Application Note

Design Considerations for Multiple Wide Bandwidth Delta-Sigma ADCs in Simultaneous-Sampling Systems

Josh Brown, Keith Nicholas

ABSTRACT

Multiple ADC devices operating in simultaneous-sampling mode are often used in data acquisition systems. Examples include sound and vibration sensors, electrical grid monitoring, and medical equipment, including ECG and in-vitro diagnostics. Texas Instruments' family of wide bandwidth delta-sigma ADCs are designed for these applications because of the small package size, daisy-chain connection options, and ability to synchronize. This application note discusses many of the design requirements for these multichannel systems.

Table of Contents

Trademarks

All trademarks are the property of their respective owners.

1 Introduction

This application note reviews clocking, synchronization, daisy-chain considerations, and board layout recommendations in multidevice simultaneous-sampling systems using the ADCs in Table 1-1. These ADCs are referred to as ADS1x7Lxx family throughout the application note. Any differences between these devices are noted but otherwise the information is generally applicable to all devices.

Table 1-1. Wide Bandwidth Delta-Sigma ADCs

Part Number	Resolution	Channel Count	Max Data Rate	Programmable Filter Coefficients	Data Interface Ports
ADS117L11	16	1	1,067kSPS	No	SPI: Configuration and Output Data
ADS127L11	24	1	1,067kSPS	No	SPI: Configuration and Output Data
ADS127L21	24	1	1,365kSPS	Yes	SPI: Configuration and Output Data
ADS127L21B	24	1	1,365kSPS	Yes	SPI: Configuration and Output Data
ADS117L14	16	4	1,365kSPS	No	SPI: Configuration Frame-sync: Output Data
ADS117L18	16	8	1,365kSPS	No	SPI: Configuration Frame-sync: Output Data
ADS127L14	24	4	1,365kSPS	No	SPI: Configuration Frame-sync: Output Data
ADS127L18	24	8	1,365kSPS	No	SPI: Configuration Frame-sync: Output Data

2 Clock Signal

A simultaneous-sampling system requires all ADCs to use the same clock signal and to be synchronized to the same clock cycle. There are two options for routing the clock signal: a single clock buffer to drive the clock signal to all ADCs, or individual clock buffers from one clock source to drive each ADC.

For either clock tree design, short clock trace lengths of no more than 5cm/ns of signal rise time generally do not need a source-termination series resistor. If this clock length is exceeded, source-terminate the clock trace with a series resistor to match the characteristic impedance of the micro-strip trace (minus that of the buffer output impedance). The source termination resistance absorbs reflected energy from the high-impedance clock inputs, keeping this energy from bouncing back to the inputs and reducing noise margin (difference between high or low input logic thresholds). With fast clock drivers, rise time can be much less than 1ns. Over-damping the PCB trace with a larger resistor value to reduce the clock signal overshoot and undershoot can increase the clock signal noise margin.

TI recommends a clock rise time of 1ns to reduce timing variances between multiple ADCs. These timing variances are a result of different logic threshold levels for the clock input due to process tolerances.

2.1 Single Clock Buffer

In systems with a small number of channels, a single clock buffer can be used to drive all ADCs. This is possible due to the small dimensions of the ADS1x7Lxx ADC, allowing for a small board layout resulting in short PCB trace lengths between the ADCs and the clock source. In a single-clock buffer layout, the PCB traces from the clock buffer to the ADCs must have equal path lengths to minimize sampling skew between the ADCs. Given the typical propagation delay of a micro-strip PCB design is 60ps/cm, a five-cm difference between the clock traces results in 300ps sampling skew between ADCs. This sampling skew appears as an additional 0.3ns of group delay in the input signal and must be included in the measurement error budget when phase angle between channels is important.

In addition to sampling skew, matched PCB trace lengths also reduce multiple line reflections from the ADC clock inputs. Multiple line reflections can lead to excessive ringing and overshoot, therefore reducing the clock signal noise margin.

Figure 2-1 shows an example of matched clock trace lengths. TI recommends this configuration for trace lengths of 5cm or less, and up to four ADCs. Using longer PCB traces or more ADCs result in excessive clock rise and fall times, reducing the clock signal noise margin.

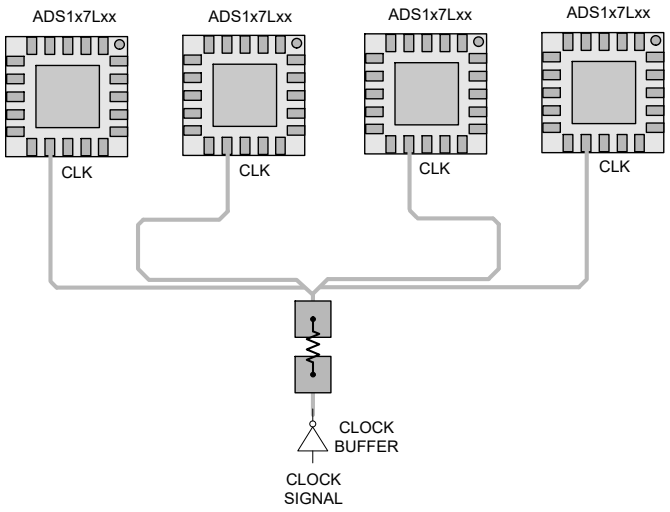


Figure 2-1. Single Clock Buffer

2.2 Multiple Clock Buffers

In systems with a large number of channels, using individual clock buffers for each ADC is conducive. However, clock buffers introduce channel-to-channel clock skew that causes sampling skew between the ADCs. A high-speed clock buffer such as LMK1C1104 has a well-specified 50ps channel-to-channel output skew specification. General-purpose logic buffers often have unspecified or excessively large output skew specifications of several nanoseconds. These general-purpose logic buffers can often be used for low clock speeds but must be evaluated for the specific application requirements. Figure 2-2 shows an example layout using multiple clock buffers. This is the preferred clock layout configuration with multiple ADCs and longer trace lengths.

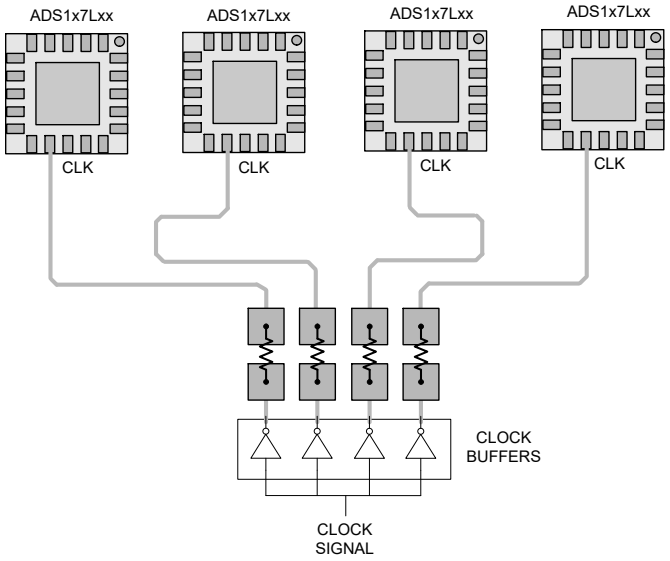


Figure 2-2. Multiple Clock Buffers

2.3 Clock Jitter

All delta-sigma ADCs operate on the principle of oversampling in which sampled data of the modulator is filtered and down-sampled by the digital filter. In addition to reducing thermal noise within the modulator by averaging data, the clock-jitter induced noise effects are averaged by the same process. The net effect of oversampling improves both the rated SNR due to thermal noise and the SNR_j due to clock jitter. Noise produced by jitter in the internal clock path is small relative to the device thermal noise. The reduction of jitter noise provided by oversampling is valid for uncorrelated clock jitter noise sources where the noise is assumed to be broadband.

Verify the clock signal is low jitter and free from glitches to achieve data sheet performance. Excessive clock jitter can lead to energy leakage or skirting effects around a large amplitude input signal and can obscure the ability to detect low level signals close to the large amplitude signal.

In general, RC oscillators integrated in controllers or the ADC do not have the required jitter performance to measure AC input signals and must be avoided. The LMK6C and CDC6C clock oscillators have very low jitter and meet the clock jitter requirements for all ADS1x7Lxx ADCs.

The amount of clock jitter that is acceptable is directly proportional to the input signal frequency and is inversely proportional to the root of the user-programmed oversampling ratio (OSR). Equation 1 and Figure 2-3 show the effect of clock jitter SNR_j.

$$SNR_j\left(dB\right) = -20 \times \log\left(\frac{2\pi \times f_{IN} \times t_j}{\sqrt{OSR}}\right) \quad (1)$$

where:

- SNR_j: theoretical SNR limit due to clock jitter (dB)
- f_{IN}: input signal frequency (Hz)
- t_j: clock jitter (s-rms)
- OSR: ADC oversampling ratio (refer to the specific ADC data sheet from Table 1-1 for OSR values)

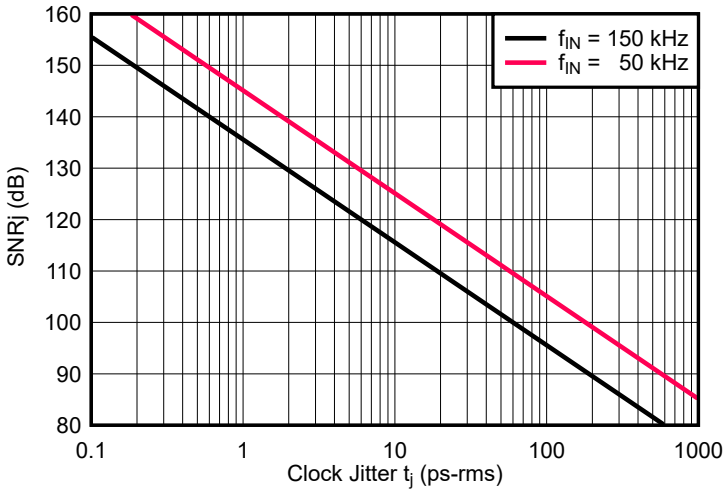


Figure 2-3. SNR_j Versus Clock Jitter

For example, with an input signal frequency = 150kHz and OSR = 32 (f_{DATA} = 400kHz), clock jitter is less than 10ps to yield SNR_j = 116dB. Preferably, SNR_j must be 6dB greater than the rated SNR of the ADC.

Correlated noise sources of clock jitter are not reduced by oversampling to the same degree as an uncorrelated noise source. To reduce one possible correlated noise source within single channel ADS1x7Lxx devices, operate the serial clock (SCLK) with a phase-coherent frequency to the ADC clock signal. Non-phase coherent SCLK and other clock frequencies present in the system can inter-modulate with the ADC clock signal producing unwanted sum and difference frequency multiples in the ADC output spectrum. Since data conversion transfer in multichannel ADCs use a frame-sync interface, the data transfer clocks are always phase-coherent, eliminating this possible correlated noise source when the SPI SCLK is not active.

Correlated types of clock signal jitter can occur elsewhere in the system and couple to the ADC clock signal through various mechanisms. These mechanisms include PCB ground plane noise that can be generated by a nearby switching power supply, capacitive and inductive coupling between parallel clock traces, or by routing unrelated clock signals through the same buffer package as the ADC clock signal. For single channel devices, shorten the SCLK and SDO/$\overline{\text{DRDY}}$ traces as much as possible and separate them from the clock signal to reduce coupling. SCLK can also be operated continuously to spread SCLK clock signal energy evenly over the entire conversion period.

3 Synchronization

Due to process variation of the internal power-up thresholds, ADCs in multiple device systems are initially out of synchronization until the ADCs are externally synchronized. Synchronization is also needed after changing the configuration of the device. The ADS1x7Lxx ADCs are synchronized by the START pin or by the SPI start bit. Do not use any ADC internal clock dividers when synchronizing multiple ADCs. If the internal clock divider is used, the resulting divided clock signal of each ADC can have a different phase, resulting in multiple clock cycle uncertainty between ADCs. All internal clock dividers must use the default value of 1 for proper synchronization.

The recommended method of synchronization is to use a single control line routed in parallel to each ADC START pin. The ADCs are synchronized at the next rising edge of the clock signal after START is asserted high. Assert START high on the falling clock edge to avoid the uncertainty of the rising clock edge used to latch START. Otherwise, a one clock period timing error between the ADCs can result. Also, always apply a continuous START signal on the falling clock edge when using synchronized control mode. The effects of START signal PCB trace-length mismatch are not as critical as matched clock traces because the ADC latches the START input on the rising clock edge, assuming the mismatch is less than ½ clock period.

Alternatively, route independent START lines to each ADC to make fine channel phase adjustments between ADCs with up to one clock cycle resolution. In this case, the clock signal skew between the ADCs is not a concern.

If the sync signal arriving to the system is asynchronous to the system clock and routed directly to the ADCs, one clock cycle of uncertainty can exist between the ADCs after synchronization. In this case, use an external circuit to synchronize the sync signal for all ADCs. The synchronization circuit releases the sync signal on the falling edge of the clock signal, making sure that the ADCs are synchronized to the same clock cycle. Figure 3-1 shows an example synchronization circuit that uses the SN74AUP2G79 dual D-type flip-flop.

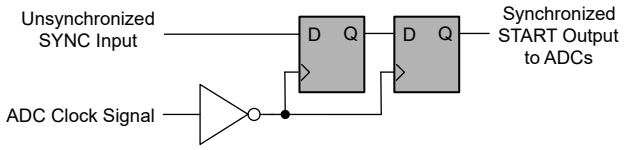


Figure 3-1. Synchronizing the SYNC Signal

Conversions can also be synchronized by the SPI for all control modes except synchronized control mode (refer to ADS1x7Lxx data sheet for details). To synchronize the ADCs via SPI, use a single chip select ($\overline{CS}$) signal connected to all ADCs. Shift in the register write command for the CONTROL register to set the START bit. When $\overline{CS}$ is taken high to end the frame, the ADCs operate on the command data simultaneously, resulting in systematic synchronization. To synchronize the ADCs to the same clock cycle, SCLK must be phase coherent to the ADC clock and $\overline{CS}$ must be taken high on the falling clock edge.

4 Anti-Alias Filter Group Delay

The digital filter used by the ADS1x7Lxx family is a linear-phase design such that all input signal frequency components are shifted the same amount of time. This behavior results in a linear-phase that is predictable and invariant as well as a constant group delay. A constant group delay maintains signal integrity for multichannel, simultaneous sampling systems, such as measuring acceleration amplitude and direction, where minimizing phase error is critical.

However, external analog anti-alias filters (AAF) can add nonlinear group delay within the signal pass-band. The AAF nonlinear group delay affects signals with *different frequency components* on the same channel, signals with the same input frequency applied to *different channels*, or both. The amount of non-linearity depends on the filter tuning and component matching.

For example, Figure 4-1 shows the group delay across input frequency for a 550kHz, 4th-order AAF critically damped filter alignment (13mdB peaking). This specific filter creates a 15ns group delay change from 0.575µs at 10kHz to 0.590µs at 100kHz. Applying a complex signal with 10kHz and 100kHz frequency components to this specific AAF input causes these frequencies to appear out of phase relative to each other at the AAF output.

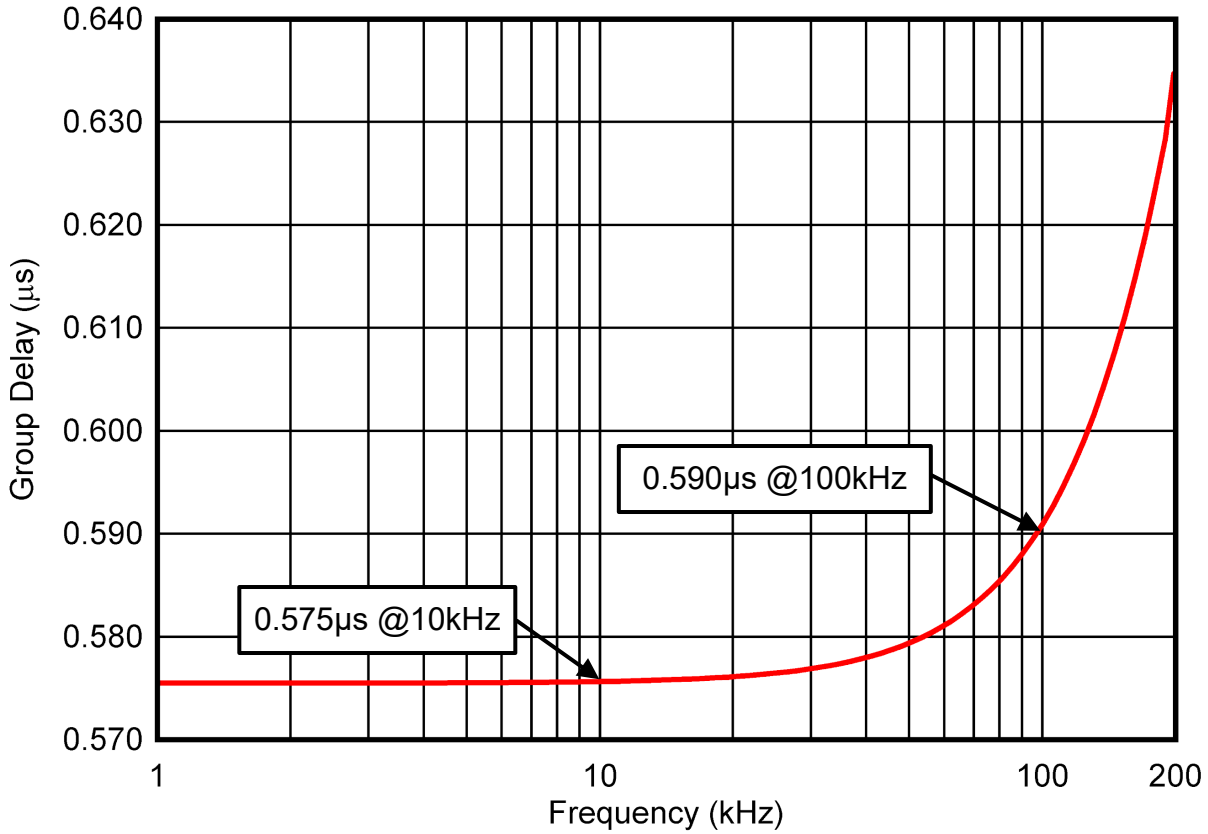


Figure 4-1. Anti-Alias Filter Group Delay

In multichannel systems, matched filter response between channels is important to maintain measurement accuracy, such as measuring acceleration amplitude and direction. Using a precision resistor and capacitor values are important to maintain similar response between channels, otherwise, the same signal on different input channels can have a different amplitude, phase, and group delay. Use 0.1% tolerance resistors and 1% capacitors for best results.

Moreover, an AAF can introduce additional phase errors in multichannel, simultaneous sampling systems due to component mismatch between channels. This behavior exists whether each channel measures a complex signal with multiple frequency components or a simple signal with a single frequency component.

Figure 4-2 shows an example of a multichannel system with an AAF at the input of each ADC channel. These filters have the exact same properties as the AAF used in the previous example and are identical other than the capacitors at the ADC inputs. As shown, the capacitor (C05) on Channel 1 has a preferable capacitance of 2.2nF while the 2.222nF capacitor (C75) on Channel 8 includes a 1% tolerance variation.

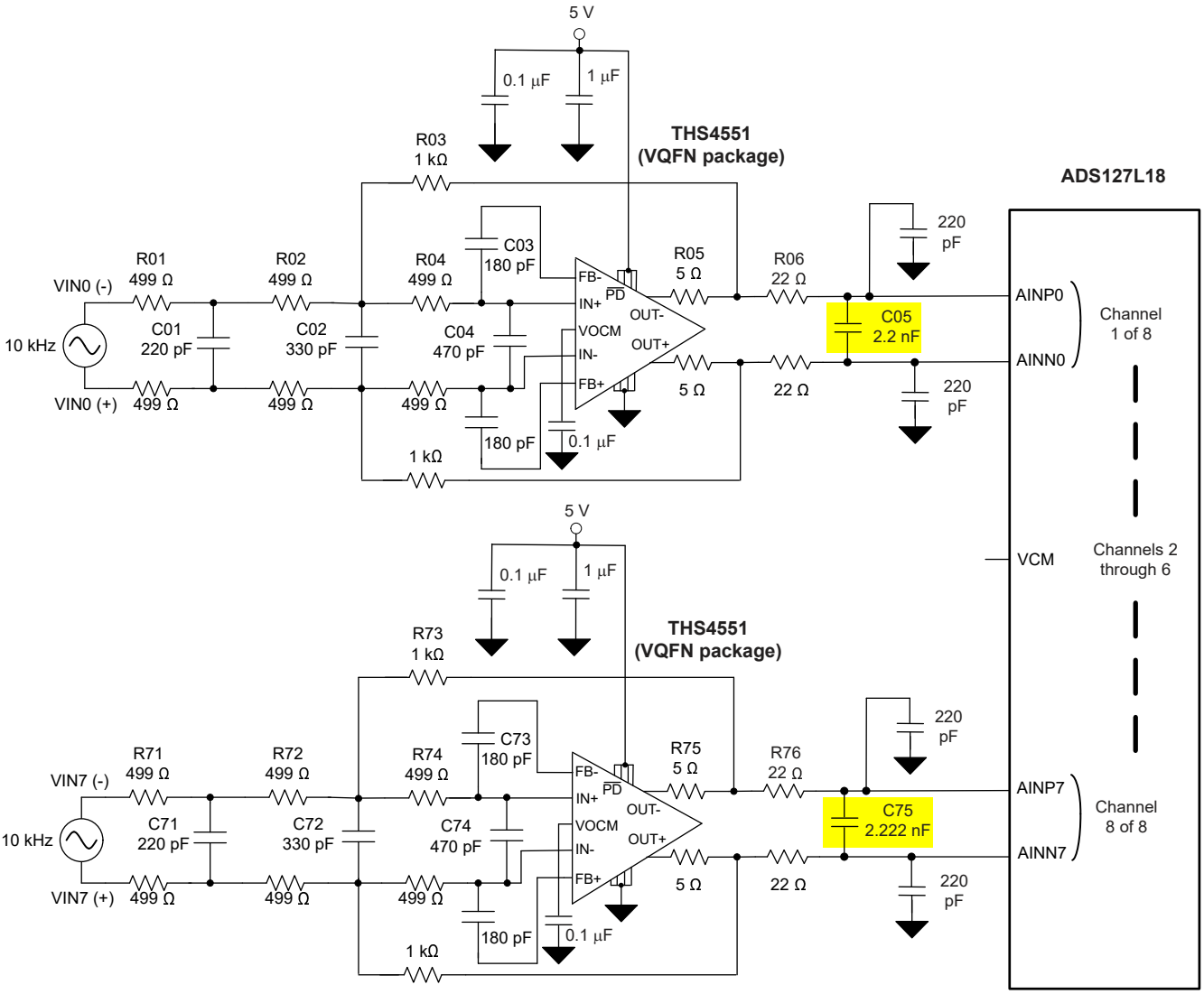


Figure 4-2. Anti-alias Filter Channel to Channel Tolerance Variations

The 1% variation between the two capacitors shown in Figure 4-2 increases the channel-to-channel group delay at 10kHz from 0.575µs to 0.576µs. Accounting for all resistor and capacitor tolerances yields a group delay variation at 10kHz of 0.573µs to 0.578µs, assuming 0.1% resistors and 1% capacitors. Therefore, use precision resistor and capacitor values to minimize channel-to-channel differences in signal amplitude, phase, and group delay.

5 Reference Voltage

Multiple ADCs can share a single voltage reference and meet data sheet specifications. Use local decoupling capacitors at the ADC REFP and REFN pins in addition to the large bulk decoupling capacitor at the voltage reference output. TI recommends enabling the ADS1x7Lxx internal REFP buffer to reduce loading of the voltage reference.

Route REFP and REFN as a differential pair with the same care as routing the input signal. Route a PCB trace 40mil(1mm) wide or greater starting at the voltage reference ground pin to the REFN pins using no intervening ground ties. Use of ground ties to the REFN trace can result in ground noise pick-up in the reference signal. Figure 5-1 shows example REFP and REFN routing and reference input pin bypassing at the ADCs.

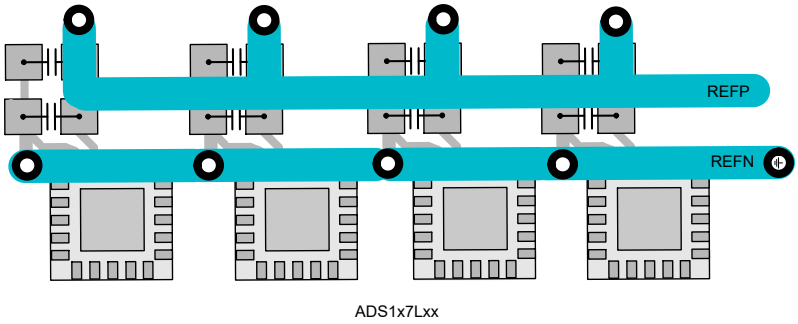


Figure 5-1. Reference Voltage Routing

6 Power Supply Bypassing and Grounding

For best performance in systems using multiple ADCs, follow the same power supply bypassing and ground plane recommendations used for single ADC systems. Refer to the ADS1x7Lxx device family data sheets for power supply bypassing recommendations. A dedicated, unbroken ground plane layer with shared analog and digital grounds for the ADCs in multichannel systems provides the best results in most cases. Figure 6-1 shows an example layout using a solid ground plane.

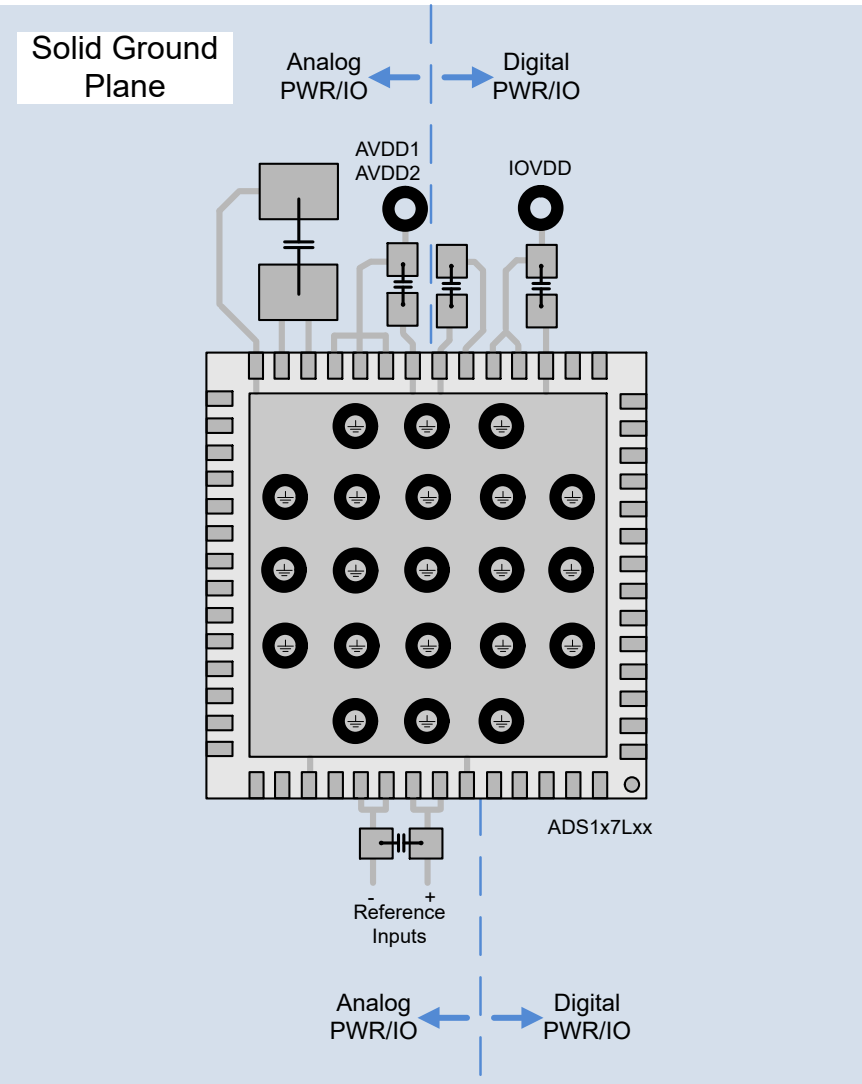


Figure 6-1. ADS1x7Lxx Example Board Layout

7 SPI Daisy-Chain Connection

One challenge with using multiple ADCs is the added complexity of the digital signal routing between ADCs and the host controller. The ADS1x7Lxx family offers a daisy-chain option to streamline the digital signal connections. The single channel ADCs use SPI for both data and configuration. The multichannel ADCs use SPI for configuration only and a frame-sync port for conversion data. The following sections discuss daisy-chain of the SPI port only. Frame-sync daisy-chain is later discussed in System Requirements for Frame-Sync Daisy-Chain Configuration.

The daisy-chain option connects the SPI data output of one ADC into the SPI data input of the next ADC in the chain. These connections effectively link the individual ADC shift registers into one longer-length shift register. From an SPI perspective, the host controller interfaces to the chained devices as a single virtual device. Figure 7-1 illustrates how the individual ADC shift registers emulate a single shift register through the daisy-chain connection.

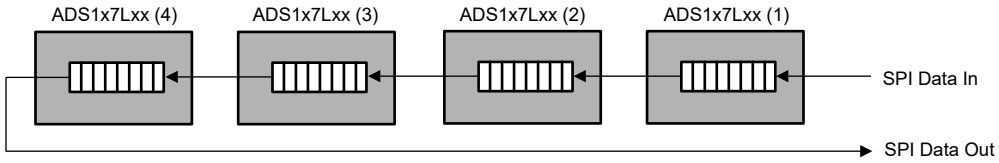


Figure 7-1. Daisy-Chain Concept

The daisy-chain connection keeps the number of SPI lines to the host controller to four, regardless of the number of ADCs connected in the chain. By comparison, a standard SPI cascade connection requires seven SPI lines to interface to four ADCs.

Figure 7-2 and Figure 7-3 show the required SPI signals for a standard SPI cascade connection and a daisy-chain connection, respectively. If using either standard SPI cascade or daisy-chain connection, additional control lines are potentially necessary. Additional control lines are often used for ADC synchronization (START), ADC reset ($\overline{\text{RESET}}$), and in the case of single channel ADCs, data-ready output ($\overline{\text{DRDY}}$).

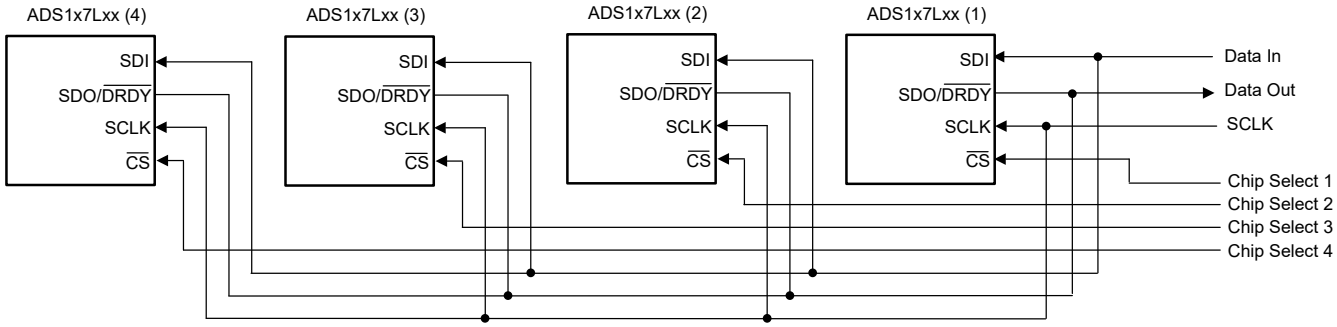


Figure 7-2. Standard SPI Cascade Connection

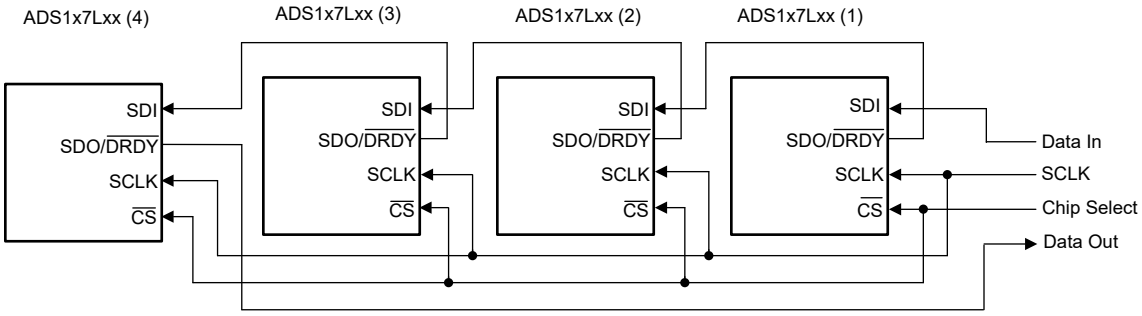


Figure 7-3. Daisy-Chain SPI Connection

7.1 SPI Daisy-Chain Communication

Daisy-chain operation for the ADS1x7Lxx family requires no special programming. The user configures the host controller to extend the data frame to the length needed to access the data from all ADCs connected in the chain. The new data frame length is set to match the number of devices in a chain multiplied by the number of bits per frame of the ADC. For example, four devices in a chain using 24b ADC data packets require the user to set the controller frame length to ninety-six bits.

When shifting data into the daisy-chain, the first block of data is targeted for the last device in the chain connection (ADC 4 in Figure 7-3). The ADCs only interpret the data in the respective shift registers when $\overline{CS}$ is taken high. This means there is no limit to the amount of shift operations as the data passes through each ADC, only the last bits shifted into each ADC matter. Figure 7-4 shows an example of 24b input data packets per each ADC to match the 24b output data size. See the ADS1x7Lxx family data sheets for additional details regarding the daisy-chain input command format.

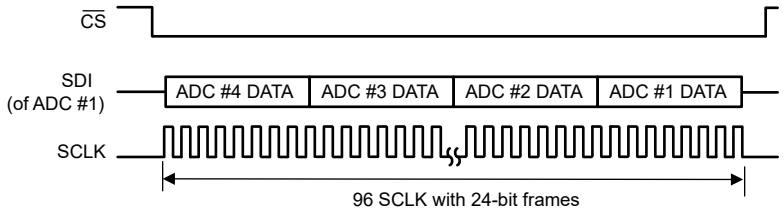


Figure 7-4. SPI Daisy-Chain Data Input Sequence

When reading data from the ADC, the first data output on SDO/$\overline{DRDY}$ is from the last device in the chain (ADC 4 in Figure 7-3), followed by data from the next device in the chain (ADC 3), and so on (see Figure 7-5). There is no interruption or gap in the data stream between devices. For example, if the ADCs are programmed for 24b SPI data packets, perform ninety-six shift operations to read data from four devices.

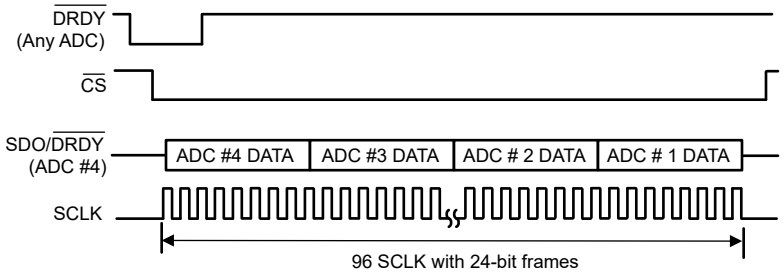
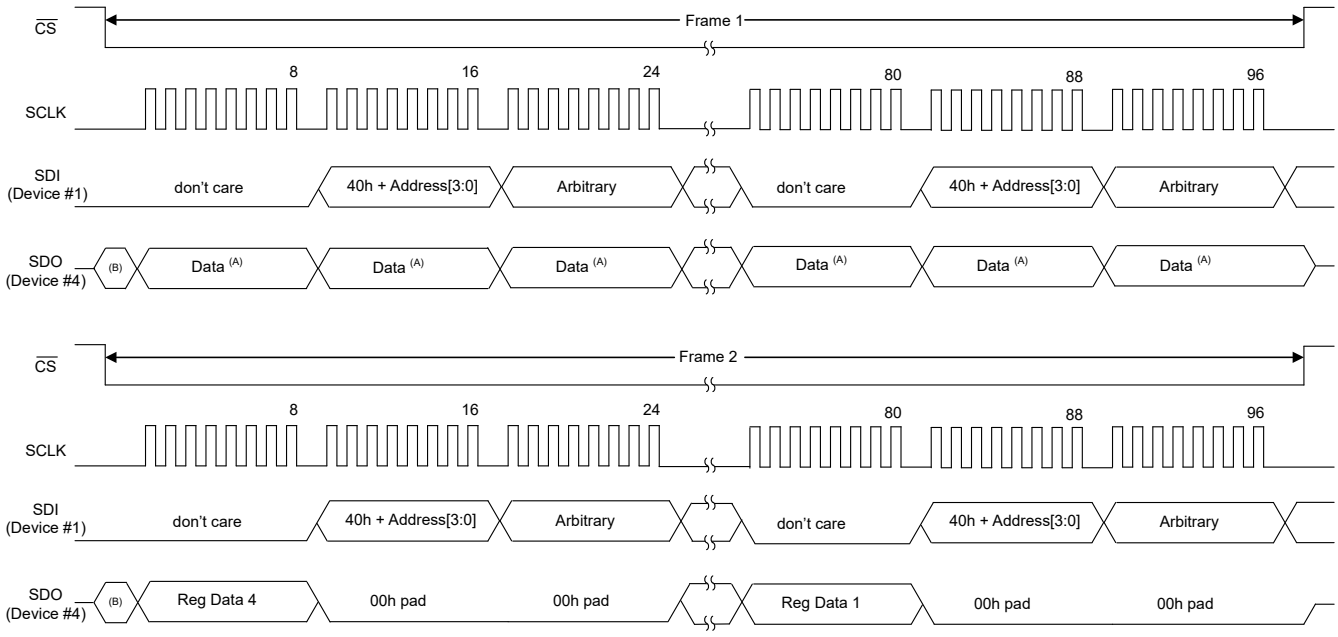


Figure 7-5. SPI Daisy-Chain Data Output Sequence

Reading register data requires two frames. The first frame inputs the read register command with the general format shown in Figure 7-4. The second frame outputs the register data, with the general format shown in Figure 7-5. The first register output data is from device #4 with two additional bytes inserted after the register data byte. The additional bytes fill out the individual ADC frames to equal the 24b data size. Even though the total number of register data bytes is four (one byte from each ADC), ninety-six shift operations are needed to read the register data from all the ADCs. The additional bytes can be 00h pad bytes, or other values depending on the specific ADC and configuration. See the ADS1x7Lxx data sheet for details on the exact contents of the returned data. Figure 7-6 is an example of a 4 ADC daisy-chain read register data, with 24b frame size per ADC.



A. Depending on the previous operation, the data field is either conversion data or register data + two 00h pad bytes.
B. Previous state of SDO/$\overline{\text{DRDY}}$ before the first SCLK.

Figure 7-6. Daisy-Chain Read Register Data

7.2 System Requirements for SPI Daisy-Chain Configuration

There are several requirements when the ADCs are configured in a daisy-chain:

1. Use the four-wire SPI mode and one $\overline{\text{CS}}$ control line to simultaneously select and deselect the ADCs in the chain.
2. For single-channel devices the SDO/$\overline{\text{DRDY}}$ pin must be programmed in the data-output only mode. Do not use the SDO/$\overline{\text{DRDY}}$ dual function mode with daisy-chain operation.
   a. The ADS117L11 and ADS127L11 ADCs default to the data-output only mode after reset and can be used in a daisy-chain configuration without additional register setting changes.
   b. The ADS127L21 and ADS127L21B default to dual SDO/$\overline{\text{DRDY}}$ mode after reset and must be reconfigured to data-output only mode by writing 00b to the DATA_MODE field in the FILTER3 register. For the following procedure, refer to Figure 7-3.
      i. First, program ADC1 SDO/$\overline{\text{DRDY}}$ to data-output only mode using a register write frame.
      ii. Once ADC1 is programmed correctly, ADC2 can then be programmed in the next register write frame.
      iii. Continue this procedure until all ADCs in the daisy-chain have been programmed to data-output only mode.
      iv. Note that each ADC in the daisy chain requires a separate write register frame, or 4 register write frames for 4 ADCs.
3. To reduce the complexity of interfacing to the ADCs, program the ADCs to the same data packet length using a parallel write operation. Program individual data packets as 16, 24 32, or 40 bits for single channel devices or 16 or 24 bits for multichannel devices.
4. Install pull-up resistors between the combined SDI-SDO connections and IOVDD. These resistors prevent the SDI inputs from floating in any condition because $\overline{\text{CS}}$ tri-states SDO.

7.3 Number of Devices in a SPI Daisy-Chain Connection for Single Channel ADCs

The following section is only applicable to single-channel devices. For multichannel devices, refer to Number of Channels in a Frame-Sync Daisy-Chain Connection.

The maximum number of single channel devices connected in a chain is limited by the SPI clock speed, the length of the ADC data frame, and the ADC data rate. In other words, the SPI clock speed must be fast enough to read data from all devices in one conversion cycle or data is lost. This requirement is also true using the standard SPI cascade connection because data is also read sequentially in this mode.

The single channel ADS1x7Lxx devices support SCLK speeds up to 50MHz. However, achieving 50MHz operation requires a non-standard SPI timing configuration, where data is clocked out and clocked in on the same clock edge. This non-standard SPI timing configuration is not supported in daisy-chain mode. Using standard opposite edge clock-out and clock-in SPI operation, and accounting for SPI propagation delay and setup time, the SCLK speed is limited to approximately 16.5MHz in a daisy-chain configuration. Operating IOVDD at 2V or greater (only possible with the single channel ADCs) reduces the propagation delay time, increasing the maximum SCLK speed to approximately 20MHz.

Equation 2 shows that the number of ADCs connected in a single daisy-chain is determined by the SCLK frequency, the data rate, and the bits per frame of each ADC.

$$\text{Maximum number of devices in a daisy - chain connection} = \left\lfloor \frac{f_{SCLK}}{f_{DATA} \times \text{bits per frame}} \right\rfloor \quad (2)$$

For example, if f_{SCLK} = 20MHz, f_{DATA} = 100kSPS and the ADC outputs 24 bits per frame, the number of devices in a single daisy-chain is limited to the floor of: 20MHz / (100kHz × 24) = 8.

If the maximum number of devices is less than desired according to Equation 2, the number of devices can be increased by using another daisy-chain with a separate data output line (SDO/$\overline{\text{DRDY}}$). The two data output lines enable parallel output-data shift operations from the two daisy-chains. $\overline{\text{CS}}$, DIN and SCLK lines can be shared between the daisy-chains to keep the number of SPI lines at a minimum. Figure 7-7 illustrates this configuration.

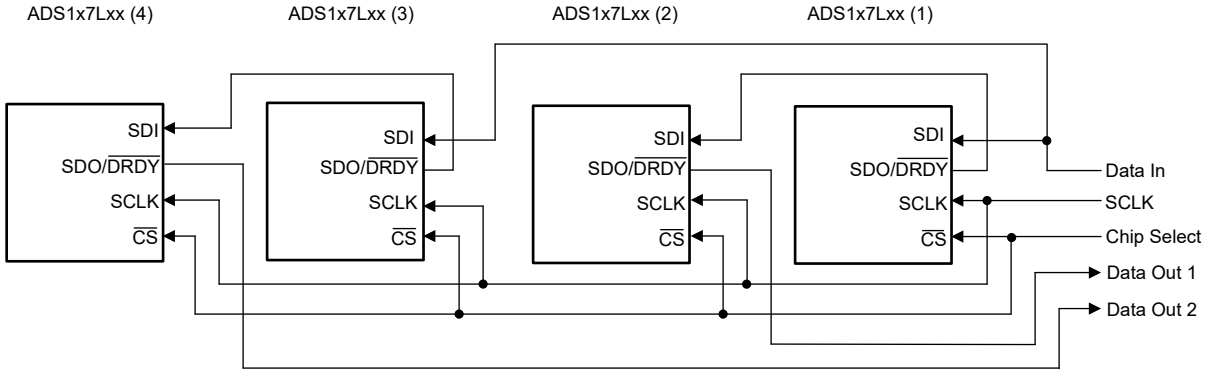


Figure 7-7. Daisy-Chain SPI Connection with 2 Parallel Output

8 Parallel SPI SDO or DRDY Connection for Single Channel ADCs

The following section is only applicable to single-channel devices. For multichannel devices, see Frame-Sync Daisy-Chain Connection for Multi-Channel ADCs.

In systems where multiple ADCs are sampling at high data rates such as 1MSPS, the conventional SPI cascade connection, Figure 7-2, and the daisy-chain connection, Figure 7-3, cannot provide the required data throughput since both configurations use a single SPI port. In this case, connect each SDO/$\overline{\text{DRDY}}$ pin in parallel to the host controller. To reduce the number of connections, the remaining signals can use a common $\overline{\text{CS}}$, SCLK, and SDI signal for each ADC. The main disadvantage of the parallel SDO/$\overline{\text{DRDY}}$ connection is the requirement for an individual SPI port for each ADC in the system.

Figure 8-1 shows an example of four ADCs using four SDO connections to the host. This example uses a single $\overline{\text{CS}}$, SCLK, and SDI control line such that all ADCs must have the same device configuration. Alternatively use separate $\overline{\text{CS}}$ control lines to enable the ADCs for independent input communication.

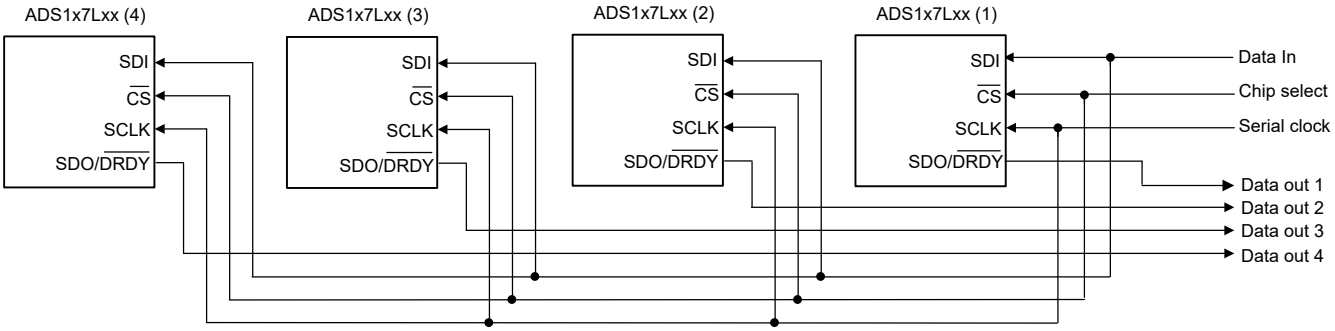


Figure 8-1. Parallel SDO/$\overline{\text{DRDY}}$ Connection

9 Determining When New Conversion Data is Available for Single Channel ADCs

The user can determine when new ADC data is ready using several methods:

1. Monitor the $\overline{\text{DRDY}}$ signal.
2. Count ADC clocks.
3. Poll the DRDY bit of the STATUS byte.

As mentioned in Section 7.2 the SDO/$\overline{\text{DRDY}}$ signal cannot be used for data ready monitoring when connected in daisy-chain configuration.

After the ADCs are synchronized, monitoring one $\overline{\text{DRDY}}$ output signal from any chosen ADC is sufficient. In some cases, the user can choose to monitor all the $\overline{\text{DRDY}}$ outputs to verify the ADCs are synchronized. For example, the user must monitor each individual $\overline{\text{DRDY}}$ pin to verify phasing between devices when an intentional phase offset is used between ADCs. For these circumstances, monitor the $\overline{\text{DRDY}}$ outputs from each ADC.

Alternatively, count clock cycles after synchronizing to predict when to read the conversion data. The digital filter restarts at synchronization, requiring additional time to produce the first conversion result. The net time delay for the first conversion result is specified as latency time in the relevant ADC data sheet from Table 1-1.

Conversion data ready can also be determined by software polling the STATUS byte. Conversion data is new (or ready) when the DRDY bit asserts high. Use the register read command to read the STATUS byte, or continuously read conversion data with the STATUS byte enabled and poll the DRDY bit. When the DRDY bit is set, the data is new since the last conversion data read. Using either of these methods requires additional SPI frames, or increased frame size, reducing the maximum data rate that can be supported.

10 Frame-Sync Daisy-Chain Connection for Multi-Channel ADCs

The ADS1x7Lxx multichannel ADCs use a dedicated frame-sync data port for conversion results that is separate from the SPI port. Instead, the SPI port is only used to read and write register configuration data. However, similar to SPI, the frame-sync data port can also be connected in a daisy-chain configuration.

The frame-sync data port for the ADS1x7Lxx multichannel family operates in controller mode such that the ADC generates both the data clock (DCLK) and the word clock (FSYNC). Additionally, the ADC can provide data on 1, 2, or 4 data-lanes (DOUT0, DOUT1, DOUT2, and DOUT3) in a daisy-chain configuration. While using all 4 data-lanes does help increase the throughput, this method comes at the cost of requiring more physical connections to the ADC.

To use frame-sync daisy-chain mode, the DP_DAISY bit in DP_CFG1 register must be set to 0b, which is also the default value. In this configuration (1, 2, or 4 data lanes), DINx data are shifted in and appended to the original channel data. Refer to the ADS1x7Lxx family data sheet for additional details.

Similar to SPI daisy-chain, frame-sync daisy-chain effectively links individual ADC shift registers into longer length registers. Figure 10-1, Figure 10-2, and Figure 10-3 illustrates these configuration options for 1, 2, and 4 data lanes, respectively.

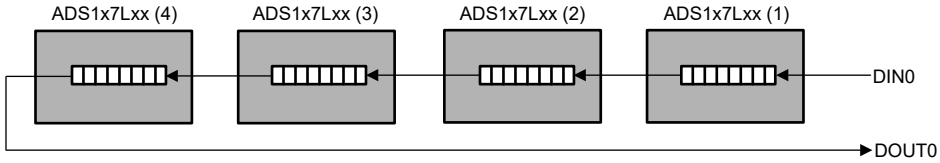


Figure 10-1. Frame-Sync One-Lane Daisy-Chain Concept

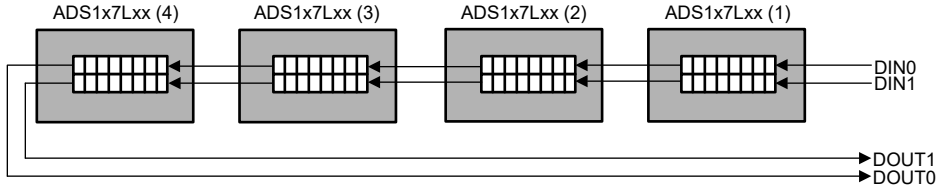


Figure 10-2. Frame-Sync Two-Lane Daisy-Chain Concept

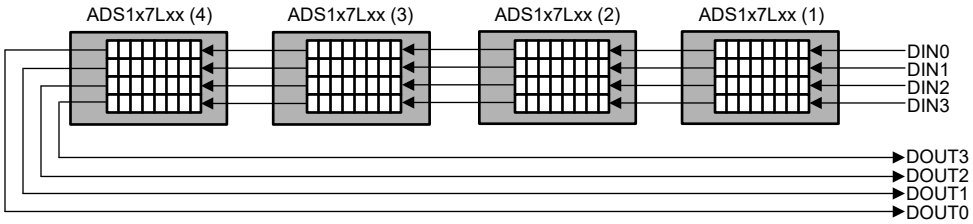


Figure 10-3. Frame-Sync Four-Lane Daisy-Chain Concept

Using a daisy-chain connection reduces the number of IO pins needed on the host controller and simplifies the routing of the board. The daisy-chain frame-sync signals include 1 DCLK line, 1 FSYNC line, and 1, 2, or 4 DOUT lines. In addition to the frame-sync signals, a common CLK and START signal are needed for proper synchronization. Additional hardware lines can be used as well, especially if using Hardware Programming mode.

Figure 10-4, Figure 10-5, and Figure 10-6 show the frame-sync daisy-chain connections for one-lane, two-lane, and four-lane configurations, respectively. Note that only one FSYNC and one DCLK connection to the frame-sync receiver is needed because all ADCs are synchronized for daisy-chain and all FSYNC and DCLK signals have the same frequency and phase.

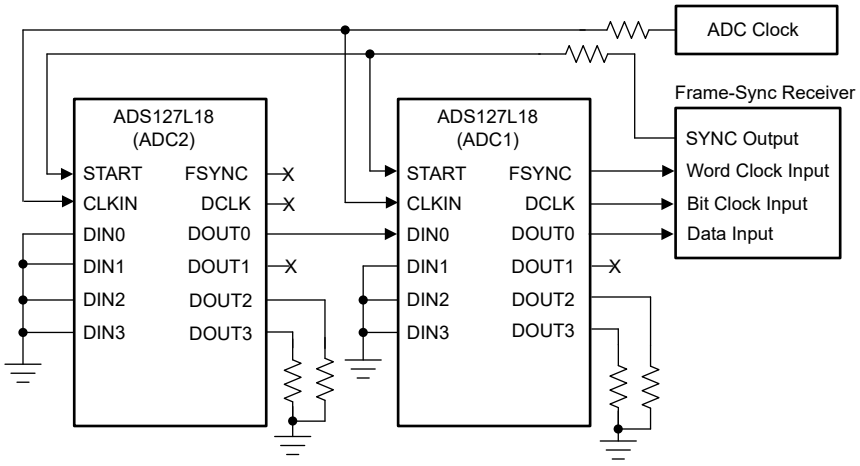


Figure 10-4. Frame-Sync Daisy-Chain One-Lane

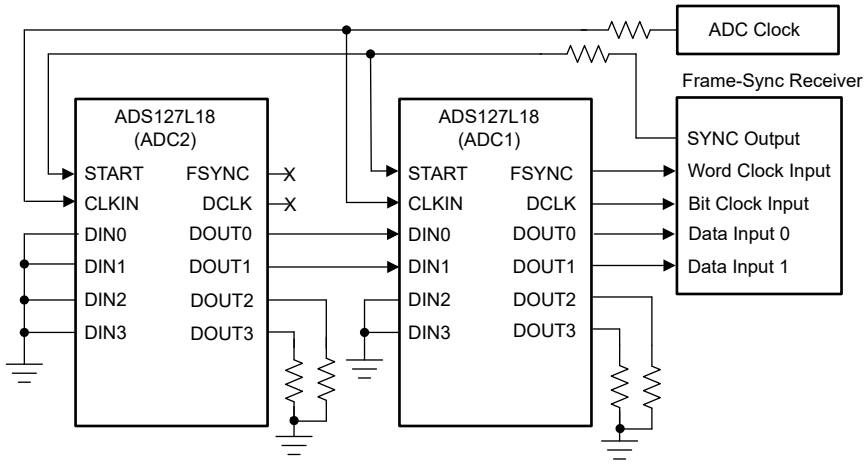


Figure 10-5. Frame-Sync Daisy-Chain Two-Lane

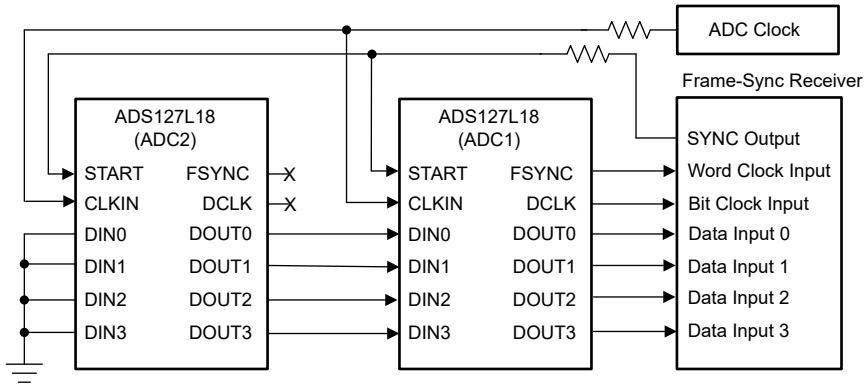


Figure 10-6. Frame-Sync Daisy-Chain Four-Lane

Select the desired number of data lanes by writing to the DP_TDM[1:0] field in the DP_CFG1 register. Figure 10-7 shows a typical ADS1x7L18 data transfer for the one data-lane configuration shown in Figure 10-4. See the ADS1x7Lxx family data sheets for additional details regarding these configurations and the resulting data output sequence.

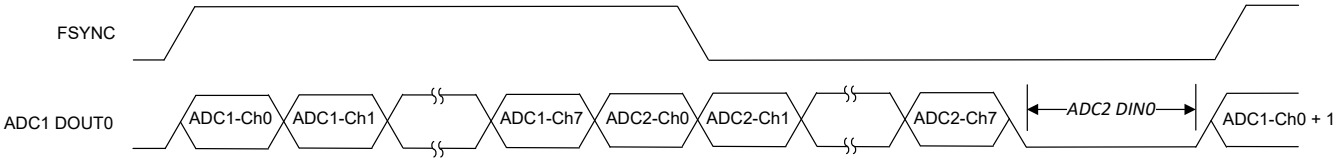


Figure 10-7. One-Lane Daisy-Chain Data

10.1 System Requirements for Frame-Sync Daisy-Chain Configuration

There are several requirements when the multichannel ADCs are configured in daisy-chain. These requirements are specific to the ADS117L14, ADS117L18, ADS127L14, and ADS127L18 devices. All register settings not specified in the following steps are default reset values.

Register Configuration Settings:

1. DP_DAISY = 0b in the DP_CFG1 register. (Default setting for SPI and Hardware Programming Mode)
2. CLK and DCLK dividers must be programmed to the divide by 1 option. CLK_DIV[2:0] = 000b in the CLK_CFG register and DCLK_DIV[1:0] = 00b in the DP_CFG2 register. (Default settings for SPI and Hardware Programming Mode)
3. Set CLK_SEL = 1b in the CLK_CFG register. External clock operation is required for frame-sync daisy-chain operation. Hardware Programming Mode uses External Clock by default. SPI Programming uses Internal Clock by default such that the user must set the CLK_SEL bit to 1.
4. DP_TDM[1:0] (TDM mode or number of data lanes) is programmed the same for all devices in the chain.
5. Reduce the complexity of interfacing to the ADCs by programming each device for the same frame length with a parallel write operation. (Program individual frame length as 16, 24, 32, or 40 bits for all devices)

Hardware Configuration Settings:

1. The devices are synchronized together using the START pin.
2. The START pin rising edge must meet the setup and hold timing requirements relative to CLK specified in the timing requirements section of the data sheet. If not met, then frame-sync daisy-chain data is corrupted.
3. Use 100kΩ pull-down resistors on unused DOUT2 and DOUT3 pins. Leave DOUT1 floating if unused. Connect unused DIN pins to GND or IOVDD.
4. All digital signals for the multichannel ADCs support 1.8V logic levels only. These devices are damaged by voltages greater than 2.2V.

Figure 10-8 shows an example of two multichannel ADCs in daisy-chain configuration using SPI Programming Mode. This configuration can be further expanded to include more than two multichannel ADCs.

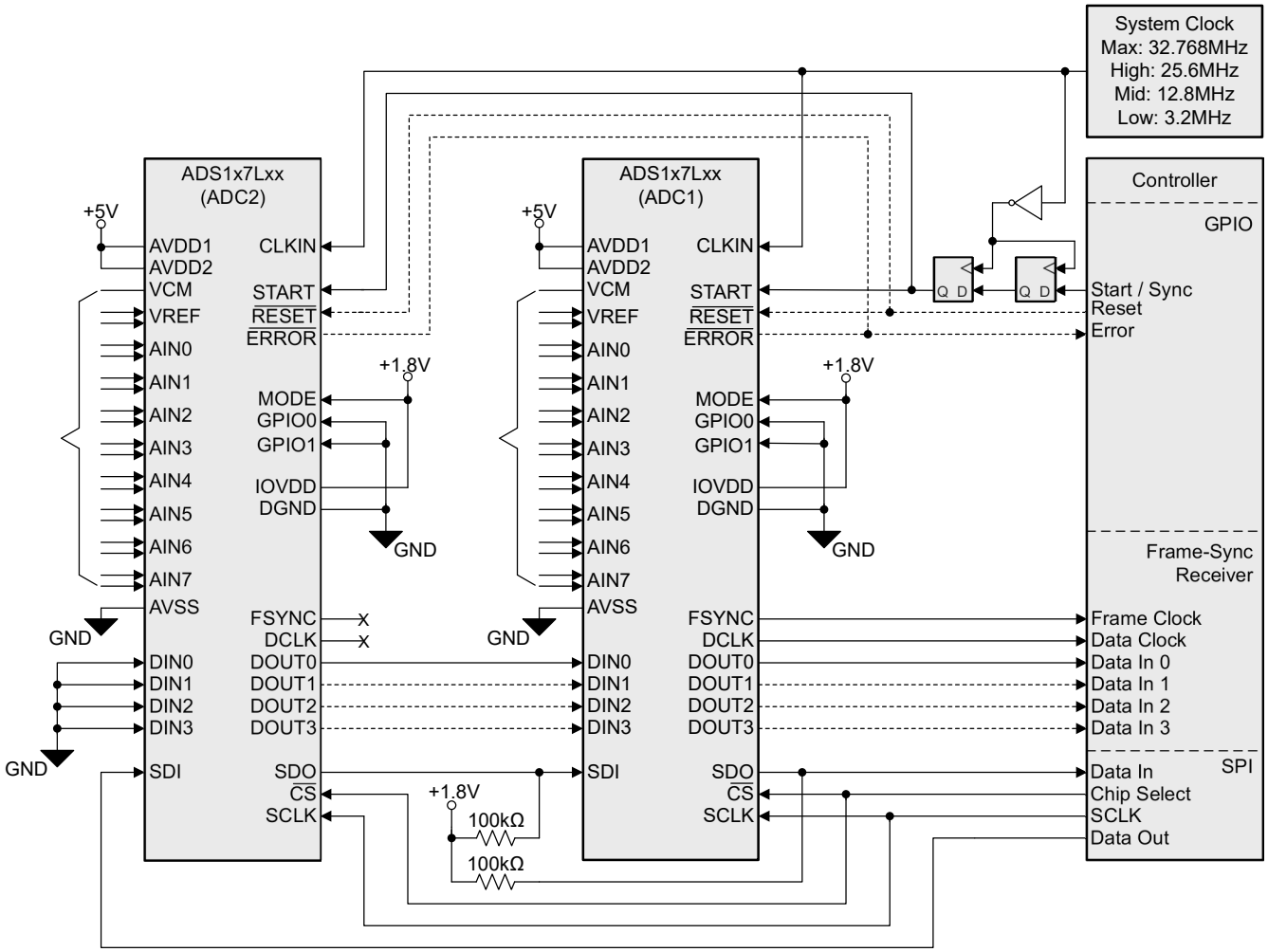


Figure 10-8. Frame-Sync Daisy-Chain with SPI Programming Mode

Notes:

1. START signal must meet setup and hold time relative to CLKIN rising edge.
2. If RESET is not connected to controller, then connect to IOVDD.
3. If ERROR pin is used, connect 100kΩ pull-up to IOVDD.
4. If DP_TDM[1:0] = 00b or 01b, connect 100kΩ pulldown resistors from DOUT2 and DOUT3 to GND.
5. Unused DIN0, DIN1, DIN2, and DIN3 pins must be connected to GND or IOVDD.
6. DOUT1 is always an output and must be left floating if not used.

10.2 Number of Channels in a Frame-Sync Daisy-Chain Connection

The maximum number of channels supported in a frame-sync daisy-chain configuration is limited by the length of the ADC data packet size, number of DOUT data lanes, and the ADC over-sampling ratio (OSR) setting. The data packet size is determined by the conversion data size (16b or 24b) and optional STATUS and CRC bytes. This results in a packet size of 16b, 24b, 32b, or 40b.

As discussed in System Requirements for Frame-Sync Daisy-Chain Configuration, program the CLK and DCLK dividers to the divide by 1 setting. This results in the data clock (DCLK) frequency equal to the ADC clock (CLK) frequency.

Depending on system requirements, choose the OSR setting to meet the specific data rate, noise level, and power consumption level. The data rate depends on the clock (CLK) frequency and OSR setting and can be calculated per Equation 3.

$$\text{Data rate} = \frac{\text{clock frequency}}{2 \times \text{OSR}} \tag{3}$$

Refer to the filter noise tables in the ADS1x7Lxx data sheet for more details. Once OSR and data packet size are determined, the maximum number of channels can be calculated per Equation 4.

$$\text{Maximum number channels} = \text{data lanes} \times \left\lfloor \frac{2 \times \text{OSR}}{\text{data packet}} \right\rfloor \tag{4}$$

where:

- data lanes: number of data lanes, valid values of 1, 2, or 4 only
- OSR: Over Sampling Ratio, refer to the filter noise tables in ADS1x7Lxx data sheet for available values
- data packet: number of bits in a data packet, values of 16, 24, 32, or 40 bits

In the following example, assume the user must support a data rate of 50kSPS with the lowest noise. In this case, operation in high-speed mode with an OSR = 256 and f_{CLK} = f_{DCLK} = 25.6MHz meets the user requirements. The user also needs 24b data and plans to include the STATUS byte. The total data packet size is 32 bits. The user also wants to minimize the number of connections to the system controller (MCU or FPGA) by using a single data lane. Using Equation 5 results in a maximum number of channels supported equal to (or 16 channels total):

$$1 \times \lfloor (2 \times 256/32) \rfloor \tag{5}$$

Assuming the user selects the 8-channel ADS127L18, two ADS127L18 ADCs can be connected in daisy-chain for a total channel count of sixteen.

11 Summary

Data acquisition systems commonly require multiple ADCs operating in parallel. Designing these multi-device systems requires special consideration regarding the clock signal, synchronization, and voltage reference. The clock signal must be low jitter and routed to the ADCs using recommended best practices. Following these guidelines minimizes clock jitter noise and reduces interference from other clock signals. Synchronizing the ADCs after power-on and after the ADC configuration is changed by asserting the START pin high on the clock signal falling edge is important. The reference voltage ground must be tied at a single point at the reference voltage ground terminal. Daisy-chaining the ADCs can be effective in simplifying the number of SPI I/O connections and frame-sync I/O connections between ADCs and the host controller. Parallel SDO/$\overline{\text{DRDY}}$ connections increase data throughput for single-channel ADCs by providing the ability to clock output data in parallel. Parallel DOUT connections for frame-sync multichannel ADCs also increase data throughput by providing the ability to clock output data in parallel.

12 References

1. Texas Instruments, *ADS117L11 400-kSPS, Wide-Bandwidth, 16-Bit, Delta-Sigma ADC*, datasheet.
2. Texas Instruments, *ADS127L21 512-kSPS, Programmable Wideband Filter, 24-Bit, Delta-Sigma ADC*, datasheet.
3. Texas Instruments, *ADS127L21B 512kSPS, Ultra-Precision, 24-Bit, Wideband Delta-Sigma ADC*, datasheet.
4. Texas Instruments, *ADS117L1x 512kSPS, Quad and Octal, Simultaneous-Sampling, 16-Bit ΔΣ ADCs*, datasheet.
5. Texas Instruments, *ADS127L1x 512kSPS, Quad and Octal, Simultaneous-Sampling, 24-Bit ADCs*, datasheet.
6. Texas Instruments, *ADS127L11 400-kSPS, Wide-Bandwidth, 24-Bit, Delta-Sigma ADC*, datasheet.

13 Revision History

Changes from Revision * (July 2021) to Revision A (December 2025)	Page
• Updated to include multichannel ADCs	2
• Added frame-sync data port in daisy-chain applications	2
• Updated format to a full application note	2
• Updated the numbering format for tables, figures, and cross-references throughout the document	2
• Updated the title of the document	2
• Added new reference and corrected existing reference	23

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