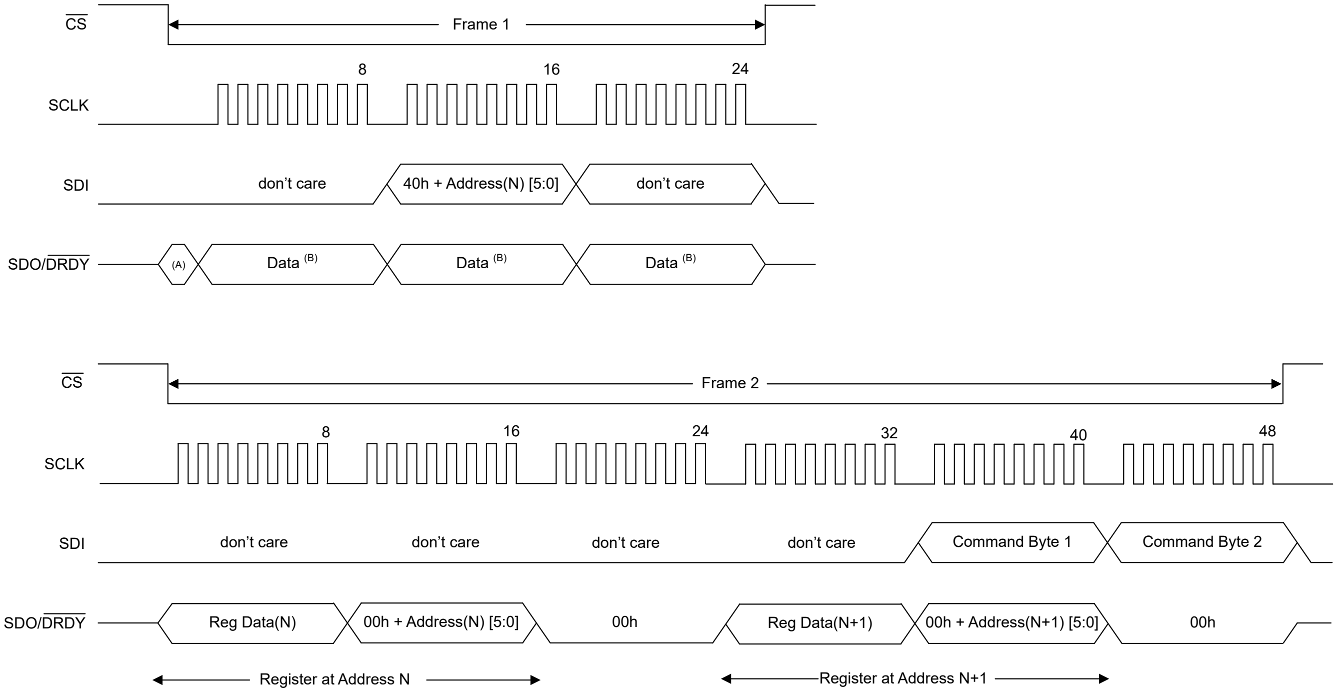
A. Optional CRC byte. If CRC byte is disabled, the frame shortens accordingly.
B. Optional STATUS header. If STATUS header is disabled, the frame shortens accordingly.
C. Depending on the previous operation, the data field is either conversion data or the register data byte + address byte + 00h padding.
D. If the SDO_MODE bit = 0b, the previous state of SDO/DRDY remains until the first SCLK rising edge. Otherwise, SDO/DRDY follows DRDY.
E. In this example, no new conversion result completed before the second conversion data read.

Figure 7-56. Conversion Data Read in Continuous Read Mode, STATUS and CRC Enabled - No New Conversion Result Available

7.5.5.2 Read Registers in Continuous Read Mode

In continuous read mode, use the same command frame to read register data as explained in the *Read Register Command* section. The data response frame returns one or multiple register data bytes depending on when CS is driven back high. The first register data byte is read from the address specified in the command frame. The register address is then automatically incremented by 1 for each subsequent register read. This is also true if the next register address points to an invalid register. The response to registers outside the valid address range is 00h for the data byte, and FFh for the address indicator byte.

Figure 7-57 shows an example of reading register data in continuous read mode with STATUS and CRC bytes disabled. This example shows reading of two consecutive registers N and N+1, however an arbitrary number of registers can be read when holding CS low for additional clock periods.

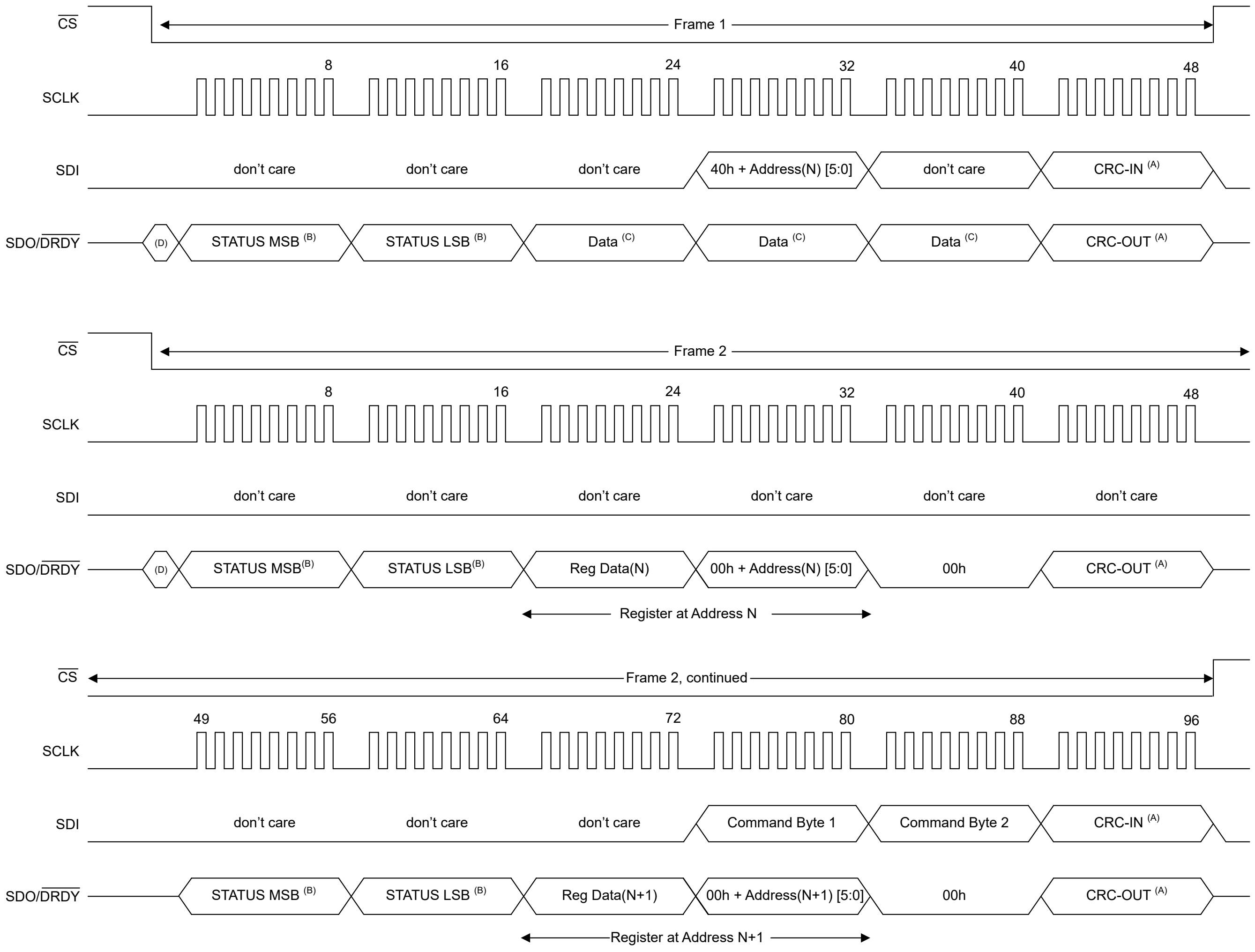


A. Previous state of SDO/DRDY before first SCLK.

B. Depending on the previous operation, the data field is either conversion data or the register data byte + address byte + 00h padding.

Figure 7-57. Read Register Data in Continuous Read Mode, STATUS Header and CRC Byte Disabled

Figure 7-58 shows an example of the read register operation in continuous read mode when STATUS header and CRC byte are enabled. In the input and output frame, do not care bytes and 00h pad bytes are used to match the data frame protocol as explained in the *Read Register Command* section. This example shows reading of two consecutive registers N and N+1, however an arbitrary number of registers can be read when holding CS low for additional clock periods.



A. Optional CRC byte. If CRC disabled, the frame shortens by one byte.

B. Optional STATUS byte. If STATUS is disabled, the frame shortens by two bytes.

C. Depending on the previous operation, the data field is either conversion data or the register data byte + address byte + 00h padding.

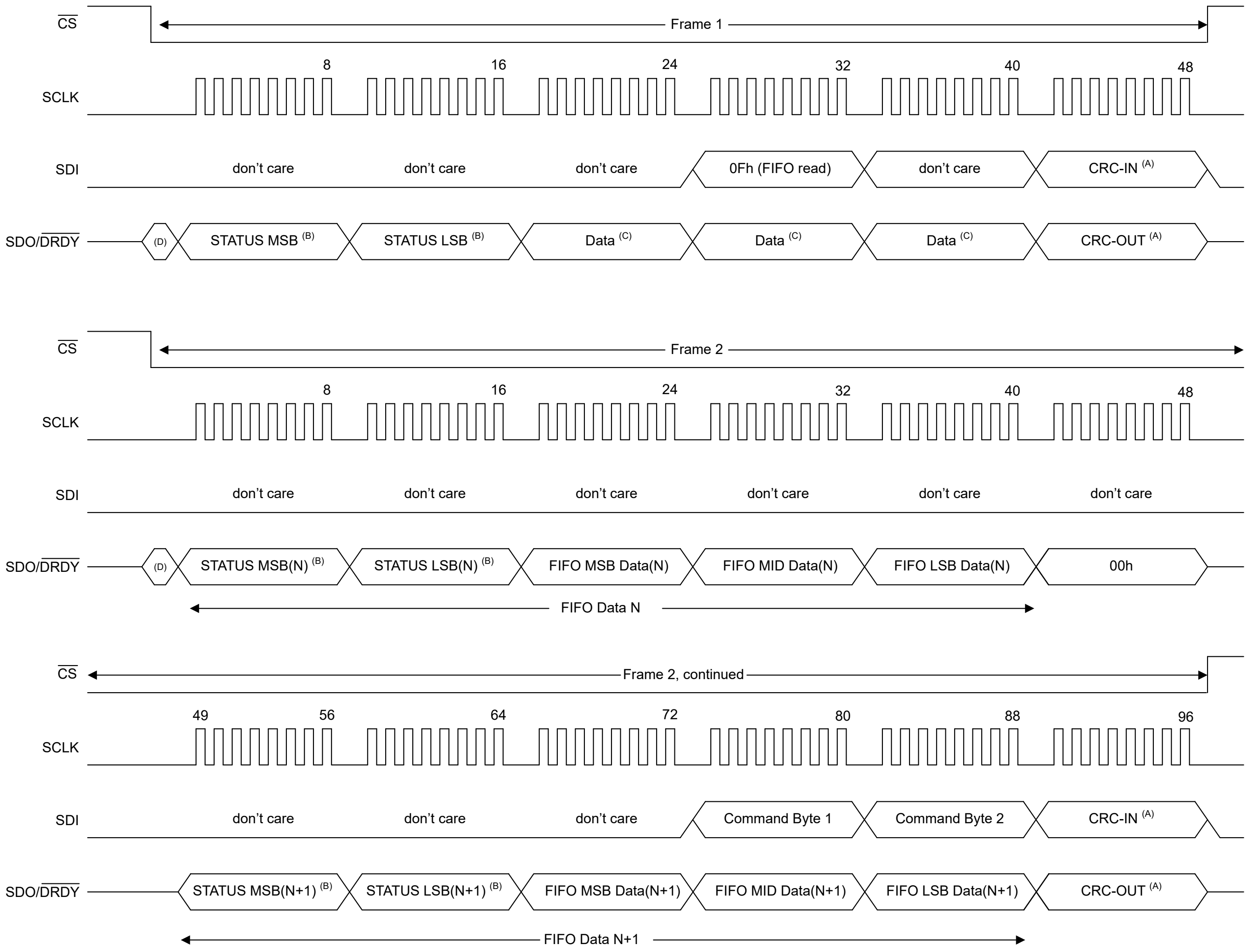
D. Previous state of SDO/ $\overline{\text{DRDY}}$ before first SCLK.

Figure 7-58. Read Register Data in Continuous Read Mode, STATUS Header and CRC Byte Enabled

7.5.5.3 Read FIFO Buffer in Continuous Read Mode

In continuous read mode, use the same command frame to read FIFO data as explained in the *Read FIFO Buffer Command* section. The data response frame returns FIFO data for one or multiple FIFO addresses depending on the timing of $\overline{\text{CS}}$ driven back high. As explained in the *FIFO Buffer Read and Write* section, the FIFO buffer read pointer defines the memory location inside the FIFO for the data word to be read. See the *FIFO Buffer Read and Write* section for a detailed description of the FIFO buffer operation. In continuous read mode, the FIFO buffer read pointer is automatically incremented by 1 for each subsequent FIFO buffer read inside the response frame.

Figure 7-59 shows an example of the read FIFO buffer operation in continuous read mode when STATUS header and CRC byte are enabled. This example shows reading data from two consecutive FIFO addresses N and N+1, however an arbitrary number of FIFO address locations can be read when holding $\overline{\text{CS}}$ low for additional SCLK periods.



A. Optional CRC byte. If CRC disabled, the frame shortens by one byte. The CRC-OUT covers the whole frame.
B. Optional STATUS header. If STATUS is disabled, the frame shortens by two bytes per FIFO address read.
C. Depending on the previous operation, the data field is either conversion data or the register data byte + address byte + 00h padding.
D. Previous state of SDO/ DRDY before first SCLK.

Figure 7-59. Read FIFO Buffer Data in Continuous Read Mode, STATUS header and CRC byte enabled

The STATUS header returned during a FIFO read command is a combination of STATUS bits stored in the FIFO at the time of the ADC conversion, and STATUS bits stored in the General Configuration Page STATUS registers at the time of the FIFO read operation. See the *Read FIFO Buffer Command* section for a detailed description.

7.5.6 SPI communication after POR or Reset

During device power-up or while the device is held in reset, the user registers reset to the default values, and the SDO/$\overline{\text{DRDY}}$ pin is in a high-Z state while the $\overline{\text{DRDY}}$ pin drives low. After power-up or reset, the SDO/$\overline{\text{DRDY}}$ pin is in output-only mode (SDO_MODE = 0b) and in a high-Z state as long as $\overline{\text{CS}}$ is high, and the $\overline{\text{DRDY}}$/GPIO1 pin is in $\overline{\text{DRDY}}$ mode. As shown in Figure 7-60 and Figure 7-61, the $\overline{\text{DRDY}}$ pin drives high after the POR is released and indicates that the device is ready for communication. The device is ready for SPI communication t_{POR} after a power up, and t_{REGACQ} after a reset.

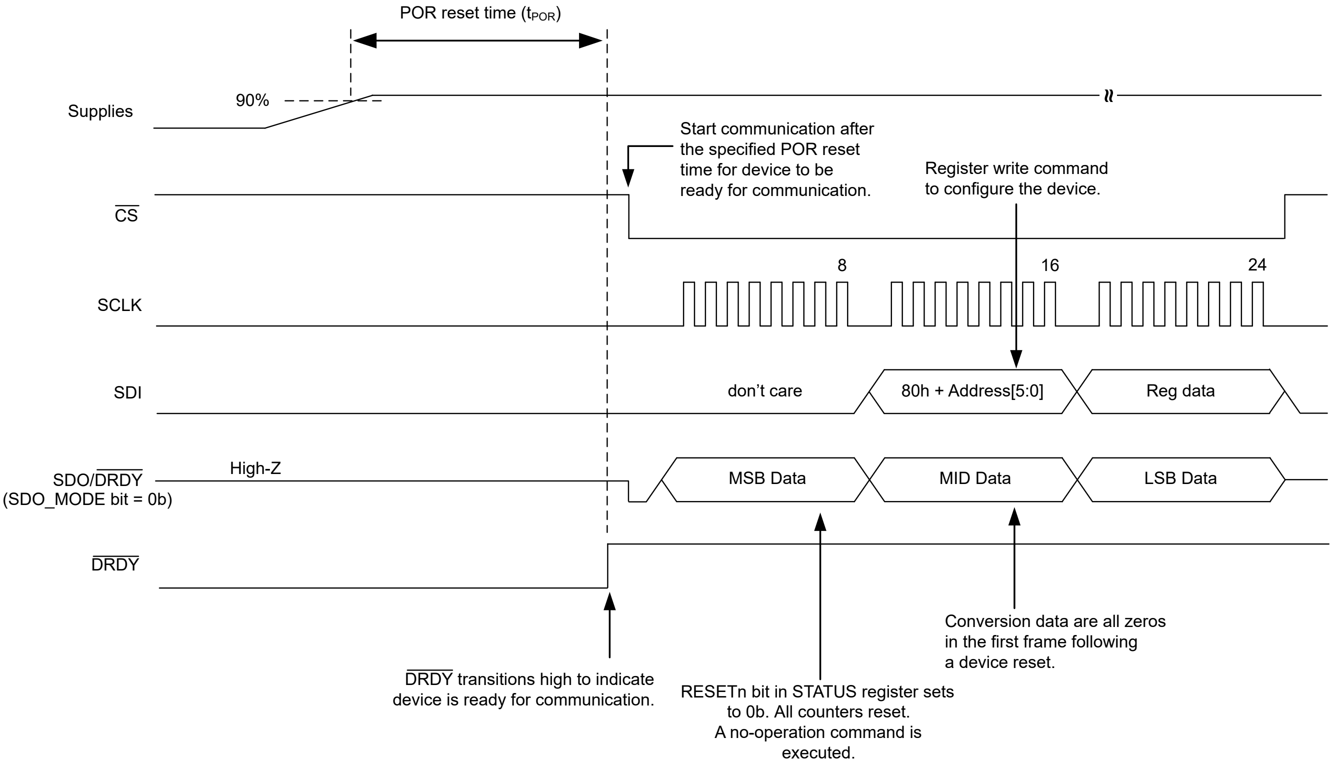

POR reset time (t_{POR})
Supplies
90%
$\overline{\text{CS}}$
SCLK
SDI
SDO/$\overline{\text{DRDY}}$
(SDO_MODE bit = 0b)
$\overline{\text{DRDY}}$
High-Z
8
16
24
don't care
80h + Address[5:0]
Reg data
MSB Data
MID Data
LSB Data
Start communication after the specified POR reset time for device to be ready for communication.
Register write command to configure the device.
$\overline{\text{DRDY}}$ transitions high to indicate device is ready for communication.
RESETn bit in STATUS register sets to 0b. All counters reset. A no-operation command is executed.
Conversion data are all zeros in the first frame following a device reset.

Figure 7-60. SPI communication after Power-On Reset

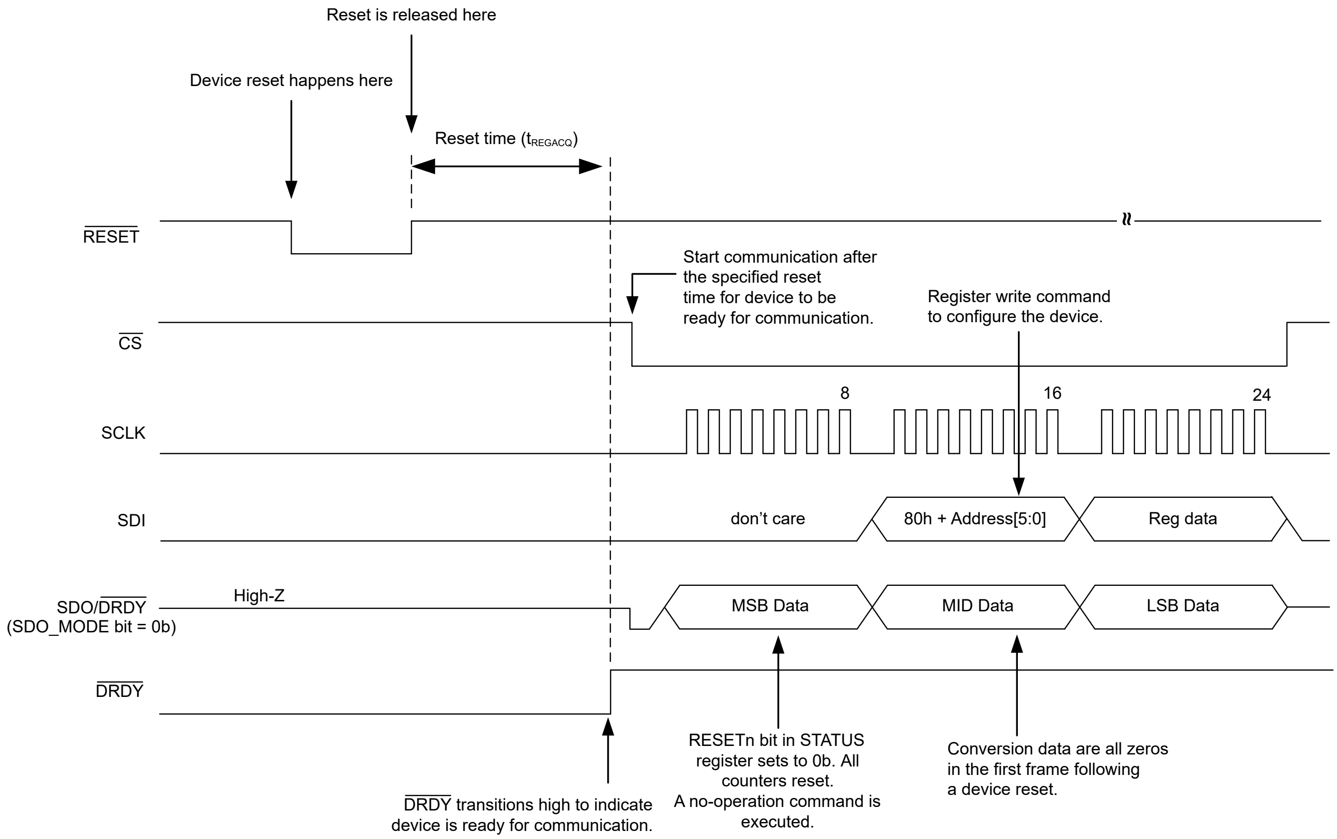

Reset is released here
Device reset happens here
Reset time (t_{REGACQ})
RESET
Start communication after the specified reset time for device to be ready for communication.
Register write command to configure the device.
CS
8
16
24
SCLK
SDI
don't care
80h + Address[5:0]
Reg data
SDO/DRDY (SDO_MODE bit = 0b)
High-Z
MSB Data
MID Data
LSB Data
DRDY
DRDY transitions high to indicate device is ready for communication.
RESETn bit in STATUS register sets to 0b. All counters reset. A no-operation command is executed.
Conversion data are all zeros in the first frame following a device reset.

Figure 7-61. SPI communication after Reset

7.5.7 $\overline{\text{DRDY}}$ Pin Behavior

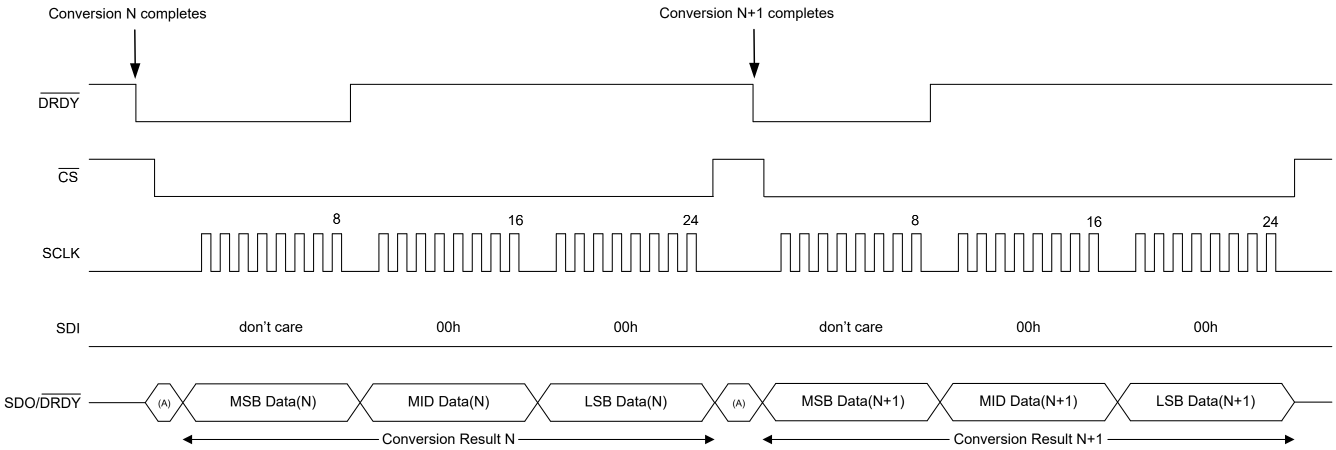
This section provides details about the $\overline{\text{DRDY}}$ pin behavior in various scenarios. In all cases, the DRDY_CFG[1:0] bits are reset to 00b. $\overline{\text{DRDY}}$ transitions low whenever new conversion data complete. If $\overline{\text{DRDY}}$ is low when a new conversion completes, then $\overline{\text{DRDY}}$ drives high $t_{w(DRH)}$ before the $\overline{\text{DRDY}}$ falling edge (see Figure 7-63 and Figure 7-65).

The device avoids data corruption if new conversions N+1 complete while conversion data N are being read. Conversion data N+1 are held in an internal buffer until the read of conversion data N is complete. In the following frame, conversion data N+1 are loaded into the SDO output buffer. $\overline{\text{DRDY}}$ does not transition high after conversion data N have been read in this case to indicate that new conversion data N+1 are available for readout (see Figure 7-65).

$\overline{\text{DRDY}}$ transitions high at the eighth SCLK falling edge during conversion data read (Figure 7-62), assuming the STATUS header is disabled. If $\overline{\text{CS}}$ is driven high before the eighth SCLK, then $\overline{\text{DRDY}}$ stays low, indicating that conversion data is not read (Figure 7-63 and Figure 7-64).

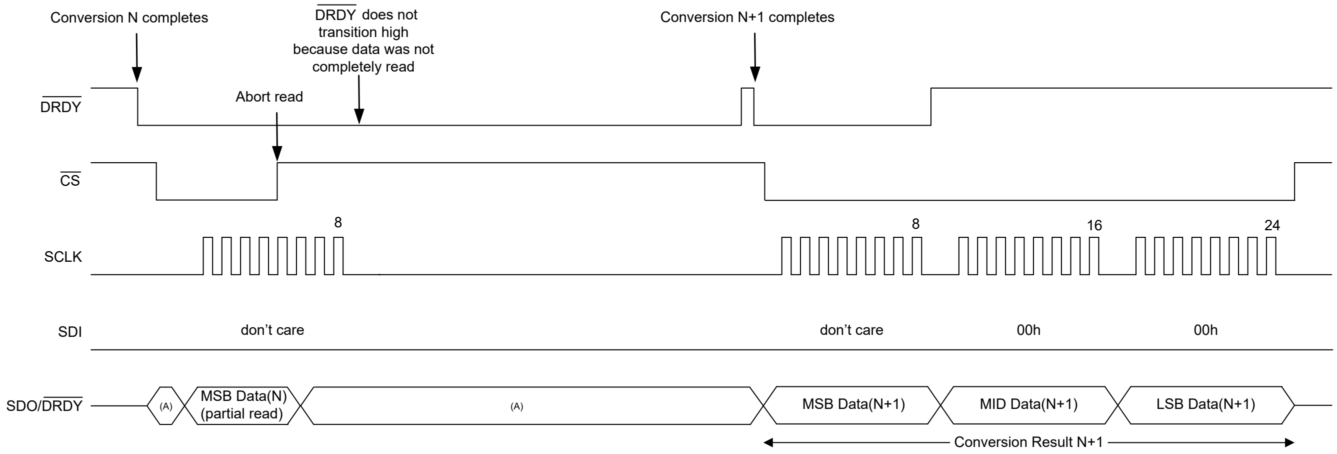
Figure 7-64 shows that the same conversion data can be read multiple times until new conversions complete. The conversion counter (CONV_COUNT[3:0] bits in the STATUS_LSB register) indicates if the same data are read again or if new data is read.

Figure 7-66 illustrates that conversion data N+1 are lost when the host does not read the data before conversions N+2 complete. The conversion counter is helpful in this situation to detect if the host missed reading the intermediate conversion results.



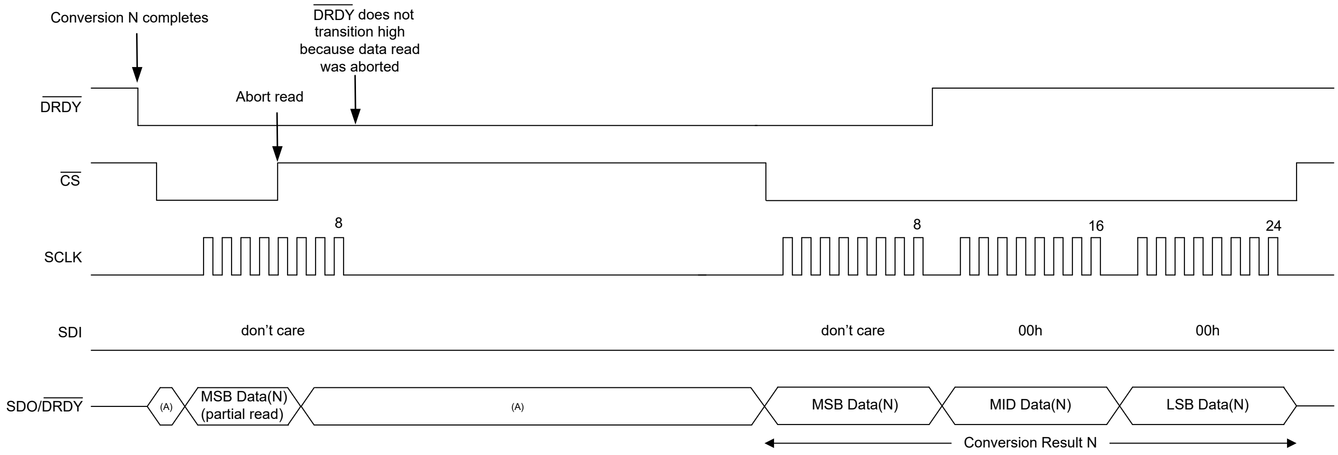
A. If the SDO_MODE bit = 0b, the previous state of SDO/$\overline{\text{DRDY}}$ remains until the first SCLK rising edge. Otherwise, SDO/$\overline{\text{DRDY}}$ follows $\overline{\text{DRDY}}$.

Figure 7-62. $\overline{\text{DRDY}}$ Pin Behavior: Reading Conversion Data Before New Conversions Complete



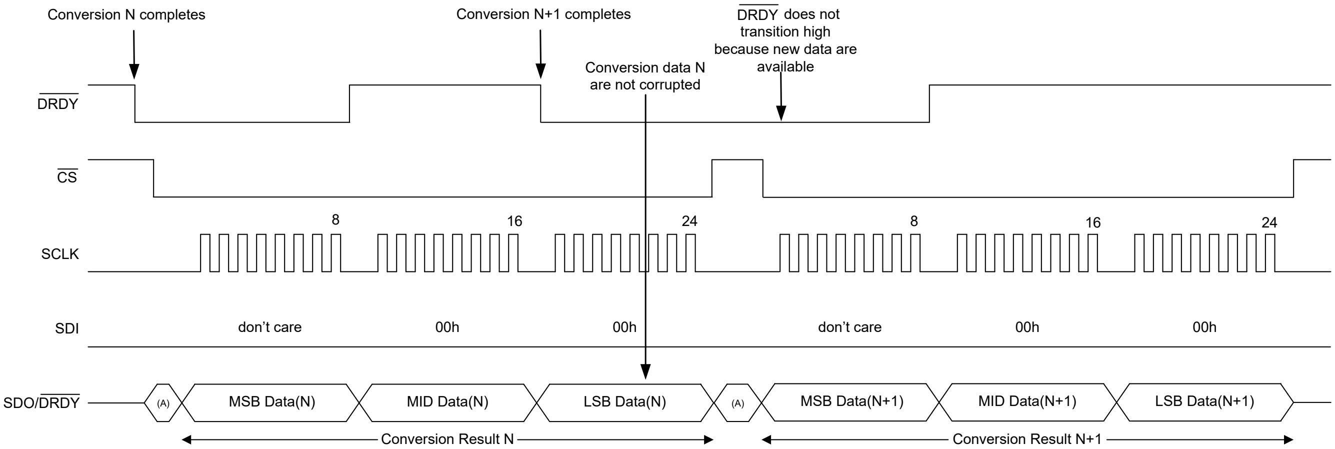
A. If the SDO_MODE bit = 0b, the previous state of SDO/$\overline{\text{DRDY}}$ remains until the first SCLK rising edge. Otherwise, SDO/$\overline{\text{DRDY}}$ follows $\overline{\text{DRDY}}$.

Figure 7-63. $\overline{\text{DRDY}}$ Pin Behavior: Incomplete Read of Conversion Data Before New Conversions Complete



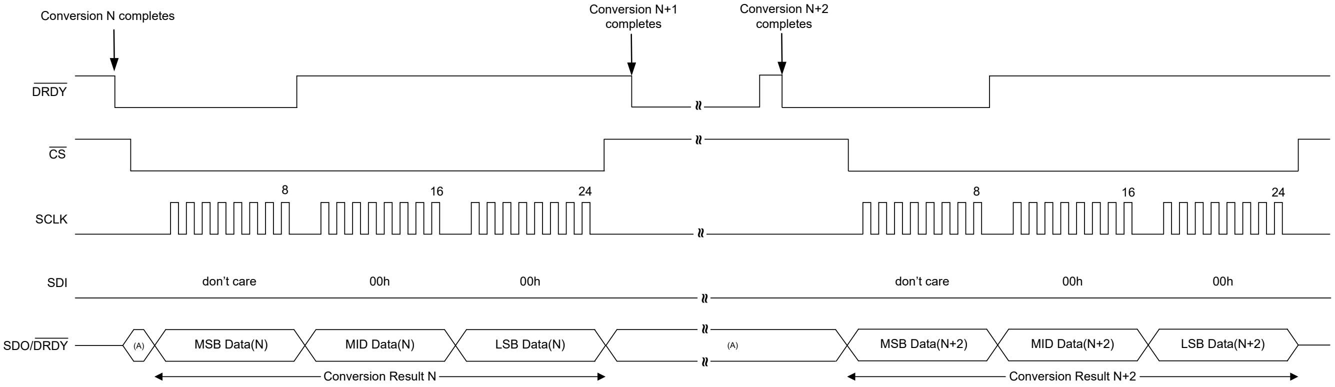
A. If the SDO_MODE bit = 0b, the previous state of SDO/$\overline{\text{DRDY}}$ remains until the first SCLK rising edge. Otherwise, SDO/$\overline{\text{DRDY}}$ follows $\overline{\text{DRDY}}$.

Figure 7-64. $\overline{\text{DRDY}}$ Pin Behavior: Incomplete Read of Conversion Data Followed by Complete Read of Same Conversion Data



A. If the SDO_MODE bit = 0b, the previous state of SDO/DRDY remains until the first SCLK rising edge. Otherwise, SDO/DRDY follows DRDY.

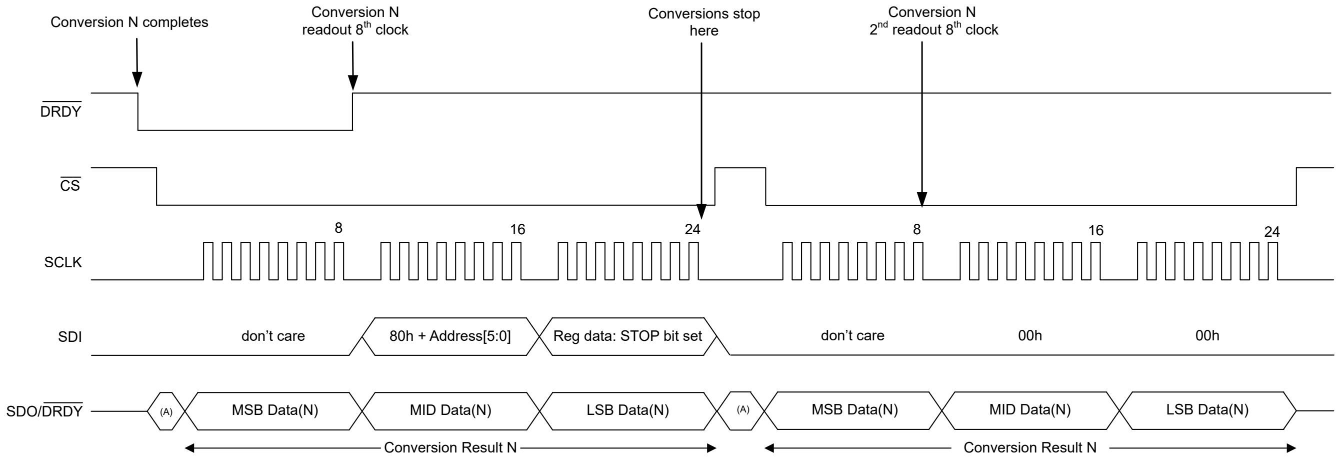
Figure 7-65. DRDY Pin Behavior: Reading Conversion Data While New Conversions Complete



A. If the SDO_MODE bit = 0b, the previous state of SDO/DRDY remains until the first SCLK rising edge. Otherwise, SDO/DRDY follows DRDY.

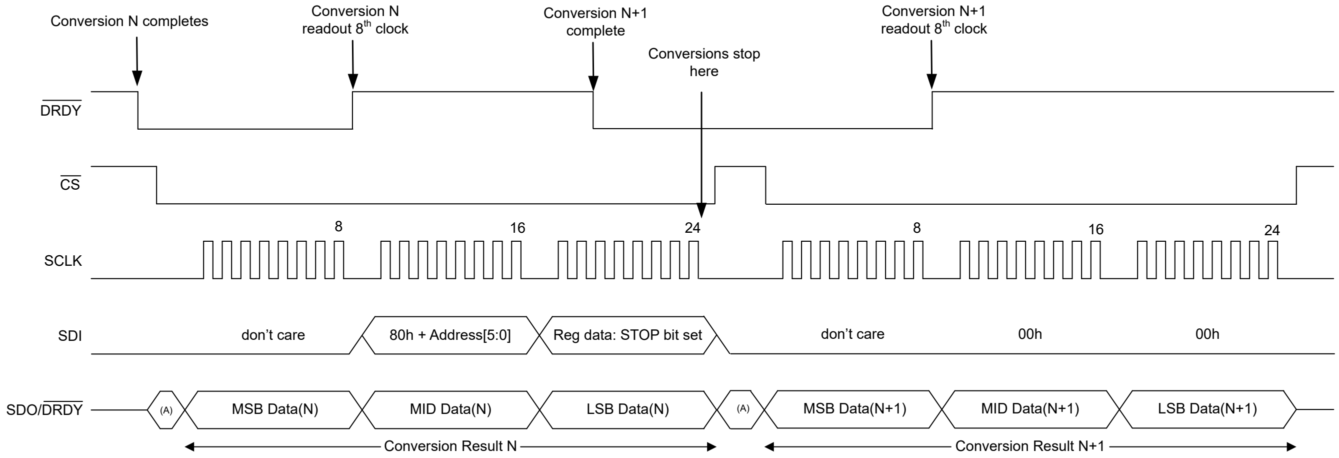
Figure 7-66. DRDY Pin Behavior: Missed Reading Intermediate Conversion Results

For the following examples, assume that STOP_BEHAVIOR[1:0] = 00b. Setting the STOP bit stops conversions at the last SCLK falling edge within the SPI frame where the CONVERSION_CTRL register is written. However, the DRDY pin does not transition high and old conversion data can still be read until new conversions become available. Figure 7-67 shows the device behavior when setting the STOP bit to abort an ongoing conversion while reading out conversion data. Figure 7-68 shows a scenario where new conversions complete while setting the STOP bit and reading out conversion data.



A. If the SDO_MODE bit = 0b, the previous state of SDO/$\overline{\text{DRDY}}$ remains until the first SCLK rising edge. Otherwise, SDO/$\overline{\text{DRDY}}$ follows $\overline{\text{DRDY}}$.

Figure 7-67. $\overline{\text{DRDY}}$ Pin Behavior: Setting the STOP Bit While Reading Conversion Data



A. If the SDO_MODE bit = 0b, the previous state of SDO/$\overline{\text{DRDY}}$ remains until the first SCLK rising edge. Otherwise, SDO/$\overline{\text{DRDY}}$ follows $\overline{\text{DRDY}}$.

Figure 7-68. $\overline{\text{DRDY}}$ Pin Behavior: Setting the STOP Bit and Reading Conversion Data While New Conversions Complete

7.5.8 Daisy-Chain Operation

In systems using multiple ADCs, the devices can be connected in a daisy-chain string to reduce the number of SPI connections. A daisy-chain connection links together the SPI output of one device to the SPI input of the next device so the devices in the chain appear as a single logical device to the host controller. There is no special programming required for daisy-chain operation, apply additional shift clocks to access all devices in the chain. For simplified operation, program the same SPI frame size for each device (for example, when enabling the CRC option of all devices, thus producing a 32 bit frame size).

Figure 7-69 shows four devices connected in a daisy-chain configuration. SDI of ADS125P08 (1) connects to the host SPI data output, and SDO/$\overline{\text{DRDY}}$ of ADS125P08 (4) connects to the host SPI data input. The shift operation is simultaneous for all devices in the chain. After each ADC shifts out the conversion data, the data of SDI appears on SDO/$\overline{\text{DRDY}}$ to drive the SDI of the next device in the chain. The shift operation continues until the last device in the chain is reached. The SPI frame ends when $\overline{\text{CS}}$ is taken high, at which time the data shifted into each device is interpreted. For daisy-chain operation, program the SDO/$\overline{\text{DRDY}}$ pin to data output only mode using the SDO_MODE bit.

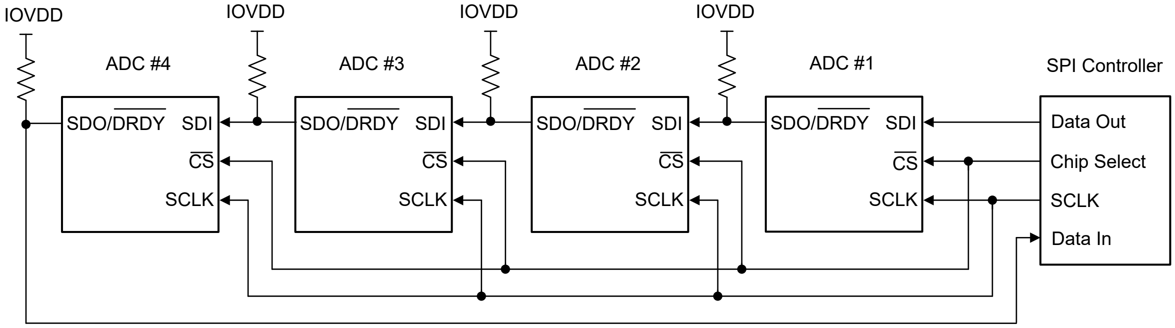


Figure 7-69. Daisy-Chain Connection

Figure 7-69 shows a pull-up resistor at each data output pin. Use a 10kΩ pull-up resistor if the SDO/$\overline{\text{DRDY}}$ pin is used in dual-function mode for fast response times between SPI frames (this is especially important when using edge triggered interrupts). However, when only using the pin as an SDO, a weaker pull-up such as a 100kΩ resistor is sufficient.

Figure 7-70 shows the 24 bits frame size of each device used at initial communication after device power up.

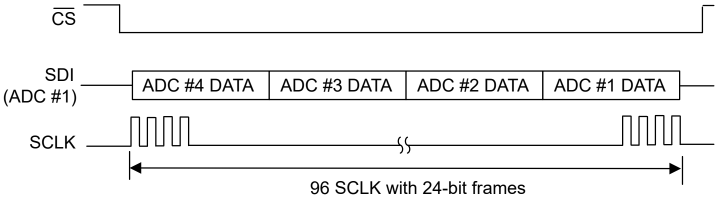
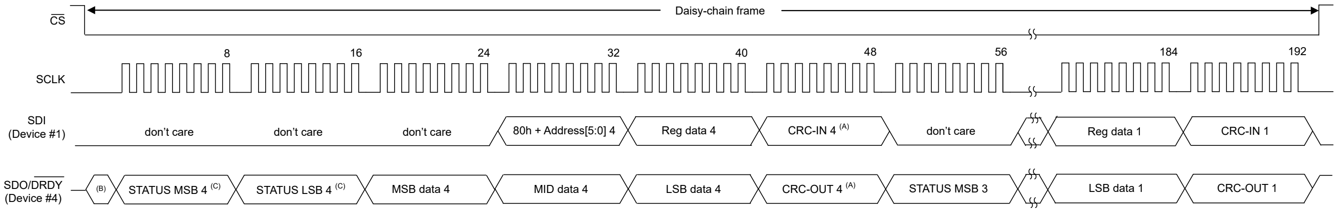


Figure 7-70. 24 bit Data Input Sequence

To input data, the host first shifts in the data intended for the last device in the chain. The number of input bytes for each ADC is sized to match the output frame size. The default frame size is 24 bits, so initially each ADC requires three bytes by prefixing a pad byte in front of the two command bytes. The input data of ADC #4 is first, followed by the input data of ADC #3, and so forth.

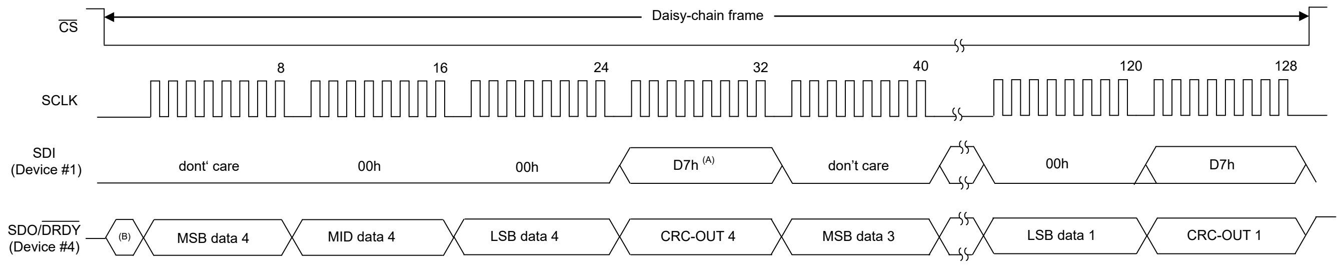
Figure 7-71 shows the detailed input data sequence for the daisy-chain write register operation of Figure 7-69. 48 bit frames for each ADC are shown (24 bits of data, with the STATUS header and CRC byte enabled). Command operations can be different for each ADC. The read register operation requires a second frame operation to read out the register data.



A. Optional CRC byte. If CRC is disabled, the frame shortens by one byte.
B. Previous state of SDO/$\overline{\text{DRDY}}$ before SCLK is applied.
C. Optional STATUS header. If STATUS is disabled, the frame shortens by two bytes.

Figure 7-71. Write Register Data in Daisy-Chain Connection

Figure 7-72 shows the clock sequence to read conversion data from the device connection provided in Figure 7-69. This example illustrates a 32 bit output frame (24 bits of data, with the CRC byte enabled). The output data of ADC (4) is first in the sequence, followed by the data of ADC (3), and so on. The total number of clocks required to shift out the data is given by the number of bits per frame × the number of devices in the chain. In this example, 32 bit output frames × four devices result in 128 total clocks.



A. Optional CRC byte. If CRC is disabled, the frame shortens by one byte.

B. Previous state of SDO/$\overline{\text{DRDY}}$ before SCLK is applied.

Figure 7-72. Read Conversion Data in Daisy-Chain Connection

As shown in Equation 31, the maximum number of devices connected in daisy-chain configuration is limited by the SCLK signal frequency, data rate, and number of bits per frame.

$$\text{Maximum devices in a chain} = \lfloor f_{SCLK} / (f_{DATA} \times \text{bits per frame}) \rfloor \quad (31)$$

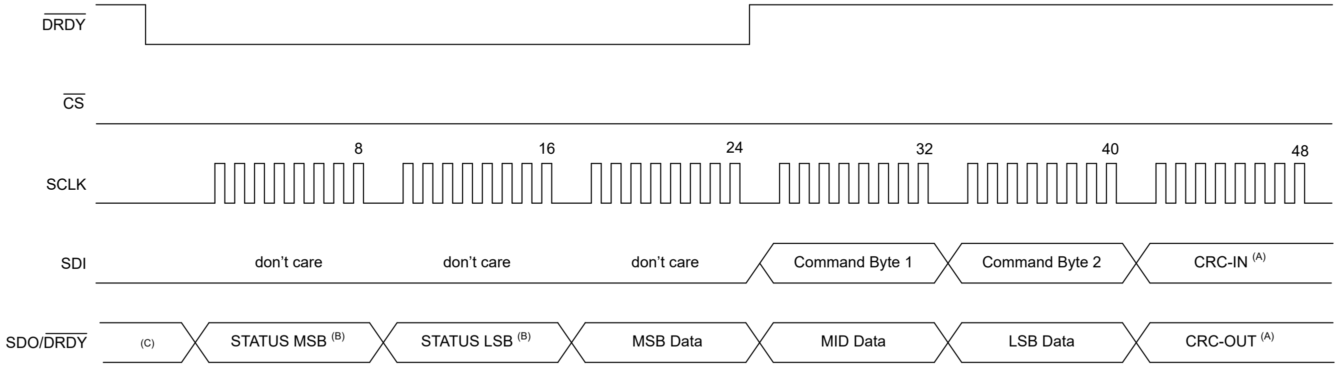
For example, if f_{SCLK} = 20MHz, f_{DATA} = 100 kSPS, and 32 bit frames are used, the maximum number of daisy-chain connected devices is the floor of: $\lfloor 20\text{MHz} / (100\text{kHz} \times 32) \rfloor = 6$.

7.5.9 3-Wire SPI Mode

The ADC has the option of 3-wire SPI operation by grounding $\overline{\text{CS}}$. 3-wire mode is detected by the ADC when $\overline{\text{CS}}$ is grounded at power up or after reset. The device changes to 4-wire SPI mode any time $\overline{\text{CS}}$ is taken high.

Because $\overline{\text{CS}}$ no longer controls frame timing in 3-wire SPI mode, SCLKs are counted by the ADC to determine the beginning and ending of a frame. The number of SCLK bits must be controlled by the host and must match the size of the output frame. The number of bits per frame depend on the device configuration. The size of the output frame is shown in Table 7-45. Because frame timing is determined by the number of SCLKs, avoid inadvertent SCLK transitions, such as those possibly occurring at power up.

3-wire SPI mode supports the same command format and clocking as the 4-wire mode, except there is no $\overline{\text{CS}}$ toggling and therefore no wait time between frames. Figure 7-73 shows an example for reading conversion data in 3-wire SPI mode.



A. Optional CRC byte. If the CRC is disabled, the frame shortens by one byte.

B. Optional STATUS header. If STATUS is disabled, the frame shortens by two bytes.

C. If the SDO_MODE bit = 0b, the previous state of SDO/$\overline{\text{DRDY}}$ remains until the first SCLK rising edge. Otherwise, SDO/$\overline{\text{DRDY}}$ follows $\overline{\text{DRDY}}$.

Figure 7-73. Read Conversion Data in 3-Wire SPI Mode

7.5.9.1 3-Wire SPI Mode Frame Re-Align

In 3-wire SPI mode, an unintended SCLK can misalign the frame, resulting in loss of frame synchronization to the host. As shown in Figure 7-74, the SPI is resynchronized without requiring an ADC reset by sending an SPI re-align pattern. The re-align pattern is a minimum of 63 consecutive 1s followed by one 0 at the 64th SCLK. The 65th SCLK starts a new SPI frame. The device also accepts a re-align pattern with more than 63 consecutive 1s followed by one 0. In that case, the new frame starts with the SCLK rising edge following the 0. Optionally, the ADC can be completely reset by toggling $\overline{\text{RESET}}$ or by the reset pattern shown in the *Reset by SPI Input Pattern* section.

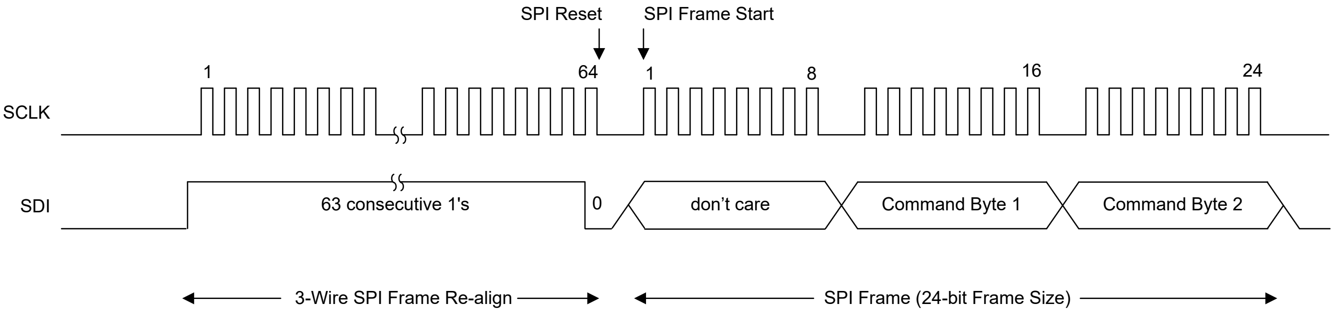


Figure 7-74. 3-Wire Mode SPI Re-Align Pattern

7.5.10 Conversion Data

Conversion data are coded depending on the CODING bit. By default, conversion data are coded in binary two's-complement format, MSB first (sign bit). Set the CODING bit to 1b for unipolar straight binary format. Table 7-49 shows the output code depending on the format. In binary two's-complement format, the conversion data clips to positive and negative full-scale code values when the input signal exceeds the respective positive and negative full-scale values. In unipolar straight binary format, conversion data clips to full-scale or zero code values when the input signal exceeds the full-scale value or the value is below zero.

Table 7-49. Ideal Output Code Versus Input Signal

DIFFERENTIAL INPUT VOLTAGE (V)	IDEAL OUTPUT CODE[1]	
	Binary two's-complement format (CODING = 0b)	Unipolar straight binary format (CODING = 1b)
≥ FSR × $(2^{24} – 1) / 2^{24}$	7FFFFFh	FFFFFFh
≥ FSR × $(2^{23} – 1) / 2^{23}$		FFFFFEh
FSR / 2^{23}	000001h	000002h
0	000000h	000000h
–FSR / 2^{23}	FFFFFFh	
–FSR × $(2^{23} – 1) / 2^{23}$	800001h	
≤ –FSR	800000h	

[1] Ideal output data, excluding offset, gain, linearity, and noise errors.

7.5.11 Data Ready

There are several methods available to determine when conversion data are ready for readback.

1. Hardware: Monitor the $\overline{\text{DRDY}}$ or the SDO/$\overline{\text{DRDY}}$ pin, see the *$\overline{\text{DRDY}}$ Pin and SDO/$\overline{\text{DRDY}}$ Pin* section for details.
2. Software: Monitor the DRDY bit of the STATUS header. See the *DRDY Bit* section for details.
3. Clock counting: Count the number of ADC main clocks to predict when data are ready. See the *Clock Counting* section for details.

7.5.11.1 $\overline{\text{DRDY}}$ Pin and SDO/$\overline{\text{DRDY}}$ Pin

$\overline{\text{DRDY}}$ is the data-ready output signal pin, and SDO/$\overline{\text{DRDY}}$ pin is a dual-function output pin. See the *Data Ready ($\overline{\text{DRDY}}$) Pin* section for a description of the $\overline{\text{DRDY}}$ pin and the *Serial Data Output/Data Ready (SDO/$\overline{\text{DRDY}}$)* section for a description of the SDO/$\overline{\text{DRDY}}$ pin.

7.5.11.2 DRDY Bit

The software method of determining data ready is by polling the DRDY bit (bit 0 of the STATUS header). When DRDY = 1b, the data are new from the last data read operation, otherwise the data provided are the previous data. After data are read, the bit stays cleared until the next conversion data are ready. To avoid missing data, poll the bit at least as often as the output data rate.

7.5.11.3 Clock Counting

Another method to determine when data are ready is to count ADC main clock cycles. This method is only possible when using an external clock because the internal clock oscillator is not observable. After synchronization or conversion start, the number of clock cycles is larger compared to the normal conversion data period. The initial number of clock cycles is equal to the latency time of the digital filter as listed in the *Digital Filter* section.

7.6 Register Map

The ADS125P08 register space is organized in multiple register pages. See the *Channel Auto-Sequencer* section for details on the auto-sequencer and the register page structure.

Page 0 is referred to as the "Status and General Configuration Page" (or just "General Configuration Page") and holds status and generic configuration data. See the *ADS125P08 Step Configuration Page* section for the register map of the Status and General Configuration Page.

A separate register page exists for each sequence step configuration (pages 1 to 32), referred to as the "Step Configuration Pages". The step registers are identified by the prefix "STEP_x", with x = 0 to 31 indicating the sequence step number. See the *ADS125P08 Step Configuration Page* section for the register map of a Step Configuration Page x.

See the *Configuring the Auto-Sequencer section* for further details on page addressing, and how to configure the auto-sequencer.

7.6.1 ADS125P08 Status and General Configuration Page

Table 7-50 lists the memory-mapped registers for the ADS125P08 Status and General Configuration Page registers. All register offset addresses not listed in Table 7-50 should be considered as reserved locations and the register contents should not be modified.

Table 7-50. Register Map

Address	Acronym	Reset	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
00h	DEVICE_ID	30h	RESERVED		CH_CNT[1:0]		DEV_ID[3:0]			
01h	REVISION_ID	01h	REV_ID[7:0]							
02h	STATUS_MSB	00h	STEP_INDICATOR[4:0]					ADC_REF_FAULTn	RESETn	DRDY
03h	STATUS_LSB	FFh	CONV_COUNT[3:0]				FIFO_FAULTn	INTERNAL_FAULTn	REG_WRITE_FAULTn	SPI_CRC_FAULTn
04h	ADC_REF_STATUS	B0h	RESERVED	AVDD_UVn	REF_UVn	MOD_OVR_FAULTn	RESERVED			
05h	DIGITAL_STATUS	FFh	CRC_FAULT_PAGE[5:0]						MEM_INTERNAL_FAULTn	REG_MAP_CRC_FAULTn
06h	AGPIO_DATA_INPUT	00h	AGPIO7_DAT_IN	AGPIO6_DAT_IN	AGPIO5_DAT_IN	AGPIO4_DAT_IN	AGPIO3_DAT_IN	AGPIO2_DAT_IN	AGPIO1_DAT_IN	AGPIO0_DAT_IN
07h	GPIO_DATA_INPUT	02h	RESERVED				GPIO3_DAT_IN	GPIO2_DAT_IN	GPIO1_DAT_IN	GPIO0_DAT_IN
08h	FIFO_SEQ_STATUS	07h	SEQ_ACTIVE	SEQ_COUNT[3:0]				FIFO_OFn	FIFO_UFn	FIFO_CRC_FAULTn
09h	FIFO_DEPTH_MSB	00h	RESERVED							FIFO_DEPTH[8]
0Ah	FIFO_DEPTH_LSB	00h	FIFO_DEPTH[7:0]							
10h	CONVERSION_CTRL	00h	START	STEP_INIT[4:0]					RESERVED	STOP
11h	RESET	00h	RESET_CODE[7:0]							
12h	ADC_CFG	0Ch	RESERVED	FIFO_TEST_EN	BUF_MODE[1:0]		SPEED_MODE[1:0]		STBY_MODE	PWDN
13h	REFERENCE_CFG	01h	RESERVED						REF_VAL	REFP_BUF_EN
14h	CLK_DIGITAL_CFG	04h	RESERVED		CLK_DIV[1:0]		CLK_SEL	OUT_DRV	SDO_MODE	CONT_READ_EN
15h	AGPIO_CFG0	00h	AGPIO3_CFG[1:0]		AGPIO2_CFG[1:0]		AGPIO1_CFG[1:0]		AGPIO0_CFG[1:0]	
16h	AGPIO_CFG1	00h	AGPIO7_CFG[1:0]		AGPIO6_CFG[1:0]		AGPIO5_CFG[1:0]		AGPIO4_CFG[1:0]	
17h	GPIO_CFG	0Ch	GPIO3_CFG[1:0]		GPIO2_CFG[1:0]		GPIO1_CFG[1:0]		GPIO0_CFG[1:0]	
18h	SPARE_CFG	00h	SPARE7	SPARE6	SPARE5	SPARE4	SPARE3	SPARE2	SPARE1	SPARE0
20h	SEQUENCER_CFG	40h	SEQ_MODE[1:0]		STOP_BEHAVIOR[1:0]		RESERVED		DRDY_CFG[1:0]	
21h	SEQUENCE_STEP_EN_0	01h	SEQ_STEP_7_EN	SEQ_STEP_6_EN	SEQ_STEP_5_EN	SEQ_STEP_4_EN	SEQ_STEP_3_EN	SEQ_STEP_2_EN	SEQ_STEP_1_EN	SEQ_STEP_0_EN
22h	SEQUENCE_STEP_EN_1	00h	SEQ_STEP_15_EN	SEQ_STEP_14_EN	SEQ_STEP_13_EN	SEQ_STEP_12_EN	SEQ_STEP_11_EN	SEQ_STEP_10_EN	SEQ_STEP_9_EN	SEQ_STEP_8_EN
23h	SEQUENCE_STEP_EN_2	00h	SEQ_STEP_23_EN	SEQ_STEP_22_EN	SEQ_STEP_21_EN	SEQ_STEP_20_EN	SEQ_STEP_19_EN	SEQ_STEP_18_EN	SEQ_STEP_17_EN	SEQ_STEP_16_EN
24h	SEQUENCE_STEP_EN_3	00h	SEQ_STEP_31_EN	SEQ_STEP_30_EN	SEQ_STEP_29_EN	SEQ_STEP_28_EN	SEQ_STEP_27_EN	SEQ_STEP_26_EN	SEQ_STEP_25_EN	SEQ_STEP_24_EN
25h	FIFO_CFG	00h	RESERVED							FIFO_EN
26h	FIFO_THRES_A_MSB	00h	RESERVED							FIFO_THRES_A[8]
27h	FIFO_THRES_A_LSB	00h	FIFO_THRES_A[7:0]							
28h	FIFO_THRES_B_MSB	00h	RESERVED							FIFO_THRES_B[8]
29h	FIFO_THRES_B_LSB	00h	FIFO_THRES_B[7:0]							
2Ah	DIAG_MONITOR_CFG	20h	RESERVED	TDAC_RANGE	FAULT_PIN_BEHAVIOR	REG_MAP_CRC_EN	RESERVED	REF_UV_EN	STATUS_EN	SPI_CRC_EN
2Bh	POSTFILTER_CFG0	00h	RESERVED					PF_AVG[1:0]		PF_CFG
2Ch	POSTFILTER_CFG1	00h	PF7_EN	PF6_EN	PF5_EN	PF4_EN	PF3_EN	PF2_EN	PF1_EN	PF0_EN
2Dh	POSTFILTER_CFG2	FFh	PF7_BYPASS	PF6_BYPASS	PF5_BYPASS	PF4_BYPASS	PF3_BYPASS	PF2_BYPASS	PF1_BYPASS	PF0_BYPASS
30h	CS_FWD_CFG	00h	CS_FWD_EN_CODE[5:0]						TIMEOUT_SEL[1:0]	
31h	AGPIO_FWD_CFG	00h	AGPIO7_FWD_EN	AGPIO6_FWD_EN	AGPIO5__FWD_EN	AGPIO4_FWD_EN	AGPIO3_FWD_EN	AGPIO2_FWD_EN	AGPIO1_FWD_EN	AGPIO0_FWD_EN

Table 7-50. Register Map (continued)

Address	Acronym	Reset	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
32h	GPIO_FWD_CFG	00h	RESERVED				GPIO3_FWD_EN	GPIO2_FWD_EN	GPIO1__FWD_EN	GPIO0_FWD_EN
3Dh	REG_MAP_CRC	00h	GENERAL_CFG_REG_MAP_CRC_VALUE[7:0]							
3Eh	PAGE_INDICATOR	00h	PAGE_INDICATOR[7:0]							
3Fh	PAGE_POINTER	00h	PAGE_POINTER[7:0]							

Complex bit access types are encoded to fit into small table cells. Table 7-51 shows the codes that are used for access types in this section.

Table 7-51. ADS125P08 Status and General Configuration Page Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
Write Type		
W	W	Write
Reset or Default Value		
-*n*		Value after reset or the default value

7.6.1.1 DEVICE_ID Register (Address = 00h) [Reset = 30h]

Return to the Summary Table.

Figure 7-75. DEVICE_ID Register

7	6	5	4	3	2	1	0
RESERVED		CH_CNT[1:0]		DEV_ID[3:0]			
R-00b		R-11b		R-0000b			

Table 7-52. DEVICE_ID Register Field Descriptions

Bit	Field	Type	Reset	Description
7:6	RESERVED	R	00b	Reserved Always reads 00b
5:4	CH_CNT[1:0]	R	11b	Channel count Always reads 11b
3:0	DEV_ID[3:0]	R	0000b	Device ID Register Values are subject to change without notice.

7.6.1.2 REVISION_ID Register (Address = 01h) [Reset = 01h]

Return to the Summary Table.

Figure 7-76. REVISION_ID Register

<table>
<tr><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td colspan="8">REV_ID[7:0]</td></tr>
<tr><td colspan="8">R-00000001b</td></tr>
</table>

Table 7-53. REVISION_ID Register Field Descriptions

Bit	Field	Type	Reset	Description
7:0	REV_ID[7:0]	R	00000001b	Revision ID Values are subject to change without notice.

7.6.1.3 STATUS_MSB Register (Address = 02h) [Reset = 00h]

Return to the Summary Table.

Figure 7-77. STATUS_MSB Register

7	6	5	4	3	2	1	0
STEP_INDICATOR[4:0]					ADC_REF_FAULTn	RESETn	DRDY
R-00000b					R-0b	R/W-0b	R-0b

Table 7-54. STATUS_MSB Register Field Descriptions

Bit	Field	Type	Reset	Description
7:3	STEP_INDICATOR[4:0]	R	00000b	Sequence step indicator Indicates the step page configuration that was used for the latest conversion result, which is currently available for readout. The step indicator resets to 00h after a device reset, in powerdown mode, or when writing to the SEQUENCER_CFG register. At the same time the conversion counter (CONV_COUNT[3:0]) resets to Fh, the sequence counter (SEQ_COUNT[3:0]) resets to 0h, and the conversion data clears.
2	ADC_REF_FAULTn	R	0b	ADC or reference out of range fault flag This bit updates when any individual bit in ADC_REF_STATUS is set. For this bit to clear, all bits in ADC_REF_STATUS must be cleared. This bit indicates either rail-to-rail buffer out of range, or modulator overrange, or reference undervoltage. 0b = Out of range fault occurred 1b = No out of range fault occurred
1	RESETn	R/W	0b	Reset flag Indicates a device reset occurred. Write 1b to clear bit to 1b. 0b = Reset occurred 1b = No reset occurred
0	DRDY	R	0b	Data-ready flag DRDY indicates when new conversion data are ready. The DRDY bit is the inverse of the DRDYn pin. Poll the bit to determine if conversion data are new or are repeated data from the last read operation. 0b = Data are not new 1b = Data are new

7.6.1.4 STATUS_LSB Register (Address = 03h) [Reset = FFh]

Return to the Summary Table.

Figure 7-78. STATUS_LSB Register

7	6	5	4	3	2	1	0
CONV_COUNT[3:0]				FIFO_FAULTn	INTERNAL_FAULTn	REG_WRITE_FAULTn	SPI_CRC_FAULTn
R-1111b				R-1b	R-1b	R-1b	R-1b

Table 7-55. STATUS_LSB Register Field Descriptions

Bit	Field	Type	Reset	Description
7:4	CONV_COUNT[3:0]	R	1111b	Conversion counter The conversion counter increments every time a new conversion completes. After reaching a counter value of Fh, the counter rolls over to 0h with the completion of the next conversion. The counter only resets to Fh in powerdown mode, after a device reset, or when writing to the SEQUENCER_CFG register. At the same time the sequence step indicator (STEP_INDICATOR[4:0]) resets to 00h, the sequence counter (SEQ_COUNT[3:0]) resets to 0h, and the conversion data clears. At the completion of the first conversion after reset, powerdown, or after writing to the SEQUENCER_CFG register, the counter reads 0h. When the sequencer is enabled (SEQ_MODE[1:0] = 10b or 11b), the counter always reads 0h for the first conversion of a step. When the sequencer is disabled (SEQ_MODE[1:0] = 00b or 01b), the counter value does not return to 0h if conversions with a new step page configuration complete. Reset the counter to Fh by writing to the SEQUENCER_CFG register before starting a conversion with a new step page configuration if desired.
3	FIFO_FAULTn	R	1b	FIFO fault flag This bit updates when any individual FIFO error bit in the FIFO_SEQ_STATUS register is set. For this bit to clear, all bits in FIFO_SEQ_STATUS must be cleared. 0b = FIFO fault occurred 1b = No FIFO fault occurred
2	INTERNAL_FAULTn	R	1b	Internal fault flag This bit updates when any individual bit in DIGITAL_STATUS is set. For this bit to clear, all bits in DIGITAL_STATUS must be cleared. 0b = Internal fault occurred 1b = No internal fault occurred
1	REG_WRITE_FAULTn	R	1b	Page or register access fault flag Indicates a write access to an invalid register address occurred. This flag sets as soon as an invalid register address is written to, and resets at the beginning of the next SPI frame. Reading from an invalid register address does not set the flag, but can be detected from the address indication inside the SPI frame of the read command. 0b = Page or register access fault occurred 1b = No page or register access fault occurred
0	SPI_CRC_FAULTn	R	1b	SPI CRC fault flag Indicates a SPI CRC fault occurred in the previous SPI frame. Bit clears automatically to 1b in every new SPI frame. 0b = SPI CRC fault occurred 1b = No SPI CRC fault occurred

7.6.1.5 ADC_REF_STATUS Register (Address = 04h) [Reset = B0h]

Return to the Summary Table.

Figure 7-79. ADC_REF_STATUS Register

7	6	5	4	3	2	1	0
RESERVED	AVDD_UVn	REF_UVn	MOD_OVR_FAULTn	RESERVED			
R-1b	R/W-0b	R/W-1b	R/W-1b	R-0000b			

Table 7-56. ADC_REF_STATUS Register Field Descriptions

Bit	Field	Type	Reset	Description
7	RESERVED	R	1b	Reserved Always reads 1b.
6	AVDD_UVn	R/W	0b	Analog supply voltage undervoltage fault flag Indicates the AVDD supply voltage dropped below the supply undervoltage threshold. Write 1b to clear bit to 1b. 0b = Supply undervoltage fault occurred 1b = No supply undervoltage fault occurred
5	REF_UVn	R/W	1b	Reference voltage undervoltage fault flag Indicates the reference voltage selected by the REF_SEL[1:0] bits dropped below the reference undervoltage threshold. Write 1b to clear bit to 1b. 0b = Reference undervoltage fault occurred 1b = No reference undervoltage fault occurred
4	MOD_OVR_FAULTn	R/W	1b	Modulator overrange fault indicator Write 1b to clear bit to 1b. 0b = Modulator overrange fault occurred 1b = No modulator overrange fault occurred
3:0	RESERVED	R	0000b	Reserved Always reads 0000b

7.6.1.6 DIGITAL_STATUS Register (Address = 05h) [Reset = FFh]

Return to the Summary Table.

Figure 7-80. DIGITAL_STATUS Register

7	6	5	4	3	2	1	0
CRC_FAULT_PAGE[5:0]						MEM_INTERNAL_FAULTn	REG_MAP_CRC_FAULTn
R-111111b						R/W-1b	R/W-1b

Table 7-57. DIGITAL_STATUS Register Field Descriptions

Bit	Field	Type	Reset	Description
7:2	CRC_FAULT_PAGE[5:0]	R	111111b	Register map CRC fault page indicator Indicates which register page shows a CRC error when the REG_MAP_CRC_FAULTn indicates a CRC fault. If multiple register pages have a CRC error, then the indicator points to the first register page address where a CRC error exists. When the CRC error on the page which was indicated by the CRC_FAULT_PAGE[5:0] bit field is corrected by providing a correct register map CRC value, and another CRC error on another register page exists, the CRC_FAULT_PAGE[5:0] bit field does not update automatically. After writing 1b to the REG_MAP_CRC_FAULTn bit field, the REG_MAP_CRC_FAULTn sets to 0b again, and the CRC_FAULT_PAGE[5:0] bit field points to the next remaining first page address which has a register map CRC error. This bit field clears to 111111b when the REG_MAP_CRC_FAULTn flag is cleared to 1b.
1	MEM_INTERNAL_FAULTn	R/W	1b	Internal memory fault flag Indicates a memory map CRC fault in the internal memory occurred or a wrong page was selected internally (PAGE_INDICATOR does not match PAGE_POINTER). Write 1b to clear bit to 1b. 0b = Memory map CRC fault occurred 1b = No memory map CRC fault occurred
0	REG_MAP_CRC_FAULTn	R/W	1b	Register map CRC fault flag Indicates a register map CRC fault in the general configuration page (register address space from 12h to 32h) or in the step configuration page occurred. Write 1b to clear bit to 1b. 0b = Register map CRC fault occurred 1b = No register map CRC fault occurred

7.6.1.7 AGPIO_DATA_INPUT Register (Address = 06h) [Reset = 00h]

Return to the Summary Table.

Figure 7-81. AGPIO_DATA_INPUT Register

7	6	5	4	3	2	1	0
AGPIO7_DAT_IN	AGPIO6_DAT_IN	AGPIO5_DAT_IN	AGPIO4_DAT_IN	AGPIO3_DAT_IN	AGPIO2_DAT_IN	AGPIO1_DAT_IN	AGPIO0_DAT_IN
R-0b	R-0b	R-0b	R-0b	R-0b	R-0b	R-0b	R-0b

Table 7-58. AGPIO_DATA_INPUT Register Field Descriptions

Bit	Field	Type	Reset	Description
7	AGPIO7_DAT_IN	R	0b	AGPIO7 data Read back value of AGPIO7 when configured as digital input or output. 0b = Low 1b = High
6	AGPIO6_DAT_IN	R	0b	AGPIO6 data Read back value of AGPIO6 when configured as digital input or output. 0b = Low 1b = High
5	AGPIO5_DAT_IN	R	0b	AGPIO5 data Read back value of AGPIO5 when configured as digital input or output. 0b = Low 1b = High
4	AGPIO4_DAT_IN	R	0b	AGPIO4 data Read back value of AGPIO4 when configured as digital input or output. 0b = Low 1b = High
3	AGPIO3_DAT_IN	R	0b	AGPIO3 data Read back value of AGPIO3 when configured as digital input or output. 0b = Low 1b = High
2	AGPIO2_DAT_IN	R	0b	AGPIO2 data Read back value of AGPIO2 when configured as digital input or output. 0b = Low 1b = High
1	AGPIO1_DAT_IN	R	0b	AGPIO1 data Read back value of AGPIO1 when configured as digital input or output. 0b = Low 1b = High
0	AGPIO0_DAT_IN	R	0b	AGPIO0 data Read back value of AGPIO0 when configured as digital input or output. 0b = Low 1b = High

7.6.1.8 GPIO_DATA_INPUT Register (Address = 07h) [Reset = 02h]

Return to the Summary Table.

Figure 7-82. GPIO_DATA_INPUT Register

7	6	5	4	3	2	1	0
RESERVED				GPIO3_DAT_IN	GPIO2_DAT_IN	GPIO1_DAT_IN	GPIO0_DAT_IN
R-0000b				R-0b	R-0b	R-1b	R-0b

Table 7-59. GPIO_DATA_INPUT Register Field Descriptions

Bit	Field	Type	Reset	Description
7:4	RESERVED	R	0000b	Reserved Always reads 0000b
3	GPIO3_DAT_IN	R	0b	GPIO3 data Read back value of GPIO3 when configured as digital input or output. Reads back 0b when GPIO3_CFG[1:0] = 00b. 0b = Low 1b = High
2	GPIO2_DAT_IN	R	0b	GPIO2 data Read back value of GPIO2 when configured as digital input or output. Reads back 0b when GPIO2_CFG[1:0] = 00b or 11b. 0b = Low 1b = High
1	GPIO1_DAT_IN	R	1b	GPIO1 data Read back value of GPIO1 when configured as digital input or output. Reads back 0b when GPIO1_CFG[1:0] = 00b. 0b = Low 1b = High
0	GPIO0_DAT_IN	R	0b	GPIO0 data Read back value of GPIO0 when configured as digital input or output. Reads back 0b when GPIO0_CFG[1:0] = 00b. 0b = Low 1b = High

7.6.1.9 FIFO_SEQ_STATUS Register (Address = 08h) [Reset = 07h]

Return to the Summary Table.

Figure 7-83. FIFO_SEQ_STATUS Register

7	6	5	4	3	2	1	0
SEQ_ACTIVE	SEQ_COUNT[3:0]				FIFO_OFn	FIFO_UFn	FIFO_CRC_FAULTn
R-0b	R-0000b				R/W-1b	R/W-1b	R-1b

Table 7-60. FIFO_SEQ_STATUS Register Field Descriptions

Bit	Field	Type	Reset	Description
7	SEQ_ACTIVE	R	0b	Sequencer active flag Indicates if conversions are currently ongoing or if conversions stopped and the device is in idle, standby or powerdown mode. 0b = Sequencer not active 1b = Sequencer active
6:3	SEQ_COUNT[3:0]	R	0000b	Completed sequence counter When SEQ_MODE[1:0] = 11b, the sequence counter indicates which sequence run the latest conversion result belongs to, which is currently available for readout. The sequence counter increments with the completion of the first conversion of a new sequence run. At the completion of the first conversion of the first sequence run, the counter reads 0h. At the completion of the first conversion of the second sequence run, the counter reads 1h. After reaching a counter value of Fh, the counter rolls over to 0h with the completion of the first conversion of the next sequence run. The counter resets to 0h at the completion of the first conversion after setting the START bit to 1b or at the rising edge of the START pin. When writing to the SEQUENCER_CFG register, in powerdown mode, or after a device reset, the counter resets to 0h immediately. At the same time the sequence step indicator (STEP_INDICATOR[4:0]) resets to 00h, the conversion counter (CONV_COUNT[3:0]) resets to Fh, and the conversion data clears. The sequence counter always reads 0h when SEQ_MODE[1:0] = 00b, 01b, or 10b.
2	FIFO_OFn	R/W	1b	FIFO overflow flag Indicates a FIFO overflow fault occurred. Write 1b to clear bit to 1b. 0b = FIFO overflow occurred 1b = No FIFO overflow occurred
1	FIFO_UFn	R/W	1b	FIFO underflow flag Indicates a FIFO underflow fault occurred. Write 1b to clear bit to 1b. 0b = FIFO underflow occurred 1b = No FIFO underflow occurred
0	FIFO_CRC_FAULTn	R	1b	FIFO CRC fault flag Indicates a FIFO CRC fault occurred. Write 1b to clear bit to 1b. 0b = FIFO CRC fault occurred 1b = No FIFO CRC fault occurred

7.6.1.10 FIFO_DEPTH_MSB Register (Address = 09h) [Reset = 00h]

Return to the Summary Table.

Figure 7-84. FIFO_DEPTH_MSB Register

7	6	5	4	3	2	1	0
RESERVED							FIFO_DEPTH[8]
R-0000000b							R-0b

Table 7-61. FIFO_DEPTH_MSB Register Field Descriptions

Bit	Field	Type	Reset	Description
7:1	RESERVED	R	0000000b	Reserved Always reads 00b
0	FIFO_DEPTH[8]	R	0b	FIFO depth indicator MSB MSB bit of the FIFO DEPTH indicator.

7.6.1.11 FIFO_DEPTH_LSB Register (Address = 0Ah) [Reset = 00h]

Return to the Summary Table.

Figure 7-85. FIFO_DEPTH_LSB Register

7	6	5	4	3	2	1	0
FIFO_DEPTH[7:0]							
R-00000000b							

Table 7-62. FIFO_DEPTH_LSB Register Field Descriptions

Bit	Field	Type	Reset	Description
7:0	FIFO_DEPTH[7:0]	R	00000000b	FIFO depth indicator LSB LSB bit of the FIFO DEPTH indicator.

7.6.1.12 CONVERSION_CTRL Register (Address = 10h) [Reset = 00h]

Return to the Summary Table.

Figure 7-86. CONVERSION_CTRL Register

7	6	5	4	3	2	1	0
START	STEP_INIT[4:0]					RESERVED	STOP
R/W-0b	R/W-00000b					R-0b	R/W-0b

Table 7-63. CONVERSION_CTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
7	START	R/W	0b	Start or re-start conversions of the ADC Write 1b to start or re-start conversions of the ADC. In one-shot control mode (SEQ_MODE = 00b), one conversion is started. In start/stop control mode (SEQ_MODE=01b), conversions are started and continue until stopped by the STOP bit. Writing 1b to START while a conversion is ongoing restarts the conversion. In modes where the sequencer is enabled (SEQ_MODE=10b or 11b), the sequence is started with the step indicated by STEP_INIT[4:0]. Writing 1b to both the START and STOP bits has no effect. START is self-clearing and always reads 0b. 0b = No operation 1b = Start or restart conversions
6:2	STEP_INIT[4:0]	R/W	00000b	Initial execution step selector Defines the sequence step which is executed first when a sequence is started.
1	RESERVED	R	0b	Reserved Always reads 0b
0	STOP	R/W	0b	Stop conversions of the ADC Write 1b to stop conversions after the current conversion completes. Writing 1b to both the START and STOP has no effect. STOP is self-clearing and always reads 0b. The STOP bit clears to 0b after the ongoing sequence finished or when the START bit is set before the ongoing sequence finished, which aborts the ongoing sequence and re-starts a new sequence. 0b = No operation 1b = Stop conversions

7.6.1.13 RESET Register (Address = 11h) [Reset = 00h]

Return to the Summary Table.

Figure 7-87. RESET Register

<table>
<tr><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td colspan="8">RESET_CODE[7:0]</td></tr>
<tr><td colspan="8">R/W-00000000b</td></tr>
</table>

Table 7-64. RESET Register Field Descriptions

Bit	Field	Type	Reset	Description
7:0	RESET_CODE[7:0]	R/W	00000000b	Device reset register Write 01011010b to reset the ADC. These bits always read 00000000b.

7.6.1.14 ADC_CFG Register (Address = 12h) [Reset = 0Ch]

Return to the Summary Table.

Figure 7-88. ADC_CFG Register

7	6	5	4	3	2	1	0
RESERVED	FIFO_TEST_EN	BUF_MODE[1:0]		SPEED_MODE[1:0]		STBY_MODE	PWDN
R-0b	R/W-0b	R/W-00b		R/W-11b		R/W-0b	R/W-0b

Table 7-65. ADC_CFG Register Field Descriptions

Bit	Field	Type	Reset	Description
7	RESERVED	R	0b	Reserved Always reads 0b
6	FIFO_TEST_EN	R/W	0b	ADC test mode enable (ADC counter enable) When this test mode is enabled, the ADC data value is incremented by 1 for each data output. 0b = Disabled 1b = Enabled
5:4	BUF_MODE[1:0]	R/W	00b	Rail-to-rail (HiZ) input buffer selection See the rail-to-rail buffer section for more details. 00b = Both HiZ input buffers off 01b = Only P HiZ input buffer turned on without level shift 10b = Both P and N HiZ input buffers turned on with level shift 11b = Both P and N HiZ input buffers turned on without level shift
3:2	SPEED_MODE[1:0]	R/W	11b	Power mode selection Selects the power mode. 00b = Very-low-speed mode: 0.8MHz mod clock mode 01b = Low-speed mode: 1.6MHz mod clock mode 10b = Mid-speed mode: 6.4MHz mod clock mode 11b = High-speed mode: 12.8MHz mod clock mode
1	STBY_MODE	R/W	0b	ADC standby mode selection Selects the ADC mode when conversions stop. 0b = Idle mode; ADC remains fully powered when conversions are stopped; the configuration from previous sequence step is still active. 1b = Standby mode; ADC powers down when conversions are stopped. Standby mode is exited when conversions restart.
0	PWDN	R/W	0b	Powerdown mode selection Powers down all circuitry except for the digital LDO to retain all user register settings. SPI communication is still possible. In powerdown mode, the step indicator (STEP_INDICATOR[4:0]) resets to 00h, the conversion counter (CONV_COUNT[3:0]) resets to Fh, the sequence counter (SEQ_COUNT[3:0]) resets to 0h, the conversion data clears, the FIFO clears, and the START bit and START pin are ignored. Setting the PWDN bit to 1b powers the device down immediately; any ongoing conversions are aborted. Any analog inputs configured as GPIO digital outputs transition into a Hi-Z state in powerdown mode. To maintain a certain logic level during powerdown, consider external pullup or pulldown resistors on the respective GPIO pins. 0b = Active 1b = Powerdown mode

7.6.1.15 REFERENCE_CFG Register (Address = 13h) [Reset = 01h]

Return to the Summary Table.

Figure 7-89. REFERENCE_CFG Register

7	6	5	4	3	2	1	0
RESERVED						REF_VAL	REFP_BUF_EN
R-000000b						R/W-0b	R/W-1b

Table 7-66. REFERENCE_CFG Register Field Descriptions

Bit	Field	Type	Reset	Description
7:2	RESERVED	R	000000b	Reserved Always reads 000000b
1	REF_VAL	R/W	0b	Internal reference value selection Selects internal reference value. 0b = Internal ADC reference value is 2.5 V 1b = Internal ADC reference value is 4.096 V
0	REFP_BUF_EN	R/W	1b	Positive reference buffer enable Enables the positive reference buffer. 0b = Disabled 1b = Enabled

7.6.1.16 CLK_DIGITAL_CFG Register (Address = 14h) [Reset = 04h]

Return to the Summary Table.

Figure 7-90. CLK_DIGITAL_CFG Register

7	6	5	4	3	2	1	0
RESERVED		CLK_DIV[1:0]		CLK_SEL	OUT_DRV	SDO_MODE	CONT_READ_EN
R-00b		R/W-00b		R/W-0b	R/W-1b	R/W-0b	R/W-0b

Table 7-67. CLK_DIGITAL_CFG Register Field Descriptions

Bit	Field	Type	Reset	Description
7:6	RESERVED	R	00b	Reserved Always reads 00b
5:4	CLK_DIV[1:0]	R/W	00b	External clock divider ratio selection Selects clock divider ratio. 00b = Divide by 1 01b = Divide by 2 10b = Divide by 8 11b = Divide by 16
3	CLK_SEL	R/W	0b	Clock selection Selects the clock source for the device. 0b = Internal oscillator 1b = External clock
2	OUT_DRV	R/W	1b	Digital output drive selection Selects the drive strength of the digital outputs. 0b = Full-drive strength 1b = Half-drive strength
1	SDO_MODE	R/W	0b	SDO pin mode selection This bit programs the mode of the SDO/DRDY pin. 0b = Data-output only mode 1b = Dual mode: data output and data ready
0	CONT_READ_EN	R/W	0b	Continuous read mode enable Allows read of multiple bytes (conversion or register data) without CS transition. 0b = Continuous read mode disabled 1b = Continuous read mode enabled (daisy-chain not available)

7.6.1.17 AGPIO_CFG0 Register (Address = 15h) [Reset = 00h]

Return to the Summary Table.

Figure 7-91. AGPIO_CFG0 Register

7	6	5	4	3	2	1	0
AGPIO3_CFG[1:0]		AGPIO2_CFG[1:0]		AGPIO1_CFG[1:0]		AGPIO0_CFG[1:0]	
R/W-00b		R/W-00b		R/W-00b		R/W-00b	

Table 7-68. AGPIO_CFG0 Register Field Descriptions

Bit	Field	Type	Reset	Description
7:6	AGPIO3_CFG[1:0]	R/W	00b	AGPIO3 configuration Configures the AGPIO3 pin behavior. 00b = Pin operates as analog input 01b = Digital input 10b = Push-pull digital output (with readback) 11b = Open-drain digital output (with readback)
5:4	AGPIO2_CFG[1:0]	R/W	00b	AGPIO2 configuration Configures the AGPIO2 pin behavior. 00b = Pin operates as analog input 01b = Digital input 10b = Push-pull digital output (with readback) 11b = Open-drain digital output (with readback)
3:2	AGPIO1_CFG[1:0]	R/W	00b	AGPIO1 configuration Configures the AGPIO1 pin behavior. 00b = Pin operates as analog input 01b = Digital input 10b = Push-pull digital output (with readback) 11b = Open-drain digital output (with readback)
1:0	AGPIO0_CFG[1:0]	R/W	00b	AGPIO0 configuration Configures the AGPIO0 pin behavior. 00b = Pin operates as analog input 01b = Digital input 10b = Push-pull digital output (with readback) 11b = Open-drain digital output (with readback)

7.6.1.18 AGPIO_CFG1 Register (Address = 16h) [Reset = 00h]

Return to the Summary Table.

Figure 7-92. AGPIO_CFG1 Register

7	6	5	4	3	2	1	0
AGPIO7_CFG[1:0]		AGPIO6_CFG[1:0]		AGPIO5_CFG[1:0]		AGPIO4_CFG[1:0]	
R/W-00b		R/W-00b		R/W-00b		R/W-00b	

Table 7-69. AGPIO_CFG1 Register Field Descriptions

Bit	Field	Type	Reset	Description
7:6	AGPIO7_CFG[1:0]	R/W	00b	AGPIO7 configuration Configures the AGPIO7 pin behavior. 00b = Pin operates as analog input 01b = Digital input 10b = Push-pull digital output (with readback) 11b = Open-drain digital output (with readback)
5:4	AGPIO6_CFG[1:0]	R/W	00b	AGPIO6 configuration Configures the AGPIO6 pin behavior. 00b = Pin operates as analog input 01b = Digital input 10b = Push-pull digital output (with readback) 11b = Open-drain digital output (with readback)
3:2	AGPIO5_CFG[1:0]	R/W	00b	AGPIO5 configuration Configures the AGPIO5 pin behavior. 00b = Pin operates as analog input 01b = Digital input 10b = Push-pull digital output (with readback) 11b = Open-drain digital output (with readback)
1:0	AGPIO4_CFG[1:0]	R/W	00b	AGPIO4 configuration Configures the AGPIO4 pin behavior. 00b = Pin operates as analog input 01b = Digital input 10b = Push-pull digital output (with readback) 11b = Open-drain digital output (with readback)

7.6.1.19 GPIO_CFG Register (Address = 17h) [Reset = 0Ch]

Return to the Summary Table.

Figure 7-93. GPIO_CFG Register

7	6	5	4	3	2	1	0
GPIO3_CFG[1:0]		GPIO2_CFG[1:0]		GPIO1_CFG[1:0]		GPIO0_CFG[1:0]	
R/W-00b		R/W-00b		R/W-11b		R/W-00b	

Table 7-70. GPIO_CFG Register Field Descriptions

Bit	Field	Type	Reset	Description
7:6	GPIO3_CFG[1:0]	R/W	00b	GPIO3 configuration Configures the GPIO3 pin behavior. 00b = Disabled (High-Z) 01b = Digital input 10b = Push-pull digital output (with readback) 11b = Pin operates as FAULTn output (with readback)
5:4	GPIO2_CFG[1:0]	R/W	00b	GPIO2 configuration Configures the GPIO2 pin behavior. 00b = Disabled (High-Z) 01b = Digital input 10b = Push-pull digital output (with readback) 11b = Pin operates as external clock (CLK) input. Set CLK_SEL = 1b to select the external clock for device operation.
3:2	GPIO1_CFG[1:0]	R/W	11b	GPIO1 configuration Configures the GPIO1 pin behavior. 00b = Disabled (High-Z) 01b = Digital input 10b = Push-pull digital output (with readback) 11b = Pin operates as DRDYn output (with readback)
1:0	GPIO0_CFG[1:0]	R/W	00b	GPIO0 configuration Configures the GPIO0 pin behavior. 00b = Disabled (High-Z) 01b = Digital input 10b = Push-pull digital output (with readback) 11b = Pin operates as START/SYNC input (with readback)

7.6.1.20 SPARE_CFG Register (Address = 18h) [Reset = 00h]

Return to the Summary Table.

Figure 7-94. SPARE_CFG Register

7	6	5	4	3	2	1	0
SPARE7	SPARE6	SPARE5	SPARE4	SPARE3	SPARE2	SPARE1	SPARE0
R/W-0b	R/W-0b	R/W-0b	R/W-0b	R/W-0b	R/W-0b	R/W-0b	R/W-0b

Table 7-71. SPARE_CFG Register Field Descriptions

Bit	Field	Type	Reset	Description
7	SPARE7	R/W	0b	Spare register bit For user functions or CRC checking. 0b = Spare is programmed to 0b 1b = Spare is programmed to 1b
6	SPARE6	R/W	0b	Spare register bit For user functions or CRC checking. 0b = Spare is programmed to 0b 1b = Spare is programmed to 1b
5	SPARE5	R/W	0b	Spare register bit For user functions or CRC checking. 0b = Spare is programmed to 0b 1b = Spare is programmed to 1b
4	SPARE4	R/W	0b	Spare register bit For user functions or CRC checking. 0b = Spare is programmed to 0b 1b = Spare is programmed to 1b
3	SPARE3	R/W	0b	Spare register bit For user functions or CRC checking. 0b = Spare is programmed to 0b 1b = Spare is programmed to 1b
2	SPARE2	R/W	0b	Spare register bit For user functions or CRC checking. 0b = Spare is programmed to 0b 1b = Spare is programmed to 1b
1	SPARE1	R/W	0b	Spare register bit For user functions or CRC checking. 0b = Spare is programmed to 0b 1b = Spare is programmed to 1b
0	SPARE0	R/W	0b	Spare register bit For user functions or CRC checking. 0b = Spare is programmed to 0b 1b = Spare is programmed to 1b

7.6.1.21 SEQUENCER_CFG Register (Address = 20h) [Reset = 40h]

Return to the Summary Table.

Figure 7-95. SEQUENCER_CFG Register

7	6	5	4	3	2	1	0
SEQ_MODE[1:0]		STOP_BEHAVIOR[1:0]		RESERVED		DRDY_CFG[1:0]	
R/W-01b		R/W-00b		R-00b		R/W-00b	

Table 7-72. SEQUENCER_CFG Register Field Descriptions

Bit	Field	Type	Reset	Description
7:6	SEQ_MODE[1:0]	R/W	01b	Sequencer execution mode selection Selects the sequencer execution mode. 00b = Sequencer disabled: The sequence step defined by the STEP_INIT[4:0] pointer in the CONVERSION_CTRL register is executed only once (single-shot operation). Step enable bits are ignored in this operational mode. STEPx_NUM_CONV[3:0] bits determine the number of conversions for this sequence step. 01b = Sequencer disabled: The sequence step defined by the STEP_INIT[4:0] pointer in the CONVERSION_CTRL register is executed and repeated indefinitely (continuous conversion operation). Step enable bits and STEPx_NUM_CONV[3:0] bits are ignored in this operational mode. 10b = Sequencer enabled: Execute the complete sequence of steps once, starting with the step defined by the STEP_INIT[4:0] pointer. If the step defined by the STEP_INIT[4:0] pointer is not enabled, then the sequence is not executed. 11b = Sequencer enabled: Execute the complete sequence of steps and repeat continuously, starting with the step defined by the STEP_INIT[4:0] pointer. If the step defined by the STEP_INIT[4:0] pointer is not enabled, then the sequence is not executed.
5:4	STOP_BEHAVIOR[1:0]	R/W	00b	Sequence stop behavior selection These bits define both the operation of the stop bit as well falling edge of START pin. 00b = Stop immediately. The currrent conversion is not completed. 01b = Stop after current conversion is completed. 10b = Stop after current sequence step is completed. If SEQ_MODE=01b while in this mode, then stop after currrent conversion is completed. 11b = Stop after full sequence is completed. If SEQ_MODE=00b while in this mode, then stop after current sequence step is completed. If SEQ_MODE=01b while in this mode, then stop after current conversion is completed.
3:2	RESERVED	R	00b	Reserved Always reads 00b
1:0	DRDY_CFG[1:0]	R/W	00b	DRDY operational mode selection Selects the DRDY operation mode. 00b = DRDY transitions after every completed conversion. 01b = DRDY transitions after every completed sequence step. 10b = DRDY transitions after every completed sequence. 11b = DRDY transitions are defined by two thresholds related to the FIFO buffer depth and defined in the FIFO_THRES_HI and FIFO_THRES_LO registers.

7.6.1.22 SEQUENCE_STEP_EN_0 Register (Address = 21h) [Reset = 01h]

Return to the Summary Table.

Figure 7-96. SEQUENCE_STEP_EN_0 Register

7	6	5	4	3	2	1	0
SEQ_STEP_7_EN	SEQ_STEP_6_EN	SEQ_STEP_5_EN	SEQ_STEP_4_EN	SEQ_STEP_3_EN	SEQ_STEP_2_EN	SEQ_STEP_1_EN	SEQ_STEP_0_EN
R/W-0b	R/W-0b	R/W-0b	R/W-0b	R/W-0b	R/W-0b	R/W-0b	R/W-1b

Table 7-73. SEQUENCE_STEP_EN_0 Register Field Descriptions

Bit	Field	Type	Reset	Description
7	SEQ_STEP_7_EN	R/W	0b	Sequencer step enable Enables sequence step 7. 0b = Step is disabled. 1b = Step is enabled.
6	SEQ_STEP_6_EN	R/W	0b	Sequencer step enable Enables sequence step 6. 0b = Step is disabled. 1b = Step is enabled.
5	SEQ_STEP_5_EN	R/W	0b	Sequencer step enable Enables sequence step 5. 0b = Step is disabled. 1b = Step is enabled.
4	SEQ_STEP_4_EN	R/W	0b	Sequencer step enable Enables sequence step 4. 0b = Step is disabled. 1b = Step is enabled.
3	SEQ_STEP_3_EN	R/W	0b	Sequencer step enable Enables sequence step 3. 0b = Step is disabled. 1b = Step is enabled.
2	SEQ_STEP_2_EN	R/W	0b	Sequencer step enable Enables sequence step 2. 0b = Step is disabled. 1b = Step is enabled.
1	SEQ_STEP_1_EN	R/W	0b	Sequencer step enable Enables sequence step 1. 0b = Step is disabled. 1b = Step is enabled.
0	SEQ_STEP_0_EN	R/W	1b	Sequencer step enable Enables sequence step 0. 0b = Step is disabled. 1b = Step is enabled.

7.6.1.23 SEQUENCE_STEP_EN_1 Register (Address = 22h) [Reset = 00h]

Return to the Summary Table.

Figure 7-97. SEQUENCE_STEP_EN_1 Register

7	6	5	4	3	2	1	0
SEQ_STEP_15_EN	SEQ_STEP_14_EN	SEQ_STEP_13_EN	SEQ_STEP_12_EN	SEQ_STEP_11_EN	SEQ_STEP_10_EN	SEQ_STEP_9_EN	SEQ_STEP_8_EN
R/W-0b	R/W-0b	R/W-0b	R/W-0b	R/W-0b	R/W-0b	R/W-0b	R/W-0b

Table 7-74. SEQUENCE_STEP_EN_1 Register Field Descriptions

Bit	Field	Type	Reset	Description
7	SEQ_STEP_15_EN	R/W	0b	Sequencer step enable Enables sequence step 15. 0b = Step is disabled. 1b = Step is enabled.
6	SEQ_STEP_14_EN	R/W	0b	Sequencer step enable Enables sequence step 14. 0b = Step is disabled. 1b = Step is enabled.
5	SEQ_STEP_13_EN	R/W	0b	Sequencer step enable Enables sequence step 13. 0b = Step is disabled. 1b = Step is enabled.
4	SEQ_STEP_12_EN	R/W	0b	Sequencer step enable Enables sequence step 12. 0b = Step is disabled. 1b = Step is enabled.
3	SEQ_STEP_11_EN	R/W	0b	Sequencer step enable Enables sequence step 11. 0b = Step is disabled. 1b = Step is enabled.
2	SEQ_STEP_10_EN	R/W	0b	Sequencer step enable Enables sequence step 10. 0b = Step is disabled. 1b = Step is enabled.
1	SEQ_STEP_9_EN	R/W	0b	Sequencer step enable Enables sequence step 9. 0b = Step is disabled. 1b = Step is enabled.
0	SEQ_STEP_8_EN	R/W	0b	Sequencer step enable Enables sequence step 8. 0b = Step is disabled. 1b = Step is enabled.

7.6.1.24 SEQUENCE_STEP_EN_2 Register (Address = 23h) [Reset = 00h]

Return to the Summary Table.

Figure 7-98. SEQUENCE_STEP_EN_2 Register

7	6	5	4	3	2	1	0
SEQ_STEP_23_EN	SEQ_STEP_22_EN	SEQ_STEP_21_EN	SEQ_STEP_20_EN	SEQ_STEP_19_EN	SEQ_STEP_18_EN	SEQ_STEP_17_EN	SEQ_STEP_16_EN
R/W-0b	R/W-0b	R/W-0b	R/W-0b	R/W-0b	R/W-0b	R/W-0b	R/W-0b

Table 7-75. SEQUENCE_STEP_EN_2 Register Field Descriptions

Bit	Field	Type	Reset	Description
7	SEQ_STEP_23_EN	R/W	0b	Sequencer step enable Enables sequence step 23. 0b = Step is disabled. 1b = Step is enabled.
6	SEQ_STEP_22_EN	R/W	0b	Sequencer step enable Enables sequence step 22. 0b = Step is disabled. 1b = Step is enabled.
5	SEQ_STEP_21_EN	R/W	0b	Sequencer step enable Enables sequence step 21. 0b = Step is disabled. 1b = Step is enabled.
4	SEQ_STEP_20_EN	R/W	0b	Sequencer step enable Enables sequence step 20. 0b = Step is disabled. 1b = Step is enabled.
3	SEQ_STEP_19_EN	R/W	0b	Sequencer step enable Enables sequence step 19. 0b = Step is disabled. 1b = Step is enabled.
2	SEQ_STEP_18_EN	R/W	0b	Sequencer step enable Enables sequence step 18. 0b = Step is disabled. 1b = Step is enabled.
1	SEQ_STEP_17_EN	R/W	0b	Sequencer step enable Enables sequence step 17. 0b = Step is disabled. 1b = Step is enabled.
0	SEQ_STEP_16_EN	R/W	0b	Sequencer step enable Enables sequence step 16. 0b = Step is disabled. 1b = Step is enabled.

7.6.1.25 SEQUENCE_STEP_EN_3 Register (Address = 24h) [Reset = 00h]

Return to the Summary Table.

Figure 7-99. SEQUENCE_STEP_EN_3 Register

7	6	5	4	3	2	1	0
SEQ_STEP_31_EN	SEQ_STEP_30_EN	SEQ_STEP_29_EN	SEQ_STEP_28_EN	SEQ_STEP_27_EN	SEQ_STEP_26_EN	SEQ_STEP_25_EN	SEQ_STEP_24_EN
R/W-0b	R/W-0b	R/W-0b	R/W-0b	R/W-0b	R/W-0b	R/W-0b	R/W-0b

Table 7-76. SEQUENCE_STEP_EN_3 Register Field Descriptions

Bit	Field	Type	Reset	Description
7	SEQ_STEP_31_EN	R/W	0b	Sequencer step enable Enables sequence step 31. 0b = Step is disabled. 1b = Step is enabled.
6	SEQ_STEP_30_EN	R/W	0b	Sequencer step enable Enables sequence step 30. 0b = Step is disabled. 1b = Step is enabled.
5	SEQ_STEP_29_EN	R/W	0b	Sequencer step enable Enables sequence step 29. 0b = Step is disabled. 1b = Step is enabled.
4	SEQ_STEP_28_EN	R/W	0b	Sequencer step enable Enables sequence step 28. 0b = Step is disabled. 1b = Step is enabled.
3	SEQ_STEP_27_EN	R/W	0b	Sequencer step enable Enables sequence step 27. 0b = Step is disabled. 1b = Step is enabled.
2	SEQ_STEP_26_EN	R/W	0b	Sequencer step enable Enables sequence step 26. 0b = Step is disabled. 1b = Step is enabled.
1	SEQ_STEP_25_EN	R/W	0b	Sequencer step enable Enables sequence step 25. 0b = Step is disabled. 1b = Step is enabled.
0	SEQ_STEP_24_EN	R/W	0b	Sequencer step enable Enables sequence step 24. 0b = Step is disabled. 1b = Step is enabled.

7.6.1.26 FIFO_CFG Register (Address = 25h) [Reset = 00h]

Return to the Summary Table.

Figure 7-100. FIFO_CFG Register

7	6	5	4	3	2	1	0
RESERVED							FIFO_EN
R-0000000b							R/W-0b

Table 7-77. FIFO_CFG Register Field Descriptions

Bit	Field	Type	Reset	Description
7:1	RESERVED	R	0000000b	Reserved Always reads 0000000b
0	FIFO_EN	R/W	0b	FIFO enable Enables the FIFO. 0b = FIFO is disabled. (FIFO is flushed and write pointer and read pointer reset, when disabled) 1b = FIFO is enabled.

7.6.1.27 FIFO_THRES_A_MSB Register (Address = 26h) [Reset = 00h]

Return to the Summary Table.

Figure 7-101. FIFO_THRES_A_MSB Register

7	6	5	4	3	2	1	0
RESERVED							FIFO_THRES_A[8]
R-0000000b							R/W-0b

Table 7-78. FIFO_THRES_A_MSB Register Field Descriptions

Bit	Field	Type	Reset	Description
7:1	RESERVED	R	0000000b	Reserved Always reads 0000000b
0	FIFO_THRES_A[8]	R/W	0b	DRDY FIFO threshold A configuration MSB MSB bit of the FIFO threshold A. Upper Threshold for DRDY transition, when sequencer is in threshold mode set by DRDY_CFG=11b. Once the FIFO_DEPTH[8:0] indicator reaches the upper threshold, DRDY will transition low.

7.6.1.28 FIFO_THRES_A_LSB Register (Address = 27h) [Reset = 00h]

Return to the Summary Table.

Figure 7-102. FIFO_THRES_A_LSB Register

7	6	5	4	3	2	1	0
FIFO_THRES_A[7:0]							
R/W-00000000b							

Table 7-79. FIFO_THRES_A_LSB Register Field Descriptions

Bit	Field	Type	Reset	Description
7:0	FIFO_THRES_A[7:0]	R/W	00000000b	DRDY FIFO threshold A configuration LSB LSB bits of the FIFO threshold A. Upper Threshold for DRDY transition, when sequencer is in threshold mode set by DRDY_CFG=11b. Once the FIFO_DEPTH[8:0] indicator reaches the upper threshold, DRDY will transition low.

7.6.1.29 FIFO_THRES_B_MSB Register (Address = 28h) [Reset = 00h]

Return to the Summary Table.

Figure 7-103. FIFO_THRES_B_MSB Register

7	6	5	4	3	2	1	0
RESERVED							FIFO_THRES_B[8]
R-0000000b							R/W-0b

Table 7-80. FIFO_THRES_B_MSB Register Field Descriptions

Bit	Field	Type	Reset	Description
7:1	RESERVED	R	0000000b	Reserved Always reads 0000000b
0	FIFO_THRES_B[8]	R/W	0b	DRDY FIFO threshold B configuration MSB MSB bit of the FIFO threshold B. Lower Threshold for DRDY transition, when sequencer is in threshold mode set by DRDY_CFG=11b. Once the FIFO_DEPTH[8:0] indicator reaches the lower threshold, DRDY will transition high.

7.6.1.30 FIFO_THRES_B_LSB Register (Address = 29h) [Reset = 00h]

Return to the Summary Table.

Figure 7-104. FIFO_THRES_B_LSB Register

7	6	5	4	3	2	1	0
FIFO_THRES_B[7:0]							
R/W-00000000b							

Table 7-81. FIFO_THRES_B_LSB Register Field Descriptions

Bit	Field	Type	Reset	Description
7:0	FIFO_THRES_B[7:0]	R/W	00000000b	DRDY FIFO threshold B configuration LSB LSB bits of the FIFO threshold B. Lower Threshold for DRDY transition, when sequencer is in threshold mode set by DRDY_CFG=11b. Once the FIFO_DEPTH[8:0] indicator reaches the lower threshold, DRDY will transition high.

7.6.1.31 DIAG_MONITOR_CFG Register (Address = 2Ah) [Reset = 20h]

Return to the Summary Table.

Figure 7-105. DIAG_MONITOR_CFG Register

7	6	5	4	3	2	1	0
RESERVED	TDAC_RANGE	FAULT_PIN_BEHAVIOR	REG_MAP_CRC_EN	RESERVED	REF_UV_EN	STATUS_EN	SPI_CRC_EN
R-0b	R/W-0b	R/W-1b	R/W-0b	R-0b	R/W-0b	R/W-0b	R/W-0b

Table 7-82. DIAG_MONITOR_CFG Register Field Descriptions

Bit	Field	Type	Reset	Description
7	RESERVED	R	0b	Reserved Always reads 0b
6	TDAC_RANGE	R/W	0b	Test DAC range selection Select test DAC range. 0b = TDAC range is 2.5 V 1b = TDAC range is 4.096 V
5	FAULT_PIN_BEHAVIOR	R/W	1b	Fault pin behavior selection Selects fault pin behavior. 0b = Fault pin output signal is static: Pin is high when no fault. Pin is low when fault occurs. 1b = Fault pin output signal is dynamic: Pin is 50/50 duty cycle signal at f#MOD#/256 when no fault. Pin low when fault.
4	REG_MAP_CRC_EN	R/W	0b	Register map CRC enable Enables the register map CRC for the General Configuration page (register addresses 12h to 32h) as well as Step Configuration page (register addresses 0h to 10h). 0b = Disabled 1b = Enabled (all step configuration pages are CRC checked)
3	RESERVED	R	0b	Reserved Always reads 0b
2	REF_UV_EN	R/W	0b	Reference monitor enable Enables the reference monitor. 0b = Reference monitor disabled 1b = Reference monitor enabled
1	STATUS_EN	R/W	0b	STATUS byte output enable Enables the STATUS byte(s) transmission on SDO as the first 2 bytes of every SPI frame. 0b = Disabled 1b = Enabled
0	SPI_CRC_EN	R/W	0b	SPI CRC enable Enables the SPI CRC on SDI and SDO. 0b = Disabled 1b = Enabled

7.6.1.32 POSTFILTER_CFG0 Register (Address = 2Bh) [Reset = 00h]

Return to the Summary Table.

Figure 7-106. POSTFILTER_CFG0 Register

7	6	5	4	3	2	1	0
RESERVED					PF_AVG[1:0]		PF_CFG
R-00000b					R/W-00b		R/W-0b

Table 7-83. POSTFILTER_CFG0 Register Field Descriptions

Bit	Field	Type	Reset	Description
7:3	RESERVED	R	00000b	Reserved Always reads 00000b
2:1	PF_AVG[1:0]	R/W	00b	Post filter average number selection Number of averages for the digital per-channel post filters. this field will be ignored if SEQ_MODE[1:0] = 00b or 01b (sequencer disabled). 00b = average 4 01b = average 8 10b = average 16 11b = average 16
0	PF_CFG	R/W	0b	Post filter cascading number selection Cascading option for digital post filters. 0b = filter not cascaded (similar to a sinc1) 1b = filter cascaded 3x (similar to a sinc3)

7.6.1.33 POSTFILTER_CFG1 Register (Address = 2Ch) [Reset = 00h]

Return to the Summary Table.

Figure 7-107. POSTFILTER_CFG1 Register

7	6	5	4	3	2	1	0
PF7_EN	PF6_EN	PF5_EN	PF4_EN	PF3_EN	PF2_EN	PF1_EN	PF0_EN
R/W-0b	R/W-0b	R/W-0b	R/W-0b	R/W-0b	R/W-0b	R/W-0b	R/W-0b

Table 7-84. POSTFILTER_CFG1 Register Field Descriptions

Bit	Field	Type	Reset	Description
7	PF7_EN	R/W	0b	Per channel post filter enable Enable for post filter 7 (tied to sequencer step 7). This bit is ignored if SEQ_MODE[1:0] = 00b or 01b (sequencer disabled). If postfilter is enabled, then the number of conversions per sequencer step is forced to 1, i.e. STEPx_NUM_CONV[3:0]=0. Only step 0 to step 7 are valid to be used if any post-filter is active. All other steps will be ignored. 0b = disabled 1b = enabled
6	PF6_EN	R/W	0b	Per channel post filter enable Enable for post filter 6 (tied to sequencer step 6). This bit is ignored if SEQ_MODE[1:0] = 00b or 01b (sequencer disabled). If postfilter is enabled, then the number of conversions per sequencer step is forced to 1, i.e. STEPx_NUM_CONV[3:0]=0. Only step 0 to step 7 are valid to be used if any post-filter is active. All other steps will be ignored. 0b = disabled 1b = enabled
5	PF5_EN	R/W	0b	Per channel post filter enable Enable for post filter 5 (tied to sequencer step 5). This bit is ignored if SEQ_MODE[1:0] = 00b or 01b (sequencer disabled). If postfilter is enabled, then the number of conversions per sequencer step is forced to 1, i.e. STEPx_NUM_CONV[3:0]=0. Only step 0 to step 7 are valid to be used if any post-filter is active. All other steps will be ignored. 0b = disabled 1b = enabled
4	PF4_EN	R/W	0b	Per channel post filter enable Enable for post filter 4 (tied to sequencer step 4). This bit is ignored if SEQ_MODE[1:0] = 00b or 01b (sequencer disabled). If postfilter is enabled, then the number of conversions per sequencer step is forced to 1, i.e. STEPx_NUM_CONV[3:0]=0. Only step 0 to step 7 are valid to be used if any post-filter is active. All other steps will be ignored. 0b = disabled 1b = enabled
3	PF3_EN	R/W	0b	Per channel post filter enable Enable for post filter 3 (tied to sequencer step 3). This bit is ignored if SEQ_MODE[1:0] = 00b or 01b (sequencer disabled). If postfilter is enabled, then the number of conversions per sequencer step is forced to 1, i.e. STEPx_NUM_CONV[3:0]=0. Only step 0 to step 7 are valid to be used if any post-filter is active. All other steps will be ignored. 0b = disabled 1b = enabled
2	PF2_EN	R/W	0b	Per channel post filter enable Enable for post filter 2 (tied to sequencer step 2). This bit is ignored if SEQ_MODE[1:0] = 00b or 01b (sequencer disabled). If postfilter is enabled, then the number of conversions per sequencer step is forced to 1, i.e. STEPx_NUM_CONV[3:0]=0. Only step 0 to step 7 are valid to be used if any post-filter is active. All other steps will be ignored. 0b = disabled 1b = enabled
1	PF1_EN	R/W	0b	Per channel post filter enable Enable for post filter 1 (tied to sequencer step 1). This bit is ignored if SEQ_MODE[1:0] = 00b or 01b (sequencer disabled). If postfilter is enabled, then the number of conversions per sequencer step is forced to 1, i.e. STEPx_NUM_CONV[3:0]=0. Only step 0 to step 7 are valid to be used if any post-filter is active. All other steps will be ignored. 0b = disabled 1b = enabled
0	PF0_EN	R/W	0b	Per channel post filter enable Enable for post filter 0 (tied to sequencer step 0). This bit is ignored if SEQ_MODE[1:0] = 00b or 01b (sequencer disabled). If postfilter is enabled, then the number of conversions per sequencer step is forced to 1, i.e. STEPx_NUM_CONV[3:0]=0. Only step 0 to step 7 are valid to be used if any post-filter is active. All other steps will be ignored. 0b = disabled 1b = enabled

7.6.1.34 POSTFILTER_CFG2 Register (Address = 2Dh) [Reset = FFh]

Return to the Summary Table.

Figure 7-108. POSTFILTER_CFG2 Register

7	6	5	4	3	2	1	0
PF7_BYPASS	PF6_BYPASS	PF5_BYPASS	PF4_BYPASS	PF3_BYPASS	PF2_BYPASS	PF1_BYPASS	PF0_BYPASS
R/W-1b	R/W-1b	R/W-1b	R/W-1b	R/W-1b	R/W-1b	R/W-1b	R/W-1b

Table 7-85. POSTFILTER_CFG2 Register Field Descriptions

Bit	Field	Type	Reset	Description
7	PF7_BYPASS	R/W	1b	Per channel post filter bypass Bypass for post filter 7 (tied to sequencer step 7). 0b = post filtered data is provided to the output 1b = post filter is bypassed, data is provided directly from sync4 filter 0b = post filtered data is provided to the output 1b = post filter is bypassed, data is provided directly from sync4 filter
6	PF6_BYPASS	R/W	1b	Per channel post filter bypass Bypass for post filter 6 (tied to sequencer step 6) In bypass mode, the data is provided directly from sync4 filter to the output. 0b = post filtered data is provided to the output 1b = post filter is bypassed, data is provided directly from sync4 filter
5	PF5_BYPASS	R/W	1b	Per channel post filter bypass Bypass for post filter 5 (tied to sequencer step5). 0b = post filtered data is provided to the output 1b = post filter is bypassed, data is provided directly from sync4 filter 0b = post filtered data is provided to the output 1b = post filter is bypassed, data is provided directly from sync4 filter
4	PF4_BYPASS	R/W	1b	Per channel post filter bypass Bypass for post filter 4(tied to sequencer step 4). 0b = post filtered data is provided to the output 1b = post filter is bypassed, data is provided directly from sync4 filter 0b = post filtered data is provided to the output 1b = post filter is bypassed, data is provided directly from sync4 filter
3	PF3_BYPASS	R/W	1b	Per channel post filter bypass Bypass for post filter 3 (tied to sequencer step 3) In bypass mode, the data is provided directly from sync4 filter to the output. 0b = post filtered data is provided to the output 1b = post filter is bypassed, data is provided directly from sync4 filter
2	PF2_BYPASS	R/W	1b	Per channel post filter bypass Bypass for post filter 2 (tied to sequencer step 2). 0b = post filtered data is provided to the output 1b = post filter is bypassed, data is provided directly from sync4 filter 0b = post filtered data is provided to the output 1b = post filter is bypassed, data is provided directly from sync4 filter
1	PF1_BYPASS	R/W	1b	Per channel post filter bypass Bypass for post filter 0 (tied to sequencer step 0). 0b = post filtered data is provided to the output 1b = post filter is bypassed, data is provided directly from sync4 filter 0b = post filtered data is provided to the output 1b = post filter is bypassed, data is provided directly from sync4 filter
0	PF0_BYPASS	R/W	1b	Per channel post filter bypass Bypass for post filter 1 (tied to sequencer step 1). 0b = post filtered data is provided to the output 1b = post filter is bypassed, data is provided directly from sync4 filter 0b = post filtered data is provided to the output 1b = post filter is bypassed, data is provided directly from sync4 filter

7.6.1.35 CS_FWD_CFG Register (Address = 30h) [Reset = 00h]

Return to the Summary Table.

Figure 7-109. CS_FWD_CFG Register

7	6	5	4	3	2	1	0
CS_FWD_EN_CODE[5:0]						TIMEOUT_SEL[1:0]	
R/W-000000b						R/W-00b	

Table 7-86. CS_FWD_CFG Register Field Descriptions

Bit	Field	Type	Reset	Description
7:2	CS_FWD_EN_CODE[5:0]	R/W	000000b	CS Forward feature enable Write 010111b to enable the CS Forward feature. The AGPIOx_FWD_EN and GPIOx_FWD_EN bits select which GPIO pins operate in CS forwarding mode. These bits always read 00000000b.
1:0	TIMEOUT_SEL[1:0]	R/W	00b	Timeout enable and duration selection Enables the SPI timeout and sets the timeout duration. When enabled the timeout checks that a rising edge of CSn happens within the selected number of MCLK cycles after a CSn falling edge. When a timeout occurs, the rest of the SPI frame on SDI is ignored before the rising edge of CSn. A new SPI transaction will start at the next CSn falling edge. 00b = Timeout disable 01b = Timeout enable with the short timeout, 256 MCLK cycles 10b = Timeout enable with the medium length timeout, 2048 MCLK cycles 11b = Timeout enable with the long timeout, 16384 MCLK cycles

7.6.1.36 AGPIO_FWD_CFG Register (Address = 31h) [Reset = 00h]

Return to the Summary Table.

Figure 7-110. AGPIO_FWD_CFG Register

7	6	5	4	3	2	1	0
AGPIO7_FWD_EN	AGPIO6_FWD_EN	AGPIO5__FWD_EN	AGPIO4_FWD_EN	AGPIO3_FWD_EN	AGPIO2_FWD_EN	AGPIO1_FWD_EN	AGPIO0_FWD_EN
R/W-0b	R/W-0b	R/W-0b	R/W-0b	R/W-0b	R/W-0b	R/W-0b	R/W-0b

Table 7-87. AGPIO_FWD_CFG Register Field Descriptions

Bit	Field	Type	Reset	Description
7	AGPIO7_FWD_EN	R/W	0b	CS forward pin enable AGPIO7 CS forward configuration. Configures the AGPIO7 pin as a CS forward / output pin. If this bit is high, the AGPIO7_CFG[1:0] bits in the AGPIO_CFG registers are ignored. 0b = AGPIO7 is not configured as CS forward. 1b = AGPIO7 is configured as CS forward.
6	AGPIO6_FWD_EN	R/W	0b	CS forward pin enable AGPIO6 CS forward configuration. Configures the AGPIO6 pin as a CS forward / output pin. If this bit is high, the AGPIO6_CFG[1:0] bits in the AGPIO_CFG registers are ignored. 0b = AGPIO6 is not configured as CS forward. 1b = AGPIO6 is configured as CS forward.
5	AGPIO5__FWD_EN	R/W	0b	CS forward pin enable AGPIO5 CS forward configuration. Configures the AGPIO5 pin as a CS forward / output pin. If this bit is high, the AGPIO5_CFG[1:0] bits in the AGPIO_CFG registers are ignored. 0b = AGPIO5 is not configured as CS forward. 1b = AGPIO5 is configured as CS forward.
4	AGPIO4_FWD_EN	R/W	0b	CS forward pin enable AGPIO4 CS forward configuration. Configures the AGPIO4 pin as a CS forward / output pin. If this bit is high, the AGPIO4_CFG[1:0] bits in the AGPIO_CFG registers are ignored. 0b = AGPIO4 is not configured as CS forward. 1b = AGPIO4 is configured as CS forward.
3	AGPIO3_FWD_EN	R/W	0b	CS forward pin enable AGPIO3 CS forward configuration. Configures the AGPIO3 pin as a CS forward / output pin. If this bit is high, the AGPIO3_CFG[1:0] bits in the AGPIO_CFG registers are ignored. 0b = AGPIO3 is not configured as CS forward. 1b = AGPIO3 is configured as CS forward.
2	AGPIO2_FWD_EN	R/W	0b	CS forward pin enable AGPIO2 CS forward configuration. Configures the AGPIO2 pin as a CS forward / output pin. If this bit is high, the AGPIO2_CFG[1:0] bits in the AGPIO_CFG registers are ignored. 0b = AGPIO2 is not configured as CS forward. 1b = AGPIO2 is configured as CS forward.
1	AGPIO1_FWD_EN	R/W	0b	CS forward pin enable AGPIO1 CS forward configuration. Configures the AGPIO1 pin as a CS forward / output pin. If this bit is high, the AGPIO1_CFG[1:0] bits in the AGPIO_CFG registers are ignored. 0b = AGPIO1 is not configured as CS forward. 1b = AGPIO1 is configured as CS forward.
0	AGPIO0_FWD_EN	R/W	0b	CS forward pin enable AGPIO0 CS forward configuration. Configures the AGPIO0 pin as a CS forward / output pin. If this bit is high, the AGPIO0_CFG[1:0] bits in the AGPIO_CFG registers are ignored. 0b = AGPIO0 is not configured as CS forward. 1b = AGPIO0 is configured as CS forward.

7.6.1.37 GPIO_FWD_CFG Register (Address = 32h) [Reset = 00h]

Return to the Summary Table.

Figure 7-111. GPIO_FWD_CFG Register

7	6	5	4	3	2	1	0
RESERVED				GPIO3_FWD_EN	GPIO2_FWD_EN	GPIO1__FWD_EN	GPIO0_FWD_EN
R-0000b				R/W-0b	R/W-0b	R/W-0b	R/W-0b

Table 7-88. GPIO_FWD_CFG Register Field Descriptions

Bit	Field	Type	Reset	Description
7:4	RESERVED	R	0000b	Reserved Always reads 0000b
3	GPIO3_FWD_EN	R/W	0b	CS forward pin enable GPIO3 CS forward configuration. Configures the GPIO3 pin as a CS forward / output pin. If this bit is high, the GPIO3_CFG[1:0] bits in the GPIO_CFG registers are ignored. 0b = GPIO3 is not configured as CS forward. 1b = GPIO3 is configured as CS forward.
2	GPIO2_FWD_EN	R/W	0b	CS forward pin enable GPIO2 CS forward configuration. Configures the GPIO2 pin as a CS forward / output pin. If this bit is high, the GPIO2_CFG[1:0] bits in the GPIO_CFG registers are ignored. 0b = GPIO2 is not configured as CS forward. 1b = GPIO2 is configured as CS forward.
1	GPIO1__FWD_EN	R/W	0b	CS forward pin enable GPIO1 CS forward configuration. Configures the GPIO1 pin as a CS forward / output pin. If this bit is high, the GPIO1_CFG[1:0] bits in the GPIO_CFG registers are ignored. 0b = GPIO1 is not configured as CS forward. 1b = GPIO1 is configured as CS forward.
0	GPIO0_FWD_EN	R/W	0b	CS forward pin enable GPIO0 CS forward configuration. Configures the GPIO0 pin as a CS forward / output pin. If this bit is high, the GPIO0_CFG[1:0] bits in the GPIO_CFG registers are ignored. 0b = GPIO0 is not configured as CS forward. 1b = GPIO0 is configured as CS forward.

7.6.1.38 REG_MAP_CRC Register (Address = 3Dh) [Reset = 00h]

Return to the Summary Table.

Figure 7-112. REG_MAP_CRC Register

7	6	5	4	3	2	1	0
GENERAL_CFG_REG_MAP_CRC_VALUE[7:0]							
R/W-00000000b							

Table 7-89. REG_MAP_CRC Register Field Descriptions

Bit	Field	Type	Reset	Description
7:0	GENERAL_CFG_REG_MAP_CRC_VALUE[7:0]	R/W	00000000b	Register map CRC for General Configuration Page Register map CRC value The register map CRC value is the user-computed CRC value of registers 0x12 to 0x32 in the general configuration page. The CRC value written to this register is compared to an internal CRC calculation. If the values do not match, the REG_MAP_CRC_FAULTn bit is set. Enable the register map CRC using the REG_MAP_CRC_EN bit.

7.6.1.39 PAGE_INDICATOR Register (Address = 3Eh) [Reset = 00h]

Return to the Summary Table.

Figure 7-113. PAGE_INDICATOR Register

7	6	5	4	3	2	1	0
PAGE_INDICATOR[7:0]							
R-00000000b							

Table 7-90. PAGE_INDICATOR Register Field Descriptions

Bit	Field	Type	Reset	Description
7:0	PAGE_INDICATOR[7:0]	R	00000000b	Register page Indicator Indicates the active register page.

7.6.1.40 PAGE_POINTER Register (Address = 3Fh) [Reset = 00h]

Return to the Summary Table.

Figure 7-114. PAGE_POINTER Register

7	6	5	4	3	2	1	0
PAGE_POINTER[7:0]							
R/W-00000000b							

Table 7-91. PAGE_POINTER Register Field Descriptions

Bit	Field	Type	Reset	Description
7:0	PAGE_POINTER[7:0]	R/W	00000000b	Register page pointer Selects the active register page.

7.6.2 ADS125P08 Step Configuration Page

Table 7-92 lists the memory-mapped registers for the ADS125P08 Step Configuration Page registers. All register offset addresses not listed in Table 7-92 should be considered as reserved locations and the register contents should not be modified.

Table 7-92. Register Map

Address	Acronym	Reset	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
00h	STEPx_AINP_CFG	00h	RESERVED			STEPx_AINP[4:0]				
01h	STEPx_AINN_CFG	00h	RESERVED			STEPx_AINN[4:0]				
02h	STEPx_ADC_REF_CFG	00h	RESERVED	STEPx_EXT_RANG	STEPx_CODING	STEPx_REF_SEL	STEPx_NUM_CONV[3:0]			
03h	STEPx_FLTR1_CFG	01h	RESERVED		STEPx_FLTR_MODE	STEPx_FLTR_OSR[4:0]				
04h	STEPx_DELAY_MSB_CFG	00h	STEPx_DELAY_MSB[7:0]							
05h	STEPx_DELAY_LSB_CFG	00h	STEPx_DELAY_LSB[7:0]							
06h	STEPx_OFFSET_CAL_MSB	00h	STEPx_OFFSET_CAL[23:16]							
07h	STEPx_OFFSET_CAL_ISB	00h	STEPx_OFFSET_CAL[15:8]							
08h	STEPx_OFFSET_CAL_LSB	00h	STEPx_OFFSET_CAL[7:0]							
09h	STEPx_GAIN_CAL_MSB	40h	STEPx_GAIN_CAL[15:8]							
0Ah	STEPx_GAIN_CAL_LSB	00h	STEPx_GAIN_CAL[7:0]							
0Bh	STEPx_OW_SYSMON_CFG	00h	RESERVED		STEPx_BOCS[1:0]		STEPx_SYS_MON[3:0]			
0Ch	STEPx_TDAC_CFG0	00h	RESERVED			STEPx_TDAC_VAL[4:0]				
0Dh	STEPx_TDAC_CFG1	00h	RESERVED			STEPx_TDAC_SEL[4:0]				
0Eh	STEPx_SPARE_CFG	00h	STEPx_SPARE7	STEPx_SPARE6	STEPx_SPARE5	STEPx_SPARE4	STEPx_SPARE3	STEPx_SPARE2	STEPx_SPARE1	STEPx_SPARE0
0Fh	STEPx_AGPIO_DATA_OUT	00h	STEPx_AGPIO7_DAT_OUT	STEPx_AGPIO6_DAT_OUT	STEPx_AGPIO5_DAT_OUT	STEPx_AGPIO4_DAT_OUT	STEPx_AGPIO3_DAT_OUT	STEPx_AGPIO2_DAT_OUT	STEPx_AGPIO1_DAT_OUT	STEPx_AGPIO0_DAT_OUT
10h	STEPx_GPIO_DATA_OUT	00h	RESERVED				STEPx_GPIO3_DAT_OUT	STEPx_GPIO2_DAT_OUT	STEPx_GPIO1_DAT_OUT	STEPx_GPIO0_DAT_OUT
3Dh	STEPx_REG_MAP_CRC	00h	STEPx_REG_MAP_CRC_VALUE[7:0]							
3Eh	STEPx_PAGE_INDICATOR	00h	STEPx_PAGE_INDICATOR[7:0]							
3Fh	STEPx_PAGE_POINTER	00h	STEPx_PAGE_POINTER[7:0]							

Complex bit access types are encoded to fit into small table cells. Table 7-93 shows the codes that are used for access types in this section.

Table 7-93. ADS125P08 Step Configuration Page Access Type Codes

Access Type	Code	Description
Read Type		
R	R	Read
Write Type		
W	W	Write
Reset or Default Value		
-n		Value after reset or the default value

7.6.2.1 STEPx_AINP_CFG Register (Address = 00h) [Reset = 00h]

Return to the Summary Table.

Figure 7-115. STEPx_AINP_CFG Register

<table>
<tr><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td colspan="3">RESERVED</td><td colspan="5">STEPx_AINP[4:0]</td></tr>
<tr><td colspan="3">R-000b</td><td colspan="5">R/W-00000b</td></tr>
</table>

Table 7-94. STEPx_AINP_CFG Register Field Descriptions

Bit	Field	Type	Reset	Description
7:5	RESERVED	R	000b	Reserved Always reads 000b
4:0	STEPx_AINP[4:0]	R/W	00000b	Positive multiplexer input selection Selects the positive analog input for the ADC. This register is ignored if system monitors are active. 00000b = AIN0 00001b = AIN1 00010b = AIN2 00011b = AIN3 00100b = AIN4 00101b = AIN5 00110b = AIN6 00111b = AIN7 01000b = AIN8 01001b = AIN9 01010b = AIN10 01011b = AIN11 01100b = AIN12 01101b = AIN13 01110b = AIN14 01111b = AIN15 10000b = AINCOM 10001b = Open 10010b = Open 10011b = Open 10100b = Open 10101b = Open 10110b = Open 10111b = Open 11000b = Open 11001b = Open 11010b = Open 11011b = Open 11100b = Open 11101b = Open 11110b = Open 11111b = Open

7.6.2.2 STEPx_AINN_CFG Register (Address = 01h) [Reset = 00h]

Return to the Summary Table.

Figure 7-116. STEPx_AINN_CFG Register

7	6	5	4	3	2	1	0
RESERVED			STEPx_AINN[4:0]				
R-000b			R/W-00000b				

Table 7-95. STEPx_AINN_CFG Register Field Descriptions

Bit	Field	Type	Reset	Description
7:5	RESERVED	R	000b	Reserved Always reads 000b
4:0	STEPx_AINN[4:0]	R/W	00000b	Negative multiplexer input selection Selects the negative analog input for the ADC. This register is ignored if system monitors are active. 00000b = AIN0 00001b = AIN1 00010b = AIN2 00011b = AIN3 00100b = AIN4 00101b = AIN5 00110b = AIN6 00111b = AIN7 01000b = AIN8 01001b = AIN9 01010b = AIN10 01011b = AIN11 01100b = AIN12 01101b = AIN13 01110b = AIN14 01111b = AIN15 10000b = AINCOM 10001b = Open 10010b = Open 10011b = Open 10100b = Open 10101b = Open 10110b = Open 10111b = Open 11000b = Open 11001b = Open 11010b = Open 11011b = Open 11100b = Open 11101b = Open 11110b = Open 11111b = Open

7.6.2.3 STEPx_ADC_REF_CFG Register (Address = 02h) [Reset = 00h]

Return to the Summary Table.

Figure 7-117. STEPx_ADC_REF_CFG Register

7	6	5	4	3	2	1	0
RESERVED	STEPx_EXT_RNG	STEPx_CODING	STEPx_REF_SEL	STEPx_NUM_CONV[3:0]			
R-0b	R/W-0b	R/W-0b	R/W-0b	R/W-0000b			

Table 7-96. STEPx_ADC_REF_CFG Register Field Descriptions

Bit	Field	Type	Reset	Description
7	RESERVED	R	0b	Reserved Always reads 0b
6	STEPx_EXT_RNG	R/W	0b	Extended input range selection This bit extends the input range by 25%. See the Input Range Section for more details. 0b = Standard input range 1b = 25% extended input range
5	STEPx_CODING	R/W	0b	Conversion data coding selection Selects the coding of the conversion data. 0b = Bipolar, two's complement format 1b = Unipolar, straight binary format
4	STEPx_REF_SEL	R/W	0b	Reference voltage source selection 0b = External voltage reference (REFP, REFN) 1b = Internal voltage reference
3:0	STEPx_NUM_CONV[3:0]	R/W	0000b	Number of ADC conversions for this sequence step Up to 512 ADC conversions can be generated for each sequence step. This number can be programmed individually per step. 0000b = 1 conversion 0001b = 2 conversions 0010b = 3 conversions 0011b = 4 conversions 0100b = 6 conversions 0101b = 8 conversions 0110b = 10 conversions 0111b = 12 conversions 1000b = 14 conversions 1001b = 16 conversions 1010b = 24 conversions 1011b = 32 conversions 1100b = 64 conversions 1101b = 128 conversions 1110b = 256 conversions 1111b = 512 conversions

7.6.2.4 STEPx_FLTR1_CFG Register (Address = 03h) [Reset = 01h]

Return to the Summary Table.

Figure 7-118. STEPx_FLTR1_CFG Register

7	6	5	4	3	2	1	0
RESERVED		STEPx_FLTR_MODE	STEPx_FLTR_OSR[4:0]				
R-00b		R/W-0b	R/W-00001b				

Table 7-97. STEPx_FLTR1_CFG Register Field Descriptions

Bit	Field	Type	Reset	Description
7:6	RESERVED	R	00b	Reserved Always reads 00b
5	STEPx_FLTR_MODE	R/W	0b	Digital filter mode selection Select sinc3 or sinc4 first stage filter. 0b = sinc4 first stage filter 1b = sinc3 first stage filter
4:0	STEPx_FLTR_OSR[4:0]	R/W	00001b	Digital filter oversampling selection These bits select the combination of oversampling ratio and sinc filter operation. Sincx = sinc3 or sinc4 filter selection made by STEPx_FLTR_MODE bit. The output data rate is equal to f_{CLK}/2/OSR. 00000b = SINCx, OSR = 12 00001b = SINCx, OSR = 16 00010b = SINCx, OSR = 24 00011b = SINCx, OSR = 32 00100b = SINCx, OSR = 64 00101b = SINCx, OSR = 128 00110b = SINCx, OSR = 256 00111b = SINCx, OSR = 512 01000b = SINCx, OSR = 1024 01001b = SINCx, OSR = 2048 01010b = SINCx, OSR = 4000 01011b = SINCx, OSR = 8000 01100b = SINCx, OSR = 16000 01101b = SINCx, OSR = 26667 01110b = SINCx, OSR = 32000 01111b = SINCx, OSR = 96000 10000b = SINCx, OSR = 160000 10001b = SINC4, OSR = 32 + SINC1, OSR = 2 10010b = SINC4, OSR = 32 + SINC1, OSR = 4 10011b = SINC4, OSR = 32 + SINC1, OSR = 8 10100b = SINC4, OSR = 32 + SINC1, OSR = 16 10101b = SINC4, OSR = 32 + SINC1, OSR = 32 10110b = SINC4, OSR = 32 + SINC1, OSR = 64 10111b = SINC4, OSR = 32 + SINC1, OSR = 125 11000b = SINC4, OSR = 32 + SINC1, OSR = 250 11001b = SINC4, OSR = 32 + SINC1, OSR = 500 11010b = SINC4, OSR = 32 + SINC1, OSR = 833 11011b = SINC4, OSR = 32 + SINC1, OSR =1000 11100b = SINC4, OSR = 32 + SINC1, OSR =3000 11101b = SINC4, OSR = 32 + SINC1, OSR = 5000 11110b = SINC4, OSR = 32 + SINC1, OSR = 20 + 99 tap FIR, 25 SPS 11111b = SINC4, OSR = 32 + SINC1, OSR = 20 + 124 tap FIR, 20 SPS

7.6.2.5 STEPx_DELAY_MSB_CFG Register (Address = 04h) [Reset = 00h]

Return to the Summary Table.

Figure 7-119. STEPx_DELAY_MSB_CFG Register

7	6	5	4	3	2	1	0
STEPx_DELAY_MSB[7:0]							
R/W-00000000b							

Table 7-98. STEPx_DELAY_MSB_CFG Register Field Descriptions

Bit	Field	Type	Reset	Description
7:0	STEPx_DELAY_MSB[7:0]	R/W	00000000b	Conversion-start delay time selection, MSB Programmable delay time before the start of the first conversion when START is applied or a sequence step is started (MSB byte). Delay time is given in number of f_{MOD} clock cycles (f_{MOD} = f_{CLK} / 2). In total, this is a 16 bit register.

7.6.2.6 STEPx_DELAY_LSB_CFG Register (Address = 05h) [Reset = 00h]

Return to the Summary Table.

Figure 7-120. STEPx_DELAY_LSB_CFG Register

7	6	5	4	3	2	1	0
STEPx_DELAY_LSB[7:0]							
R/W-00000000b							

Table 7-99. STEPx_DELAY_LSB_CFG Register Field Descriptions

Bit	Field	Type	Reset	Description
7:0	STEPx_DELAY_LSB[7:0]	R/W	00000000b	Conversion-start delay time selection, LSB Programmable delay time before the start of the first conversion when START is applied or a sequence step is started (LSB byte). Delay time is given in number of f_{MOD} clock cycles (f_{MOD} = f_{CLK} / 2). In total, this is a 16 bit register.

7.6.2.7 STEPx_OFFSET_CAL_MSB Register (Address = 06h) [Reset = 00h]

Return to the Summary Table.

Figure 7-121. STEPx_OFFSET_CAL_MSB Register

7	6	5	4	3	2	1	0
STEPx_OFFSET_CAL[23:16]							
R/W-00000000b							

Table 7-100. STEPx_OFFSET_CAL_MSB Register Field Descriptions

Bit	Field	Type	Reset	Description
7:0	STEPx_OFFSET_CAL[23:16]	R/W	00000000b	Offset calibration coefficient, MSB Sets the offset calibration coefficient.

7.6.2.8 STEPx_OFFSET_CAL_ISB Register (Address = 07h) [Reset = 00h]

Return to the Summary Table.

Figure 7-122. STEPx_OFFSET_CAL_ISB Register

7	6	5	4	3	2	1	0
STEPx_OFFSET_CAL[15:8]							
R/W-00000000b							

Table 7-101. STEPx_OFFSET_CAL_ISB Register Field Descriptions

Bit	Field	Type	Reset	Description
7:0	STEPx_OFFSET_CAL[15:8]	R/W	00000000b	Offset calibration coefficient, ISB Sets the offset calibration coefficient.

7.6.2.9 STEPx_OFFSET_CAL_LSB Register (Address = 08h) [Reset = 00h]

Return to the Summary Table.

Figure 7-123. STEPx_OFFSET_CAL_LSB Register

7	6	5	4	3	2	1	0
STEPx_OFFSET_CAL[7:0]							
R/W-00000000b							

Table 7-102. STEPx_OFFSET_CAL_LSB Register Field Descriptions

Bit	Field	Type	Reset	Description
7:0	STEPx_OFFSET_CAL[7:0]	R/W	00000000b	Offset calibration coefficient, LSB Sets the offset calibration coefficient.

7.6.2.10 STEPx_GAIN_CAL_MSB Register (Address = 09h) [Reset = 40h]

Return to the Summary Table.

Figure 7-124. STEPx_GAIN_CAL_MSB Register

7	6	5	4	3	2	1	0
STEPx_GAIN_CAL[15:8]							
R/W-01000000b							

Table 7-103. STEPx_GAIN_CAL_MSB Register Field Descriptions

Bit	Field	Type	Reset	Description
7:0	STEPx_GAIN_CAL[15:8]	R/W	01000000b	Gain calibration coefficient, MSB Sets the gain calibration coefficient.

7.6.2.11 STEPx_GAIN_CAL_LSB Register (Address = 0Ah) [Reset = 00h]

Return to the Summary Table.

Figure 7-125. STEPx_GAIN_CAL_LSB Register

7	6	5	4	3	2	1	0
STEPx_GAIN_CAL[7:0]							
R/W-00000000b							

Table 7-104. STEPx_GAIN_CAL_LSB Register Field Descriptions

Bit	Field	Type	Reset	Description
7:0	STEPx_GAIN_CAL[7:0]	R/W	00000000b	Gain calibration coefficient, LSB Sets the gain calibration coefficient.

7.6.2.12 STEPx_OW_SYSMON_CFG Register (Address = 0Bh) [Reset = 00h]

Return to the Summary Table.

Figure 7-126. STEPx_OW_SYSMON_CFG Register

7	6	5	4	3	2	1	0
RESERVED	RESERVED	STEPx_BOCS[1:0]		STEPx_SYS_MON[3:0]			
R-0b	R-0b	R/W-00b		R/W-0000b			

Table 7-105. STEPx_OW_SYSMON_CFG Register Field Descriptions

Bit	Field	Type	Reset	Description
7	RESERVED	R	0b	Reserved Always reads 0b
6	RESERVED	R	0b	Reserved Always reads 0b
5:4	STEPx_BOCS[1:0]	R/W	00b	Burnout current source enable and magnitude selection Selects the current value of the burnout current source. 00b = off 01b = 2 µA 10b = 8 µA 11b = 85 µA
3:0	STEPx_SYS_MON[3:0]	R/W	0000b	System monitor input selection Selects the system monitor input for the ADC. The AINP[4:0] and AINN[4:0] bits have no effect when one of the system monitors is selected. The analog inputs as well as the TDAC muxed signals are disconnected from the buffer when a system monitor is selected. Internal 2.5V diagnostic reference is used for all settings. 0000b = off (monitors not selected) 0001b = Internal short: Positive and negative inputs shorted to AVSS 0010b = Temperature Sensor 0011b = (AVDD-AVSS)/3 0100b = (CAPA-AVSS)/1 0101b = (IOVDD-DGND)/3 0110b = (CAPD-DGND)/1 0111b = (REFP-REFN)/3 1000b = off (monitors not selected) 1001b = off (monitors not selected) 1010b = off (monitors not selected) 1011b = off (monitors not selected) 1100b = off (monitors not selected) 1101b = off (monitors not selected) 1110b = off (monitors not selected) 1111b = off (monitors not selected)

7.6.2.13 STEPx_TDAC_CFG0 Register (Address = 0Ch) [Reset = 00h]

Return to the Summary Table.

Figure 7-127. STEPx_TDAC_CFG0 Register

7	6	5	4	3	2	1	0
RESERVED			STEPx_TDAC_VAL[4:0]				
R-000b			R/W-00000b				

Table 7-106. STEPx_TDAC_CFG0 Register Field Descriptions

Bit	Field	Type	Reset	Description
7:5	RESERVED	R	000b	Reserved Always reads 000b
4:0	STEPx_TDAC_VAL[4:0]	R/W	00000b	TestDAC output value selection This is a 5-bit DAC with straight binary coding (equidistant test points). Reference value is same as selected on global page but uses the diagnostic/redundant reference.

7.6.2.14 STEPx_TDAC_CFG1 Register (Address = 0Dh) [Reset = 00h]

Return to the Summary Table.

Figure 7-128. STEPx_TDAC_CFG1 Register

7	6	5	4	3	2	1	0
RESERVED			STEPx_TDAC_SEL[4:0]				
R-000b			R/W-00000b				

Table 7-107. STEPx_TDAC_CFG1 Register Field Descriptions

Bit	Field	Type	Reset	Description
7:5	RESERVED	R	000b	Reserved Always reads 000b
4:0	STEPx_TDAC_SEL[4:0]	R/W	00000b	Test DAC multiplexer input selection This register is ignored if system monitors are active. Selects the multiplexer channel for injection of Test DAC output signal. 00000b = open 00001b = TDAC unbuffered to positive input; negative input is connected to AVSS 00010b = TDAC unbuffered to negative input, positive input is connected to AVSS 00011b = AIN0 00100b = AIN1 00101b = AIN2 00110b = AIN3 00111b = AIN4 01000b = AIN5 01001b = AIN6 01010b = AIN7 01011b = AIN8 01100b = AIN9 01101b = AIN10 01110b = AIN11 01111b = AIN12 10000b = AIN13 10001b = AIN14 10010b = AIN15 10011b = REFP/TDAC pin 10100b = open 10101b = open 10110b = open 10111b = open 11000b = open 11001b = open 11010b = open 11011b = open 11100b = open 11101b = open 11110b = open 11111b = open

7.6.2.15 STEPx_SPARE_CFG Register (Address = 0Eh) [Reset = 00h]

Return to the Summary Table.

Figure 7-129. STEPx_SPARE_CFG Register

7	6	5	4	3	2	1	0
STEPx_SPARE7	STEPx_SPARE6	STEPx_SPARE5	STEPx_SPARE4	STEPx_SPARE3	STEPx_SPARE2	STEPx_SPARE1	STEPx_SPARE0
R/W-0b	R/W-0b	R/W-0b	R/W-0b	R/W-0b	R/W-0b	R/W-0b	R/W-0b

Table 7-108. STEPx_SPARE_CFG Register Field Descriptions

Bit	Field	Type	Reset	Description
7	STEPx_SPARE7	R/W	0b	Spare register bit For user functions or CRC checking. 0b = Spare is programmed to 0b 1b = Spare is programmed to 1b
6	STEPx_SPARE6	R/W	0b	Spare register bit For user functions or CRC checking. 0b = Spare is programmed to 0b 1b = Spare is programmed to 1b
5	STEPx_SPARE5	R/W	0b	Spare register bit For user functions or CRC checking. 0b = Spare is programmed to 0b 1b = Spare is programmed to 1b
4	STEPx_SPARE4	R/W	0b	Spare register bit For user functions or CRC checking. 0b = Spare is programmed to 0b 1b = Spare is programmed to 1b
3	STEPx_SPARE3	R/W	0b	Spare register bit For user functions or CRC checking. 0b = Spare is programmed to 0b 1b = Spare is programmed to 1b
2	STEPx_SPARE2	R/W	0b	Spare register bit For user functions or CRC checking. 0b = Spare is programmed to 0b 1b = Spare is programmed to 1b
1	STEPx_SPARE1	R/W	0b	Spare register bit For user functions or CRC checking. 0b = Spare is programmed to 0b 1b = Spare is programmed to 1b
0	STEPx_SPARE0	R/W	0b	Spare register bit For user functions or CRC checking. 0b = Spare is programmed to 0b 1b = Spare is programmed to 1b

7.6.2.16 STEPx_AGPIO_DATA_OUT Register (Address = 0Fh) [Reset = 00h]

Return to the Summary Table.

Figure 7-130. STEPx_AGPIO_DATA_OUT Register

7	6	5	4	3	2	1	0
STEPx_AGPIO7_DAT_OUT	STEPx_AGPIO6_DAT_OUT	STEPx_AGPIO5_DAT_OUT	STEPx_AGPIO4_DAT_OUT	STEPx_AGPIO3_DAT_OUT	STEPx_AGPIO2_DAT_OUT	STEPx_AGPIO1_DAT_OUT	STEPx_AGPIO0_DAT_OUT
R/W-0b	R/W-0b	R/W-0b	R/W-0b	R/W-0b	R/W-0b	R/W-0b	R/W-0b

Table 7-109. STEPx_AGPIO_DATA_OUT Register Field Descriptions

Bit	Field	Type	Reset	Description
7	STEPx_AGPIO7_DAT_OUT	R/W	0b	AGPIO7 data Write value of AGPIO7 when configured as output. Bit setting has no effect when AGPIO7 is configured as input. 0b = Low 1b = High
6	STEPx_AGPIO6_DAT_OUT	R/W	0b	AGPIO6 data Write value of AGPIO6 when configured as output. Bit setting has no effect when AGPIO6 is configured as input. 0b = Low 1b = High
5	STEPx_AGPIO5_DAT_OUT	R/W	0b	AGPIO5 data Write value of AGPIO5 when configured as output. Bit setting has no effect when AGPIO5 is configured as input. 0b = Low 1b = High
4	STEPx_AGPIO4_DAT_OUT	R/W	0b	AGPIO4 data Write value of AGPIO4 when configured as output. Bit setting has no effect when AGPIO4 is configured as input. 0b = Low 1b = High
3	STEPx_AGPIO3_DAT_OUT	R/W	0b	AGPIO3 data Write value of AGPIO3 when configured as output. Bit setting has no effect when AGPIO3 is configured as input. 0b = Low 1b = High
2	STEPx_AGPIO2_DAT_OUT	R/W	0b	AGPIO2 data Write value of AGPIO2 when configured as output. Bit setting has no effect when AGPIO2 is configured as input. 0b = Low 1b = High
1	STEPx_AGPIO1_DAT_OUT	R/W	0b	AGPIO1 data Write value of AGPIO1 when configured as output. Bit setting has no effect when AGPIO1 is configured as input. 0b = Low 1b = High
0	STEPx_AGPIO0_DAT_OUT	R/W	0b	AGPIO0 data Write value of AGPIO0 when configured as output. Bit setting has no effect when AGPIO0 is configured as input. 0b = Low 1b = High

7.6.2.17 STEPx_GPIO_DATA_OUT Register (Address = 10h) [Reset = 00h]

Return to the Summary Table.

Figure 7-131. STEPx_GPIO_DATA_OUT Register

7	6	5	4	3	2	1	0
RESERVED				STEPx_GPIO3_DAT_OUT	STEPx_GPIO2_DAT_OUT	STEPx_GPIO1_DAT_OUT	STEPx_GPIO0_DAT_OUT
R-0000b				R/W-0b	R/W-0b	R/W-0b	R/W-0b

Table 7-110. STEPx_GPIO_DATA_OUT Register Field Descriptions

Bit	Field	Type	Reset	Description
7:4	RESERVED	R	0000b	Reserved Always reads 0000b
3	STEPx_GPIO3_DAT_OUT	R/W	0b	GPIO3 data Write value of GPIO3 when configured as output. Bit setting has no effect when GPIO3 is configured as input. 0b = Low 1b = High
2	STEPx_GPIO2_DAT_OUT	R/W	0b	GPIO2 data Write value of GPIO2 when configured as output. Bit setting has no effect when GPIO2 is configured as input. 0b = Low 1b = High
1	STEPx_GPIO1_DAT_OUT	R/W	0b	GPIO1 data Write value of GPIO1 when configured as output. Bit setting has no effect when GPIO1 is configured as input. 0b = Low 1b = High
0	STEPx_GPIO0_DAT_OUT	R/W	0b	GPIO0 data Write value of GPIO0 when configured as output. Bit setting has no effect when GPIO0 is configured as input. 0b = Low 1b = High

7.6.2.18 STEPx_REG_MAP_CRC Register (Address = 3Dh) [Reset = 00h]

Return to the Summary Table.

Figure 7-132. STEPx_REG_MAP_CRC Register

<table>
<tr><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td colspan="8">STEPx_REG_MAP_CRC_VALUE[7:0]</td></tr>
<tr><td colspan="8">R/W-00000000b</td></tr>
</table>

Table 7-111. STEPx_REG_MAP_CRC Register Field Descriptions

Bit	Field	Type	Reset	Description
7:0	STEPx_REG_MAP_CRC_VALUE[7:0]	R/W	00000000b	Register map CRC for Step Configuration Page Register map CRC value The register map CRC value is the user-computed CRC value of 0x00 to 0x10 registers in the step page. The CRC value written to this register is compared to an internal CRC calculation. If the values do not match, the REG_MAP_CRC_FAULTn bit is set. Enable the register map CRC using the REG_MAP_CRC_EN bit.

7.6.2.19 STEPx_PAGE_INDICATOR Register (Address = 3Eh) [Reset = 00h]

Return to the Summary Table.

Figure 7-133. STEPx_PAGE_INDICATOR Register

<table>
<tr><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td colspan="8">STEPx_PAGE_INDICATOR[7:0]</td></tr>
<tr><td colspan="8">R-00000000b</td></tr>
</table>

Table 7-112. STEPx_PAGE_INDICATOR Register Field Descriptions

Bit	Field	Type	Reset	Description
7:0	STEPx_PAGE_INDICATOR[7:0]	R	00000000b	Register page Indicator Indicates the active register page.

7.6.2.20 STEPx_PAGE_POINTER Register (Address = 3Fh) [Reset = 00h]

Return to the Summary Table.

Figure 7-134. STEPx_PAGE_POINTER Register

7	6	5	4	3	2	1	0
STEPx_PAGE_POINTER[7:0]							
R/W-00000000b							

Table 7-113. STEPx_PAGE_POINTER Register Field Descriptions

Bit	Field	Type	Reset	Description
7:0	STEPx_PAGE_POINTER[7:0]	R/W	00000000b	Register page pointer Selects the active register page.

8 Application and Implementation

Note

Information in the following applications sections is not part of the TI component specification, and TI does not warrant its accuracy or completeness. TI's customers are responsible for determining suitability of components for their purposes, as well as validating and testing their design implementation to confirm system functionality.

8.1 Application Information

8.1.1 Serial Interface Connections

Figure 8-1 shows the basic interface connections for the ADS125P08.

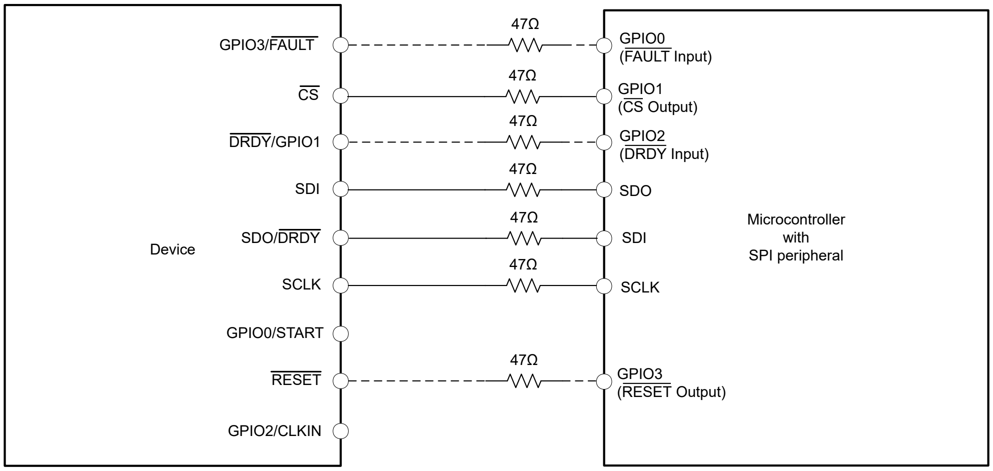


Figure 8-1. Serial Interface Connections

Most microcontroller SPI peripherals can interface with the device. The interface operates in SPI mode 1 where CPOL = 0 and CPHA = 1. In SPI mode 1, SCLK idles low and data are launched or changed only on SCLK rising edges; data are latched or read on SCLK falling edges.

Optionally, route the dedicated DRDY pin to a falling edge triggered interrupt-capable GPIO of the host controller in case new data ready indication through an interrupt is desired. By default, the DRDY/GPIO1 pin is already configured as a DRDY output (GPIO1_CFG[1:0] = 11b). Alternatively, disable the DRDY/GPIO1 pin by setting GPIO1_CFG[1:0] bits to 00b to reduce the number of connections requiring isolation.

The FAULT pin can be interfaced to the host controller as well in case fault indication through a pin is desired besides the fault indication through the fault flags. For that purpose, configure the GPIO3/FAULT pin as a FAULT output (GPIO3_CFG[1:0] = 11b).

Optionally, place 47Ω resistors in series with all digital input and output pins. This resistance smooths sharp signal transitions, suppresses overshoot, and offers some overvoltage protection. Care must be taken to meet all SPI timing requirements because the additional resistors interact with the bus capacitances present on the digital signal lines.

Pullup or pulldown resistors can be placed on the digital input and output signal lines in case a certain signal level needs to be driven during power up of the device or the microcontroller.

8.1.2 Interfacing with Multiple Devices

The ADS125P08 offers three methods to operate multiple devices on a single SPI bus:

- Daisy-chaining using a single CS signal for all devices as explained in the *Daisy-Chain Operation* section. The host connects to SDI of the first device in the chain to transmit data. The SDO signal of the first device in the chain connects to the SDI signal of the next device, and so on. The host controller receives data from the SDO signal of the last device in the chain. All devices share the same SCLK signal. This method allows the host to talk to all devices in the chain at the same time. However, depending on the number of devices connected in the chain, the SPI frame can get very long.

- Using a dedicated $\overline{\text{CS}}$ signal for each device as shown in Figure 8-2. All devices share the SCLK, SDI and SDO/$\overline{\text{DRDY}}$ signals in this case. Only the device with $\overline{\text{CS}}$ low drives the SDO/$\overline{\text{DRDY}}$ pin. The SDO/$\overline{\text{DRDY}}$ outputs of all other devices, which have $\overline{\text{CS}}$ high, are in a high-Z state to avoid contention on the SDO line. The host controller interfaces with each device one at a time.
- Using the chip select forward mode (CS-FWD mode) as explained in the *Chip Select Forwarding* section.

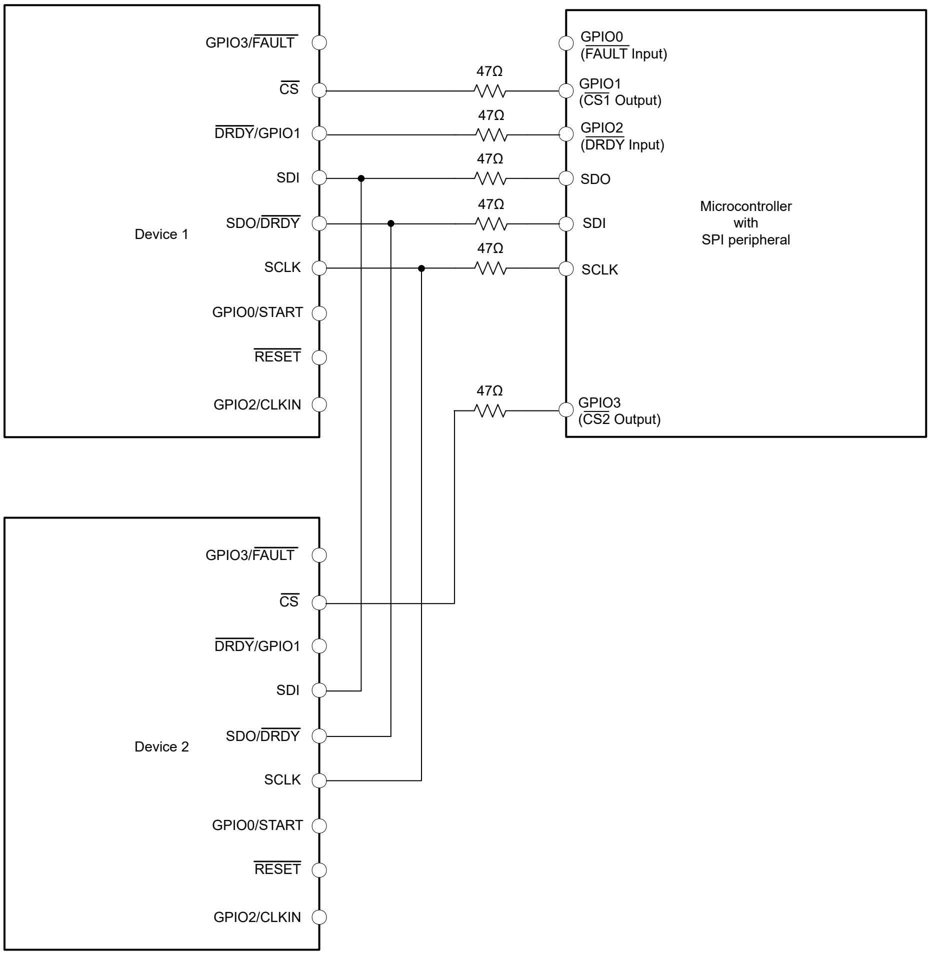

Device 1
GPIO3/$\overline{\text{FAULT}}$
$\overline{\text{CS}}$
$\overline{\text{DRDY}}$/GPIO1
SDI
SDO/$\overline{\text{DRDY}}$
SCLK
GPIO0/START
$\overline{\text{RESET}}$
GPIO2/CLKIN
47Ω
47Ω
47Ω
47Ω
47Ω
47Ω
Microcontroller
with
SPI peripheral
GPIO0
($\overline{\text{FAULT}}$ Input)
GPIO1
($\overline{\text{CS1}}$ Output)
GPIO2
($\overline{\text{DRDY}}$ Input)
SDO
SDI
SCLK
GPIO3
($\overline{\text{CS2}}$ Output)
Device 2
GPIO3/$\overline{\text{FAULT}}$
$\overline{\text{CS}}$
$\overline{\text{DRDY}}$/GPIO1
SDI
SDO/$\overline{\text{DRDY}}$
SCLK
GPIO0/START
$\overline{\text{RESET}}$
GPIO2/CLKIN

Figure 8-2. Multiple Device Serial Interface Connections using individual $\overline{\text{CS}}$ Signals

8.1.3 Unused Inputs and Outputs

Follow these guidelines for unused device pin connections:

- Leave any unused analog inputs floating or connect the unused analog inputs to GND.
- When not using the REFP, REFN, GPIO0, GPIO1, GPIO2, GPIO3, $\overline{\text{FAULT}}$, $\overline{\text{DRDY}}$, or CLK functions, configure the respective pins as disabled/high impedance pins (GPIOx_CFG[1:0] = 00b) and follow the guidelines for unused analog inputs above.
- When not using the $\overline{\text{RESET}}$ pin, leave the pin floating (due to the internal pullup resistor) or optionally connect to an external pullup resistor.

8.1.4 Device Initialization

Figure 8-3 illustrates the sequence steps required to initialize the ADS125P08, and to start conversions in continuous sequence mode. In this example, the device uses the dedicated $\overline{\text{DRDY}}$ pin to indicate availability of new conversion data to the host controller.

Configure the SPI of the host controller to the SPI mode defined by CPOL = 0, CPHA = 1 and configure the host controller GPIO connected to the device $\overline{\text{DRDY}}$ pin as a falling edge triggered interrupt input.

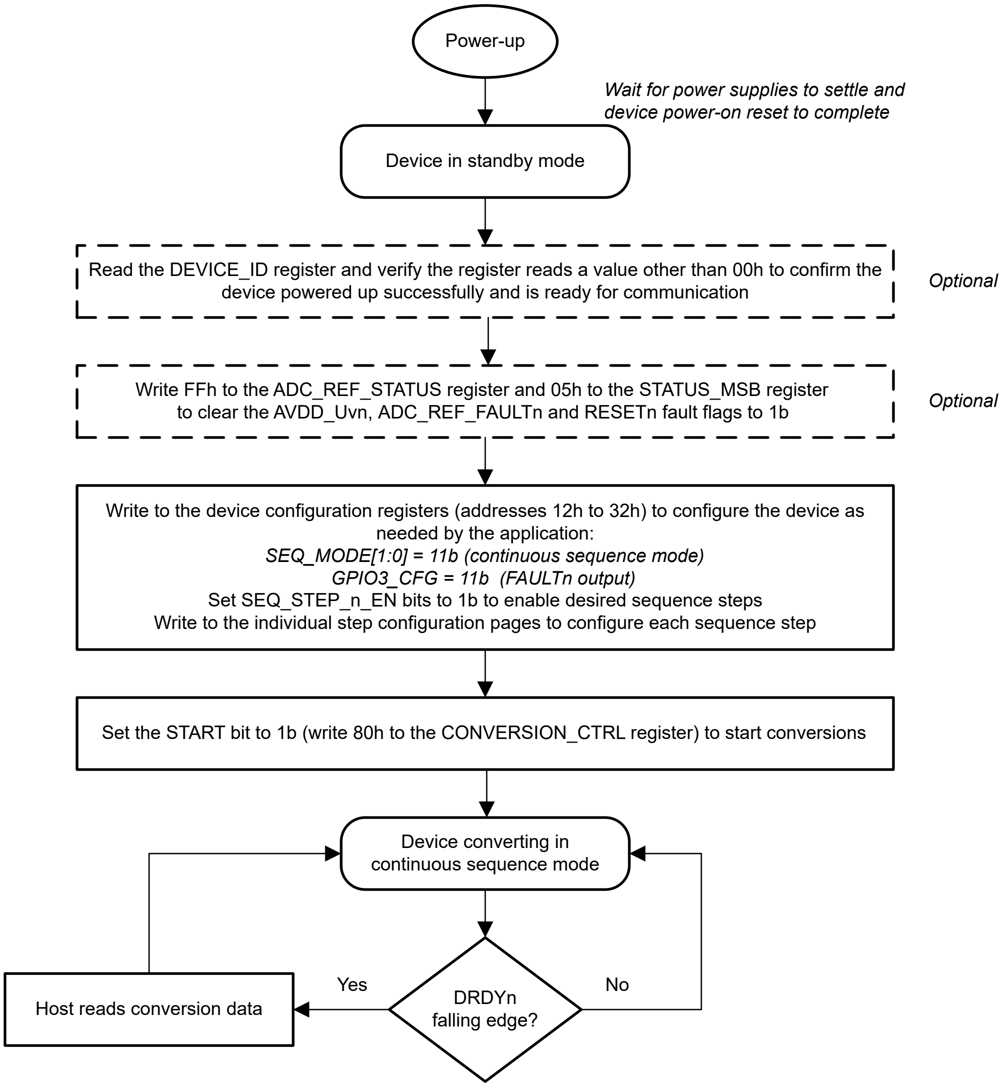

Power-up
Wait for power supplies to settle and device power-on reset to complete
Device in standby mode
Read the DEVICE_ID register and verify the register reads a value other than 00h to confirm the device powered up successfully and is ready for communication
Optional
Write FFh to the ADC_REF_STATUS register and 05h to the STATUS_MSB register to clear the AVDD_Uvn, ADC_REF_FAULTn and RESETn fault flags to 1b
Optional
Write to the device configuration registers (addresses 12h to 32h) to configure the device as needed by the application:
SEQ_MODE[1:0] = 11b (continuous sequence mode)
GPIO3_CFG = 11b (FAULTn output)
Set SEQ_STEP_n_EN bits to 1b to enable desired sequence steps
Write to the individual step configuration pages to configure each sequence step
Set the START bit to 1b (write 80h to the CONVERSION_CTRL register) to start conversions
Device converting in continuous sequence mode
DRDYn falling edge?
Yes
No
Host reads conversion data

Figure 8-3. Device Initialization Flow Chart

8.2 Typical Application

The ADS125P08 integrates all necessary features (such as input multiplexer, buffered external reference input and high impedance buffer) to implement a multitude of sensor measurements in parallel, such as pressure, temperature, voltage and current for various applications.

Figure 8-4 shows an example to implement a resistive bridge sensor measurement with temperature compensation using the ADS125P08. The bridge temperature is typically used in the host controller to compensate for the bridge temperature drift.

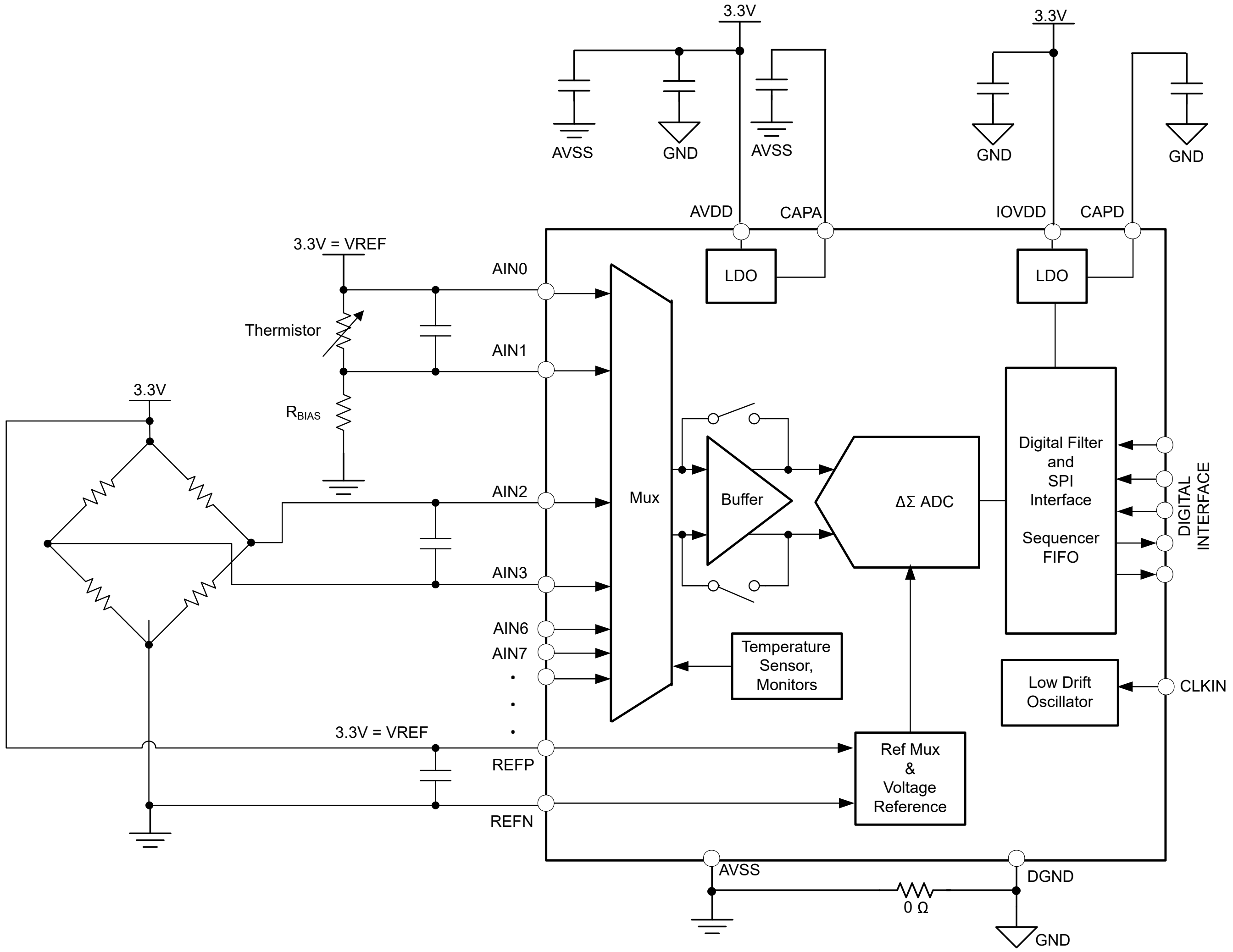


Figure 8-4. Resistive Bridge Sensor Measurement with the ADS125P08, Using the Analog Supply as Bridge Excitation and a Thermistor for Bridge Temperature Measurement

8.2.1 Design Requirements

Table 8-1. Resistive Bridge Sensor Measurement With Temperature Compensation

DESIGN PARAMETER	VALUE
Supply voltage	3.3V (analog and digital)
$V_{Excitation}$ (Bridge excitation voltage)	3.3V
Bridge sensitivity	2.0 mV/V (±15%)

Table 8-1. Resistive Bridge Sensor Measurement With Temperature Compensation (continued)

DESIGN PARAMETER	VALUE
Non-linearity	±0.05 %FS
Non-repeatability	
Hysteresis	
Bridge Resistance	1000 Ω (±1%)
Operating Temperature Range	–10°C to +40°C

8.2.2 Detailed Design Procedure

The circuit implementation in Figure 8-4 uses the analog supply to excite the bridge sensor. Use the bridge excitation voltage as the external reference voltage for the ADC to implement a *ratiometric* bridge measurement. Equation 32 through Equation 34 show how to derive the relationship between ADC output codes and the applied bridge signal using a pressure sensor as an example. Equation 34 shows that the output codes are independent of the excitation voltage in this ratiometric circuit implementation.

$$V_{Bridge} = V_{AIN2} – V_{AIN3} = (Pressure_{APL} / Pressure_{MAX}) \times Sensitivity \times V_{Excitation} \qquad (32)$$

$$Code / 2^n = V_{Bridge} / V_{REF} \qquad (33)$$

$$Code / 2^n = (Pressure_{APL} / Pressure_{MAX}) \times Sensitivity \qquad (34)$$

Where:

- $V_{Excitation} = V_{REF} = AVDD$
- $Pressure_{APL}$ = the applied pressure
- $Pressure_{MAX}$ = the maximum capacity of the pressure sensor. Means the pressure where the bridge sensor outputs the full-scale output signal
- Sensitivity = the sensitivity of the bridge sensor typically given in mV/V of bridge excitation
- n depends on the selected coding scheme and the ADC resolution, see below:

The guideline for selection of n is:

- n = 15 (16bit ADC, binary two's complement format)
- n = 16 (16bit ADC, unipolar straight binary format)
- n = 23 (24bit ADC, binary two's complement format)
- n = 24 (24bit ADC, unipolar straight binary format)

The example in Figure 8-4 shows a thermistor to measure the bridge temperature. The reference voltage output is used in this case to implement a *ratiometric* thermistor measurement. The conversion result according to Equation 36 is only dependent on the bias resistor (R_{BIAS}) and the thermistor resistance (which strongly depends on the temperature).

$$V_{Thermistor} = V_{AIN0} – V_{AIN1} = V_{REF} \times R_{Thermistor} / (R_{Thermistor} + R_{BIAS}) \qquad (35)$$

$$Code / 2^n = (V_{Thermistor}) / V_{REF} = (R_{Thermistor}) / (R_{Thermistor} + R_{BIAS}) \qquad (36)$$

Where n follows the guidelines of Equation 34.

Use one of the GPIO outputs to control a switch placed between the bridge sensor and GND for applications where the bridge sensor needs to be powered down periodically to save power.

For more information about resistive bridge sensor measurement circuits and the implementation using TI ADCs see the *A Basic Guide to Bridge Measurements* application note.

8.2.3 Application Performance Plots - Crosstalk

Figure 8-5 illustrates a setup for a common channel-to-channel crosstalk test procedure. Each even numbered input channel (examples: AIN0, AIN2, AIN4, and more) is driven by an individual voltage source. All odd numbered channels (examples: AIN1, AIN3, AIN5, and more) are shorted together to ground.

The test procedure is defined as follows: Apply 0.3V constant signal to the input channel under test (example: AIN0), and apply the "interference" pattern shown in Figure 8-6 to all remaining even numbered channels (examples: AIN2, AIN4, … AIN14). Measure the ADC output of the channel under test (example: AIN0) to determine transient deviations from a constant output code. Repeat this procedure for every even-numbered channel. Commonly, the variation in output code is desired to be ±1 LSB on the 16 bit level, or smaller.

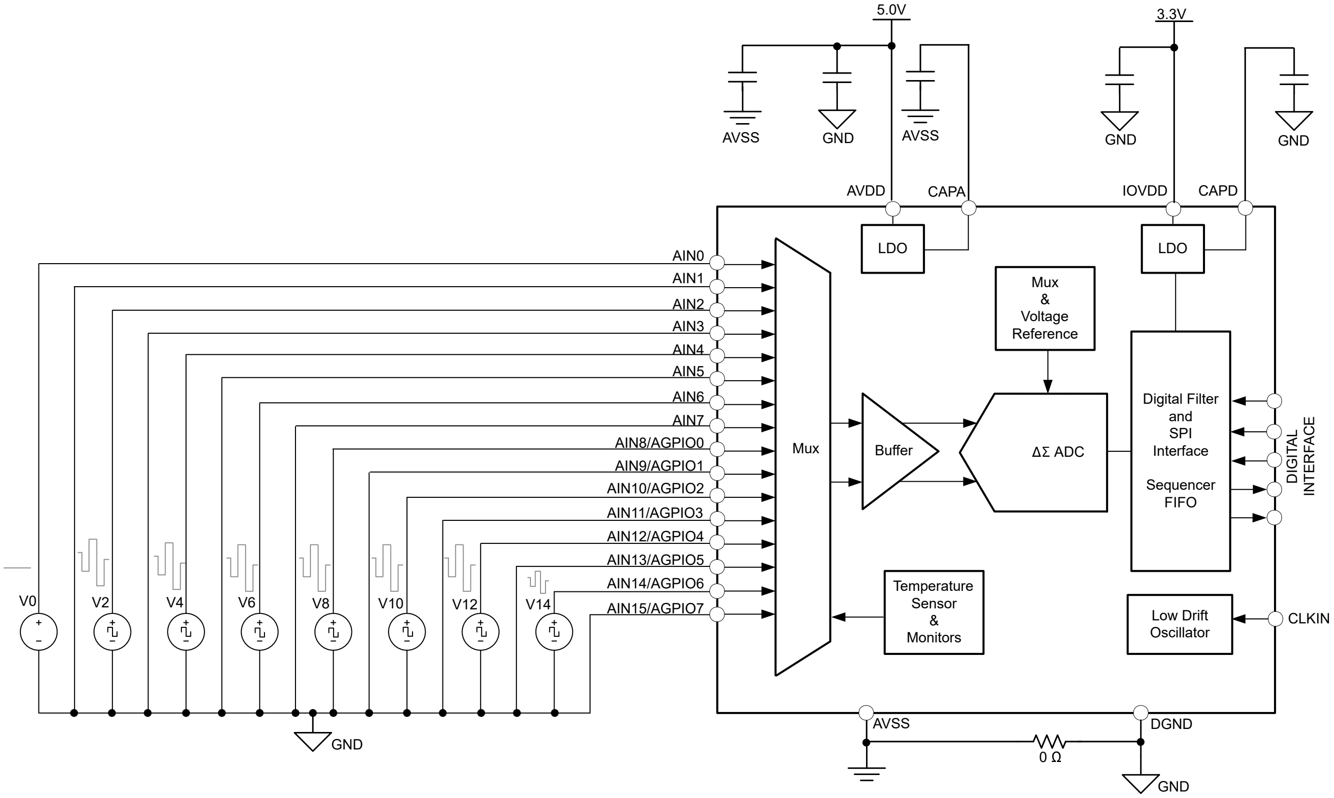


Figure 8-5. Crosstalk Test Setup

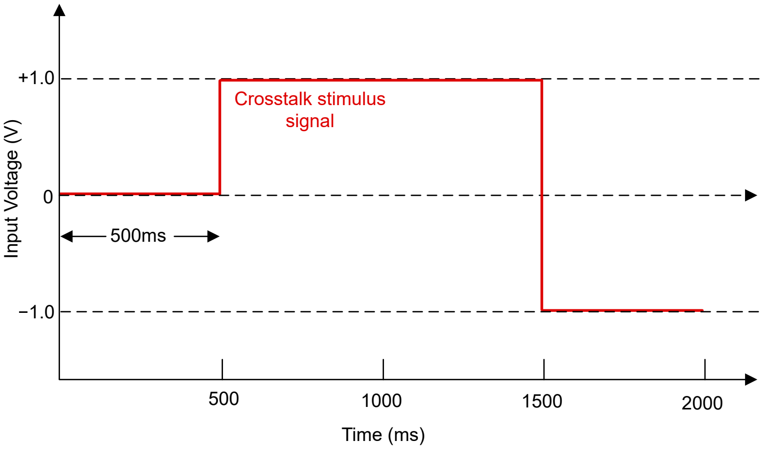


Figure 8-6. Crosstalk Stimulus Input Signal

Figure 8-7 and Figure 8-8 depict the measured results for the crosstalk test using the setup from Figure 8-5.

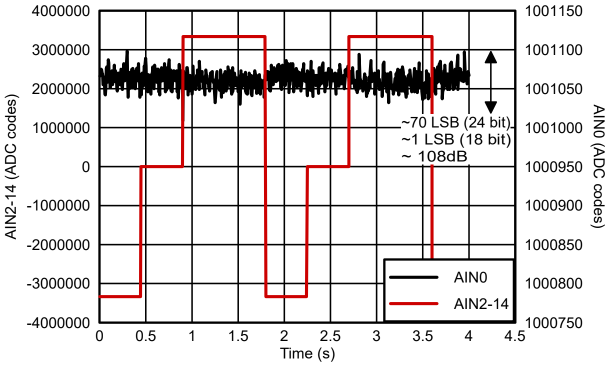


Figure 8-7. Crosstalk Measurement for ADC Input AIN0

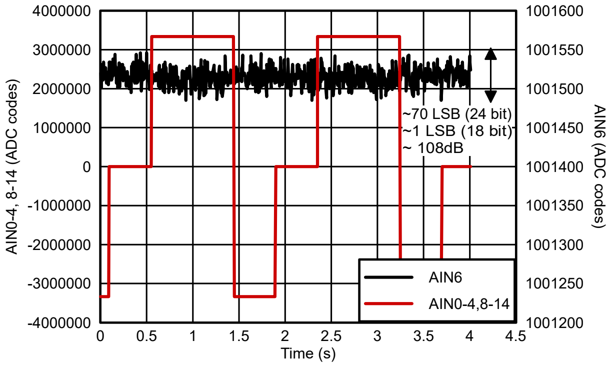


Figure 8-8. Crosstalk Measurement for ADC Input AIN6

All inputs except the input channel "under test" undergo large transient voltage steps as described by the test procedure above. The impact on the channel under test, for example AIN0 in Figure 8-7, is visible in the zoomed plot for AIN0 output codes (see secondary y-axis). In all cases, the deviation from the ideal output is approximately 70 codes (LSBs) on the 24 bit level or less, which corresponds to 1LSB on the 18 bit level.

All other channels perform similar to AIN0 when undergoing the same test. As another example and to demostrate the similarity between channels, the impact on channel AIN6 from crosstalk is also shown as seen in Figure 8-8. All other channels (AIN2, AIN4, AIN8, AIN10, AIN12, AIN14) are tested using the same procedure, and crosstalk in the order of 18 bit level or less is confirmed.

8.3 Power Supply Recommendations

8.3.1 Power Supplies

The device requires two power supplies: analog (AVDD) and digital (IOVDD). The analog power supply can be independently chosen from the digital power supply. The IOVDD supply sets the logic levels for the serial interface pins ($\overline{\text{CS}}$, SCLK, SDI, SDO/$\overline{\text{DRDY}}$) and other digital I/O pins. The AVDD supply sets the logic levels for the GPIOs (GPIO0 to GPIO3).

8.3.2 Power-Supply Sequencing

The power supplies can be sequenced in any order, but in no case must any analog or digital inputs exceed the respective analog or digital power-supply voltage and current limits.

8.3.3 Power-Supply Decoupling

Good power-supply decoupling is important to achieve optimum device performance. As shown in Figure 8-9, AVDD, IOVDD and CAPD must each be decoupled with at least a 1µF capacitor to GND. Also, AVDD, CAPA, REFOUT and REFP must be bypassed with a 1µF capacitor to AVSS. Place the supply bypass capacitors as close to the device power-supply pins as possible using low-impedance connections. Use multi-layer ceramic chip capacitors (MLCCs) that offer low equivalent series resistance (ESR) and inductance (ESL) characteristics for power-supply decoupling purposes. For very sensitive systems, or for systems in harsh noise environments, avoiding the use of vias for connecting the capacitors to the device pins can offer superior noise immunity. The use of multiple vias in parallel lowers the overall inductance and is beneficial for connections to ground planes.

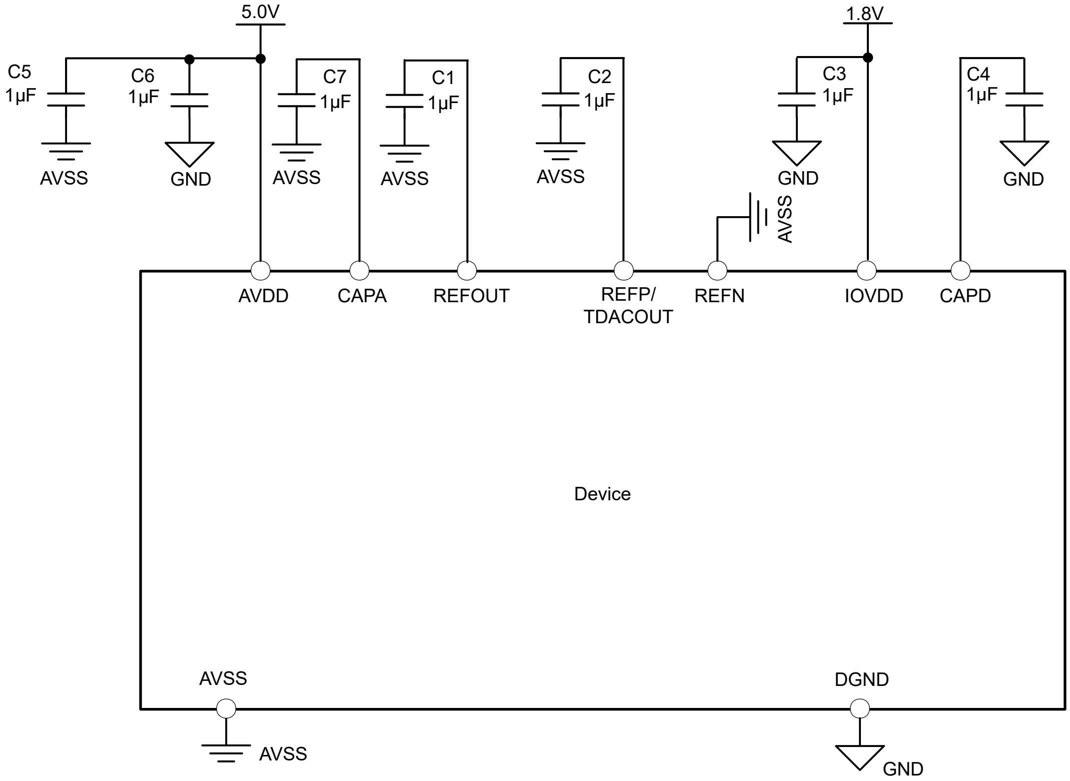


Figure 8-9. Power Supply Decoupling

8.4 Layout

8.4.1 Layout Guidelines

The following basic recommendations for the ADS125P08 layout help achieve the best possible performance of the ADC.

- For best performance, dedicate an entire PCB layer to a ground plane and do not route any other signal traces on this layer. However, depending on restrictions imposed by specific end equipment, a dedicated ground plane is not always practical. If ground plane separation is necessary, make a direct connection of the planes at the device. Do not connect individual ground planes at multiple locations to avoid the creation of unintentional ground loops.
- Use ceramic capacitors (for example, X7R grade) for the power-supply decoupling capacitors. High-K capacitors (Y5V) are not recommended. Place the required capacitors as close as possible to the device pins using short, direct traces. Placing the bypass capacitors on the same layer as close to the device yields the best results.
- Route digital traces away from all analog inputs and associated components to minimize interference.
- Provide good ground return paths. Signal return currents flow on the path of least impedance. If the ground plane is cut or has other traces that block the current from flowing right next to the signal trace, another path must be found to return to the source and complete the circuit. If forced into a larger path, the chance that the signal radiates increases. Sensitive signals are more susceptible to EMI interference.
- Consider the resistance and inductance of the routing. Often, traces for the inputs have resistances that react with the input bias current and cause an added error voltage. Reducing the loop area enclosed by the source signal and the return current reduces the inductance in the path. Reducing the inductance reduces the EMI pickup and reduces the high-frequency impedance at the input of the device.
- Watch for parasitic thermocouples in the layout. Dissimilar metals going from each analog input to the sensor can create a parasitic thermocouple that can add an offset to the measurement. Differential inputs must be matched for both the inputs going to the measurement source.
- Use C0G capacitors for the RC filters on the analog inputs.
- Fill void areas on signal layers with ground fill.

- When applying an external clock, be sure the clock is free of overshoot and glitches. A source-termination resistor placed at the clock buffer often helps reduce overshoot. Glitches present on the clock input can lead to noise within the conversion data.

8.4.2 Layout Example

Figure 8-10 shows a basic layout example for the ADS125P08:

- C1 is the required capacitor at the REFOUT pin to AVSS.
- C2 is the required capacitor at the REFP pin to AVSS.
- C3, C4, C5, C6 and C7 are the power supply decoupling capacitors.
- Differential antialiasing filters are shown for the differential analog input pairs AIN0-AIN1, AIN2-AIN3, AIN4-AIN5, AIN6-AIN7, AIN8-AIN9, AIN10-AIN11, AIN12-AIN13, and AIN14-AIN15, respectively.
- Optional series resistors (R1 to R8) are shown for the SPI and digital lines. The series resistors help to reduce overshoot and ringing on the digital lines by smoothing the signal edges.

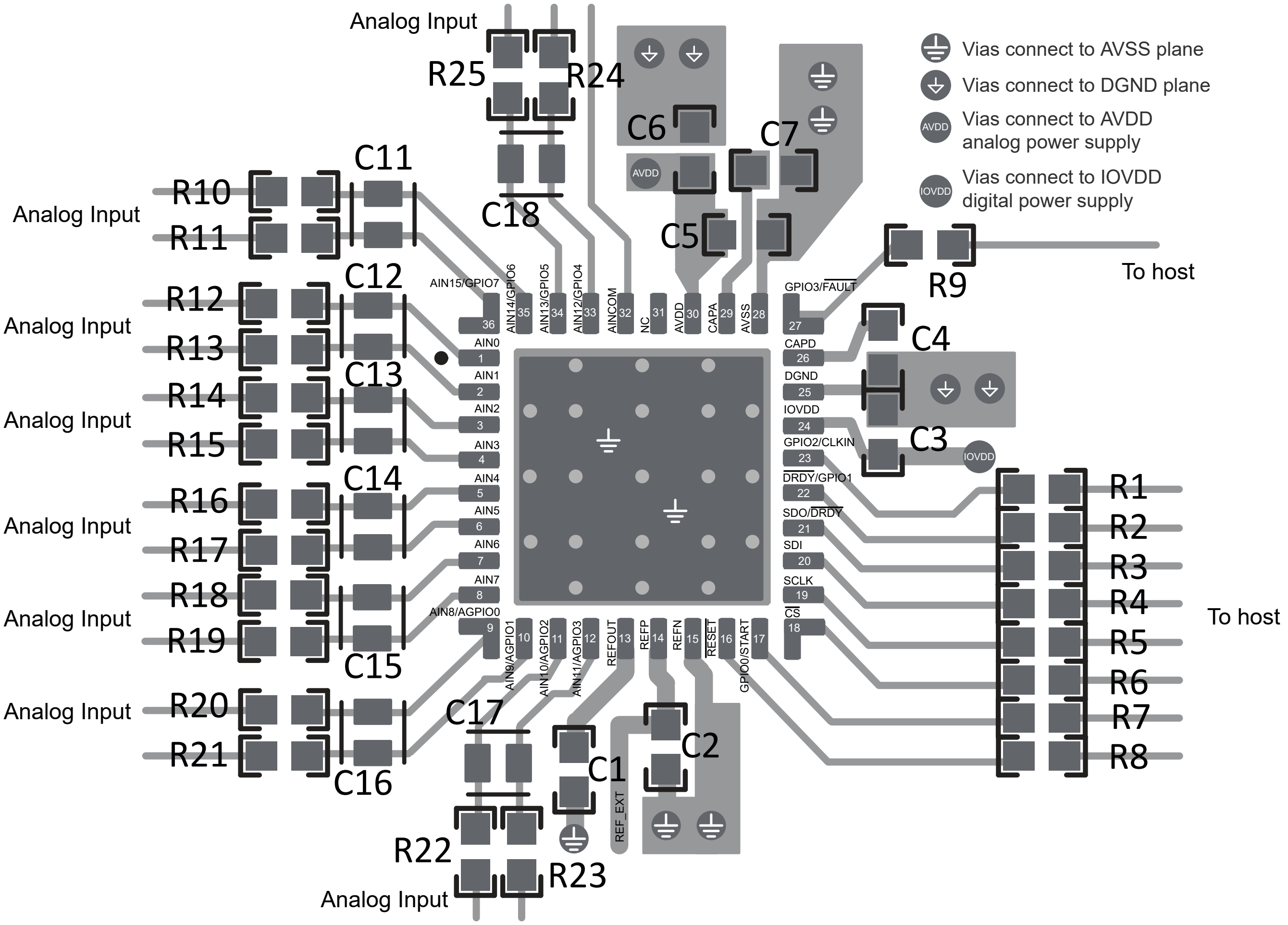


Figure 8-10. Layout Example

9 Device and Documentation Support

9.1 Documentation Support

9.1.1 Related Documentation

For related documentation see the following:

- Texas Instruments, *REF60xx High-Precision Voltage Reference with Integrated ADC Drive Buffer* data sheet
- Texas Instruments, *Design Methodology for MFB Filters in ADC Interface Applications* application note
- Texas Instruments, *QFN and SON PCB Attachment* application note

9.2 Receiving Notification of Documentation Updates

To receive notification of documentation updates, navigate to the device product folder on ti.com. Click on *Notifications* to register and receive a weekly digest of any product information that has changed. For change details, review the revision history included in any revised document.

9.3 Support Resources

TI E2E™ support forums are an engineer's go-to source for fast, verified answers and design help — straight from the experts. Search existing answers or ask your own question to get the quick design help you need.

Linked content is provided "AS IS" by the respective contributors. They do not constitute TI specifications and do not necessarily reflect TI's views; see TI's Terms of Use.

9.4 Trademarks

TI E2E™ is a trademark of Texas Instruments.
All trademarks are the property of their respective owners.

9.5 Electrostatic Discharge Caution

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.

9.6 Glossary

TI Glossary	This glossary lists and explains terms, acronyms, and definitions.

10 Revision History

NOTE: Page numbers for previous revisions may differ from page numbers in the current version.

DATE	REVISION	NOTES
December 2025	*	Initial Release

11 Mechanical, Packaging, and Orderable Information

The following pages include mechanical, packaging, and orderable information. This information is the most current data available for the designated devices. This data is subject to change without notice and revision of this document. For browser-based versions of this data sheet, refer to the left-hand navigation.

PACKAGING INFORMATION

Orderable part number	Status (1)	Material type (2)	Package | Pins	Package qty | Carrier	RoHS (3)	Lead finish/ Ball material (4)	MSL rating/ Peak reflow (5)	Op temp (°C)	Part marking (6)
ADS125P08IRHBR	Active	Production	VQFN (RHB) | 36	3000 | LARGE T&R	-	NIPDAU	Level-2-260C-1 YEAR	-40 to 125	ADS125 P08

[1] **Status:** For more details on status, see our product life cycle.

[2] **Material type:** When designated, preproduction parts are prototypes/experimental devices, and are not yet approved or released for full production. Testing and final process, including without limitation quality assurance, reliability performance testing, and/or process qualification, may not yet be complete, and this item is subject to further changes or possible discontinuation. If available for ordering, purchases will be subject to an additional waiver at checkout, and are intended for early internal evaluation purposes only. These items are sold without warranties of any kind.

[3] **RoHS values:** Yes, No, RoHS Exempt. See the TI RoHS Statement for additional information and value definition.

[4] **Lead finish/Ball material:** Parts may have multiple material finish options. Finish options are separated by a vertical ruled line. Lead finish/Ball material values may wrap to two lines if the finish value exceeds the maximum column width.

[5] **MSL rating/Peak reflow:** The moisture sensitivity level ratings and peak solder (reflow) temperatures. In the event that a part has multiple moisture sensitivity ratings, only the lowest level per JEDEC standards is shown. Refer to the shipping label for the actual reflow temperature that will be used to mount the part to the printed circuit board.

[6] **Part marking:** There may be an additional marking, which relates to the logo, the lot trace code information, or the environmental category of the part.

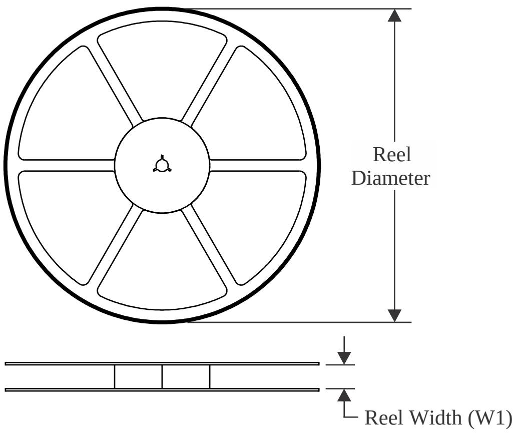
Multiple part markings will be inside parentheses. Only one part marking contained in parentheses and separated by a "~" will appear on a part. If a line is indented then it is a continuation of the previous line and the two combined represent the entire part marking for that device.

Important Information and Disclaimer: The information provided on this page represents TI's knowledge and belief as of the date that it is provided. TI bases its knowledge and belief on information provided by third parties, and makes no representation or warranty as to the accuracy of such information. Efforts are underway to better integrate information from third parties. TI has taken and continues to take reasonable steps to provide representative and accurate information but may not have conducted destructive testing or chemical analysis on incoming materials and chemicals. TI and TI suppliers consider certain information to be proprietary, and thus CAS numbers and other limited information may not be available for release.

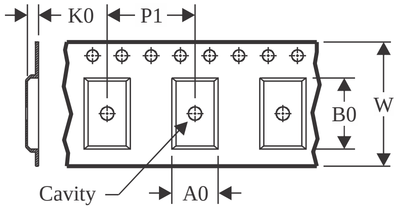
In no event shall TI's liability arising out of such information exceed the total purchase price of the TI part(s) at issue in this document sold by TI to Customer on an annual basis.

TAPE AND REEL INFORMATION

REEL DIMENSIONS

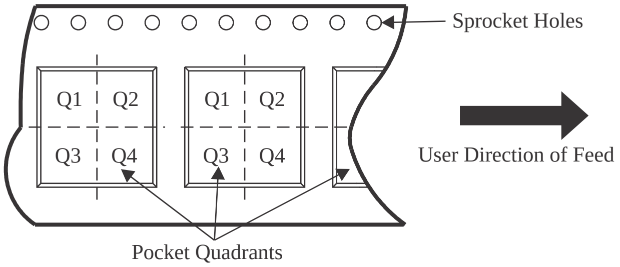


TAPE DIMENSIONS



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

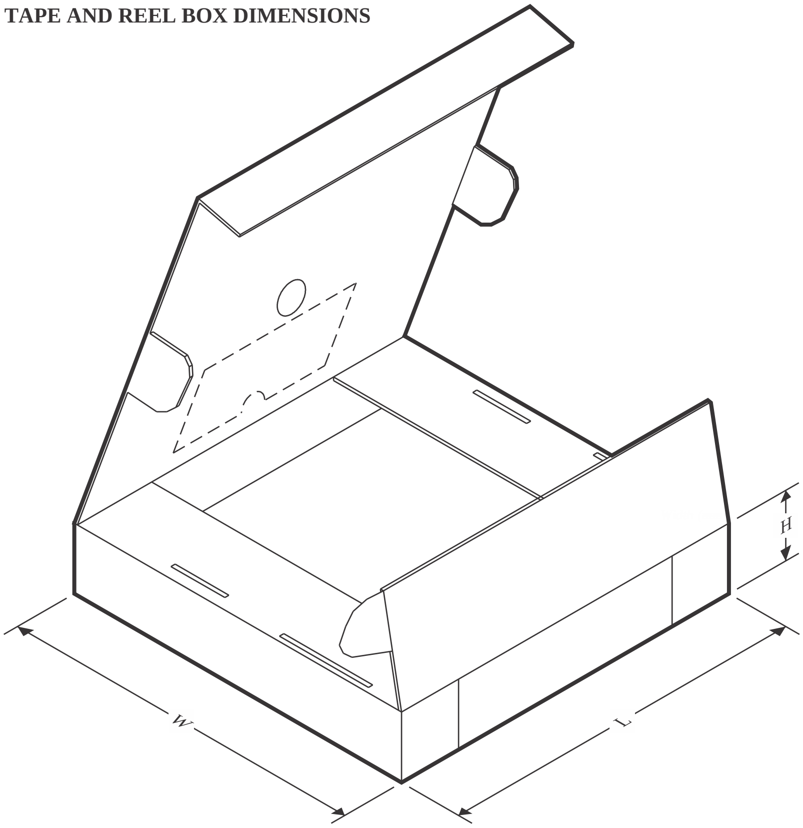
QUADRANT ASSIGNMENTS FOR PIN 1 ORIENTATION IN TAPE



*All dimensions are nominal

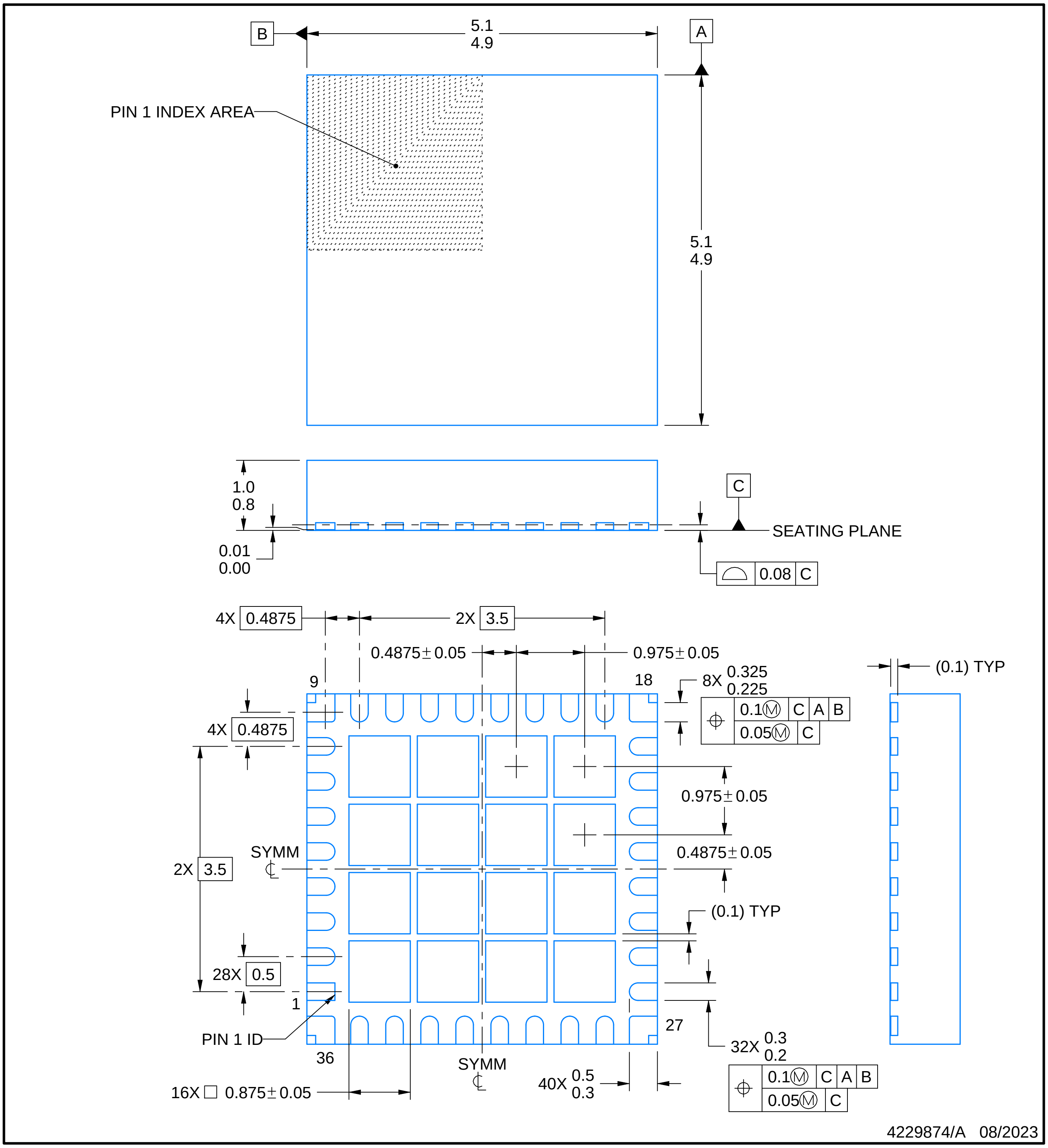
Device	Package Type	Package Drawing	Pins	SPQ	Reel Diameter (mm)	Reel Width W1 (mm)	A0 (mm)	B0 (mm)	K0 (mm)	P1 (mm)	W (mm)	Pin1 Quadrant
ADS125P08IRHBR	VQFN	RHB	36	3000	330.0	12.4	5.3	5.3	1.1	8.0	12.0	Q2

TAPE AND REEL BOX DIMENSIONS



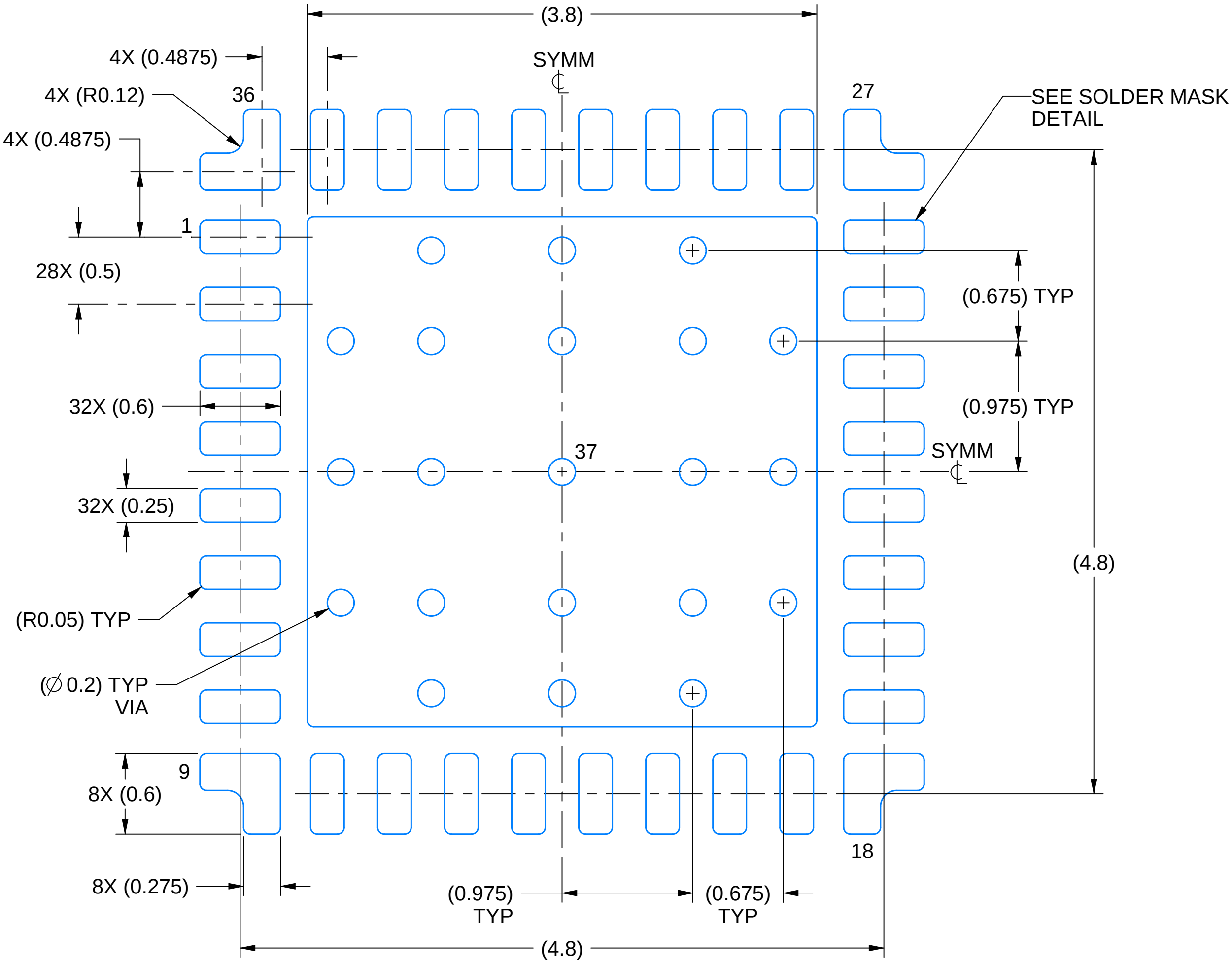
*All dimensions are nominal

Device	Package Type	Package Drawing	Pins	SPQ	Length (mm)	Width (mm)	Height (mm)
ADS125P08IRHBR	VQFN	RHB	36	3000	346.0	346.0	33.0

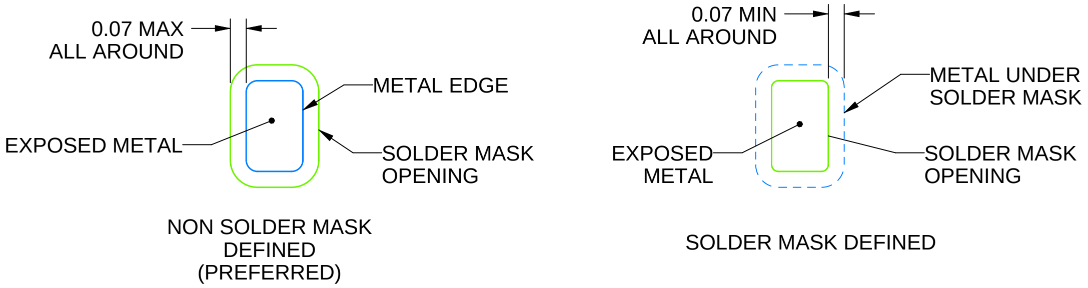


NOTES:

1. All linear dimensions are in millimeters. Any dimensions in parenthesis are for reference only. Dimensioning and tolerancing per ASME Y14.5M.
2. This drawing is subject to change without notice.
3. The package thermal pad must be soldered to the printed circuit board for thermal and mechanical performance.



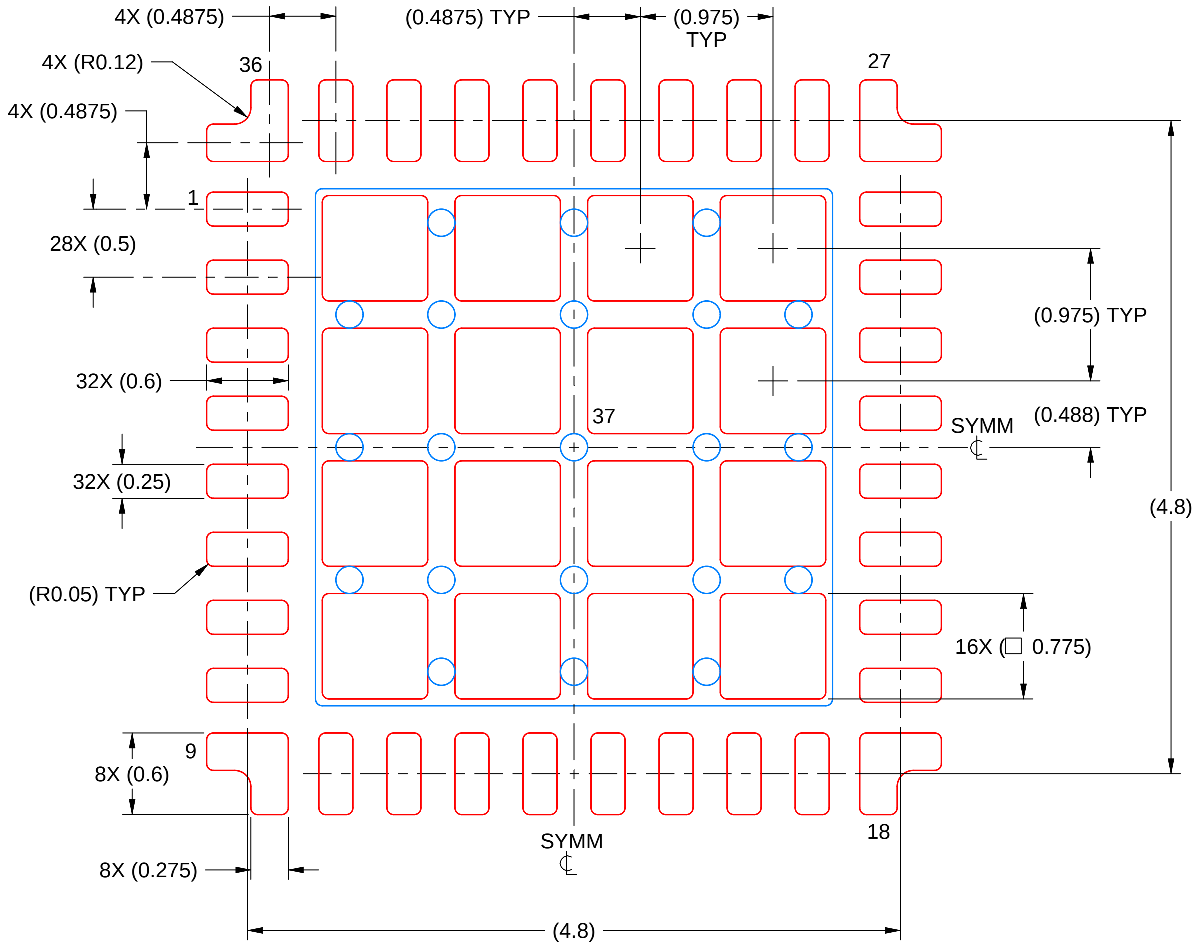
LAND PATTERN EXAMPLE
EXPOSED METAL SHOWN
SCALE: 18X



SOLDER MASK DETAILS

NOTES: (continued)

4. This package is designed to be soldered to a thermal pad on the board. For more information, see Texas Instruments literature number SLUA271 (www.ti.com/lit/slua271).
5. Vias are optional depending on application, refer to device data sheet. If any vias are implemented, refer to their locations shown on this view. It is recommended that vias under paste be filled, plugged or tented.



SOLDER PASTE EXAMPLE
BASED ON 0.125 MM THICK STENCIL
SCALE: 18X

EXPOSED PAD 33
67% PRINTED SOLDER COVERAGE BY AREA UNDER PACKAGE

4229874/A 08/2023

NOTES: (continued)

6. Laser cutting apertures with trapezoidal walls and rounded corners may offer better paste release. IPC-7525 may have alternate design recommendations.

IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATASHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES “AS IS” AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS AND IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, regulatory or other requirements.

These resources are subject to change without notice. TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource. Other reproduction and display of these resources is prohibited. No license is granted to any other TI intellectual property right or to any third party intellectual property right. TI disclaims responsibility for, and you fully indemnify TI and its representatives against any claims, damages, costs, losses, and liabilities arising out of your use of these resources.

TI’s products are provided subject to TI’s Terms of Sale, TI’s General Quality Guidelines, or other applicable terms available either on ti.com or provided in conjunction with such TI products. TI’s provision of these resources does not expand or otherwise alter TI’s applicable warranties or warranty disclaimers for TI products. Unless TI explicitly designates a product as custom or customer-specified, TI products are standard, catalog, general purpose devices.

TI objects to and rejects any additional or different terms you may propose.

Last updated 10/2025