Application Brief

ADS127L11 in Simultaneous-Sampling Systems

Overview

In data acquisition systems such as collecting data from sound and vibration sensors, data from the electrical grid and data from medical equipment such as EEG, multiple ADC channels operating in simultaneous-sampling mode are often implemented. Based on the small package size, daisy-chain connection option and ability to synchronize, the ADS127L11 is suited for these types of applications. This application brief reviews clocking, synchronizing and daisy-chain considerations for simultaneous-sampling systems using the ADS127L11.

Clock Signal

The basic requirements of a simultaneous-sampling system are to use the same clock signal for all ADCs and to synchronize the ADCs 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 the ADCs, or individual clock buffers from one clock source to drive each ADC.

Single Clock Buffer

Due to the small dimensions of the ADC, in many cases it is straightforward to design the clock signal using a single clock buffer to drive the ADCs. In a single-clock buffer layout, the PCB traces from the clock buffer to the ADCs should have equal path lengths to minimize sampling skew between the ADCs. Given the typical propagation delay of a microstrip PCB design is 150 ps/in, a one-inch difference between the clock traces results in 150 ps sampling skew between ADCs. See Figure 1 as an example of matched clock trace lengths.

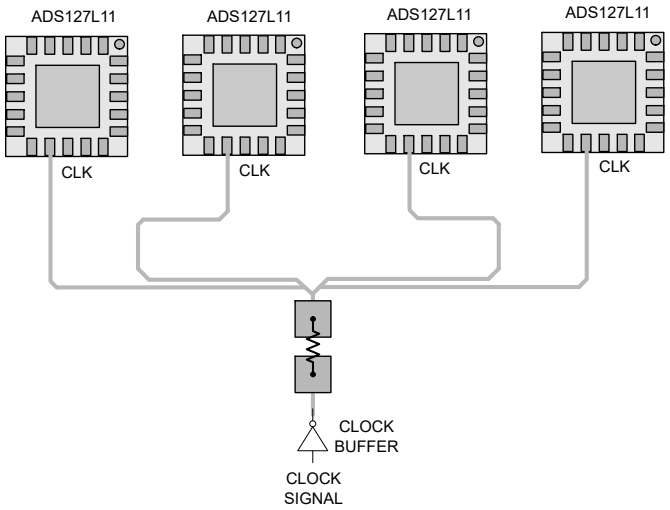

Figure 1. Single Clock Buffer

Perhaps just as an important advantage of matched PCB trace lengths is to eliminate the multiple line reflections from the ADC clock inputs. Multiple line reflections normally occur using unmatched trace lengths which can lead to excessive ringing and overshoot, therefore reducing the noise margin (difference between high or low input logic thresholds) of the clock signal.

Multiple Clock Buffers

In some cases, it can be conducive to use individual clock buffers for each ADC. However, the clock buffers can introduce channel-channel clock skew, which will contribute sampling skew between the ADCs. A high-speed clock buffer such as LMK1C1104 has a well-specified 50-ps channel-to-channel output skew specification. General-purpose logic buffers often have unspecified or excessively large output skew specifications; therefore they should be evaluated to achieve the desired channel-to channel phase match requirement. See Figure 2 for an example layout using multiple clock buffers.

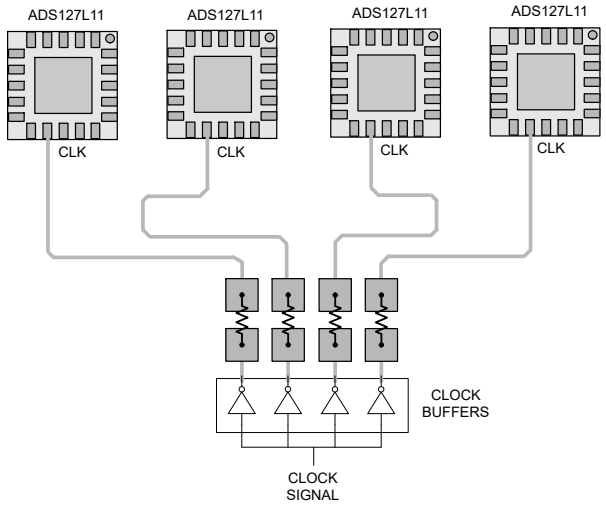

Figure 2. Multiple Clock Buffers

As a general rule of thumb for either clock tree design, the clock trace length should be no more than 2 in/ns of signal rise time. If exceeded, source-terminate the clock trace with a series resistor to match the characteristic impedance of the microstrip trace (minus that of the buffer output impedance). The source termination resistance absorbs reflected energy from the high-impedance clock inputs from bouncing back to the inputs that will negatively impact the noise margin. With fast clock drivers (< 1ns rise time), overdamping the PCB trace with a larger resistor to slow down the clock signal rise time can increase the noise margin of the clock signal.

ADC Sample Variance Error

Another important consideration for simultaneous-sampling systems is the internal sample variance error between ADCs. The variance error is due to process variation that affects propagation delay in the internal path from the clock signal input to the modulator. The sampling instant of the modulator is the key attribute. The sample variance error is minimized using a clock signal with rise time 1 ns or less. The ADC sample variance error between individual ADCs is specified under identical operating conditions: ambient temperature and power supply voltage, using a coincident arriving clock and analog input signals.

Anti-Alias Filter Group Delay

The digital filter of the ADS127L11 is a linear-phase design. Linear phase means all input signal frequency components are shifted the same amount of time. When the filter phase is plotted in time vs. frequency, or plotted in radians vs frequency, the phase is either a horizontal line (time) or a straight line with constant slope (radians). For either plot format, the phase is linear (constant group delay). Since phase is defined by the digital filter, the resulting phase is predictable and invariant.

However, the analog anti-alias filter (AAF) at the ADC inputs can add nonlinear group delay within the signal passband. The amount of non-linearity depends on the filter tuning. For example. a 500 kHz, 4th-order AAF critically damped filter alignment (2-mdB peaking) causes the group delay to change from 0.577 µs at 10 kHz to 0.595 µs at the edge of 165 kHz passband when sampling at 400 kSPS.

Clock Jitter

The ADS127L11 operates 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 is to improve both the rated SNR, stemming from 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's 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 white.

To achieve data sheet performance, be sure the clock signal is low jitter and free from glitches. 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 signal. In general, RC oscillators embedded in controllers do not have the required jitter performance and should be avoided. The amount of clock jitter that is acceptable is proportional to the input signal frequency and is inversely proportional to the root of the user-programmed oversampling ratio (OSR). Equation 1 and Figure 3 show the effect of clock jitter to SNR_j.

$$SNR_j\,(dB) = -20 \cdot \log(2\pi \cdot f_{IN} \cdot t_j / \sqrt{OSR}) \quad (1)$$

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 (see the ADS127L11 data sheet for OSR values)

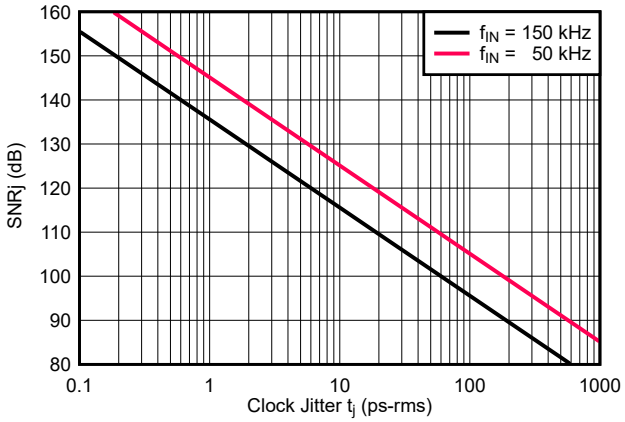

Figure 3. SNR_j vs. Clock Jitter

For example, with an input signal frequency = 150 kHz and OSR = 32 (f_{DATA} = 400 kHz), clock jitter should be less than 10 ps to yield SNRj = 116 dB. Ideally, SNRj should be 6 dB 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, operate SCLK with a phase-coherent frequency to the ADC clock signal. Non-phase coherent SCLK and other clock frequencies present in the system can intermodulate with the ADC clock signal producing unwanted sum and difference frequency multiples in the ADC output spectrum. Correlated types of clock signal jitter can occur elsewhere in the system and can couple to the ADC clock signal through various mechanisms. These include through the system ground, such as nearby operation of a switching power supply, capacitive and mutual inductance coupling between parallel clock traces or by routing unrelated clock signals through the same buffer package as the ADC clock signal. Separate and shorten the SCLK and SDO/$\overline{DRDY}$ traces as much as possible from the clock signal to reduce coupling. In some cases, it may be beneficial when reading ADC data to operate SCLK over the entire conversion period with extra SCLKs to spread SCLK clock signal energy evenly over the conversion period.

Synchronization

Due to process variation of the internal power-up thresholds, the ADCs are initially out of synchronization until they are externally synchronized. Synchronization is also needed after changing the configuration of the device. The ADCs are synchronized by the START pin or by the SPI start bit.

The most straightforward method of synchronization is to use a single control line routed in parallel to the START control inputs of the ADCs. The ADCs are synchronized at the next rising edge of the clock signal after START is asserted high. Assert START high on the falling edge of clock to avoid the uncertainty of the rising edge of clock used to latch START. Otherwise, a one clock timing error between the ADCs could result. Note for proper operation of the synchronized control mode when applying a continuous START signal, START should always be applied on the falling edge of clock. The effects of START signal PCB trace-length mismatch is not as critical as matched clock traces because the ADC re-times the START inputs on the rising edge of clock, provided the mismatch is less than ½ clock period.

Optionally, independent START lines to each ADC can be used to make fine adjustments of channel phasing between the ADCs with up to one clock cycle resolution. In this case, the clock signal skew between the ADCs is not be a concern.

If the sync signal arriving to the system is asynchronous to the system clock and routed directly to the ADCs, then one clock cycle of uncertainty can exist between the ADCs after synchronization. In this case, an external synchronizing circuit to synchronize the sync signal for all ADCs is shown in Figure 4. The synchronizer releases the sync signal on the falling edge of the clock signal, ensuring the ADCs are synchronized to the same clock cycle. The SN74AUP2G79 dual D-type flip-flop can be used for this purpose.

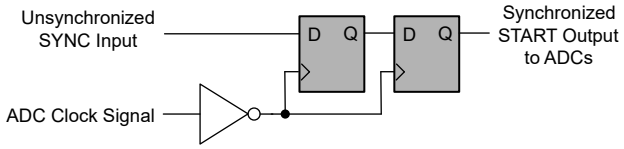

Figure 4. Synchronizing the SYNC Signal

Conversions can also be synchronized by the SPI interface (not for use in the synchronized control mode). To synchronize the ADCs by SPI, shift in the register write command to the CONTROL register in a parallel operation to set the START bit. The daisy-chain connection provides the parallel load functionality because the ADCs are gated and controlled by use of a single chip select signal. 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 clock and $\overline{CS}$ taken high on the falling edge of clock.

SPI Daisy-Chain Connection

One challenge of using multiple ADCs is the added complexity of the SPI signal routing between ADCs and the host controller. The ADS127L11 offers a daisy-chain option to streamline the SPI connections. The daisy-chain option loops the SPI data output of one ADC into the SPI data input of the next ADC in the chain. The daisy-chain connection can be thought of as linking 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 5 illustrates how the individual ADC shift registers emulate a single shift register through the daisy-chain connection.

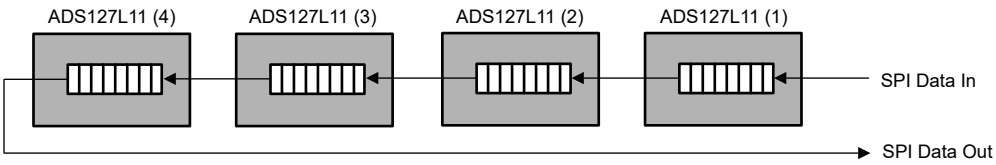

Figure 5. Daisy-Chain Concept

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, if connected in a standard SPI cascade connection using four chip-select lines (one for each ADC), seven SPI lines are needed to interface to four ADCs.

See Figure 6 and Figure 7 for comparison. If using either standard SPI cascade or daisy-chain connection, additional control lines may be necessary for ADC data-ready output ($\overline{DRDY}$), ADC synchronization (START), and ADC reset ($\overline{RESET}$).

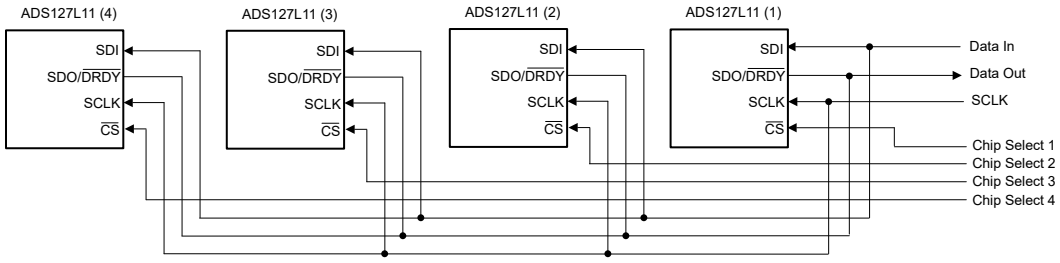

Figure 6. Conventional SPI Cascade Connection

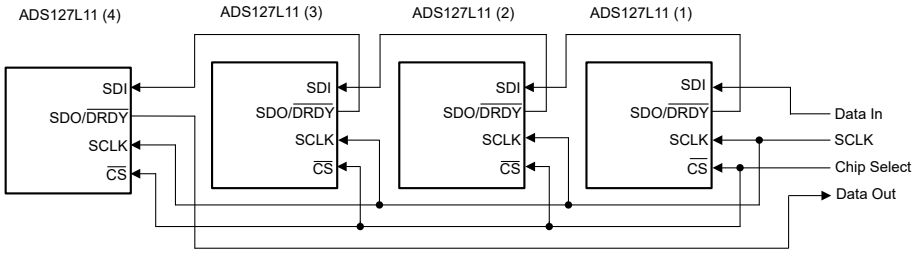

Figure 7. Daisy-Chain SPI Connection

There is no special programming required for daisy-chain operation. The host controller simply extends the data frame to the length needed to access the data from all the ADCs connected in the chain. The new data frame length to set to match the number of devices in a chain x the number of bits per frame of the ADC. For example, with four devices in a chain using 24-bit ADC frame lengths, the new frame length is 96 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 this case). The ADCs interpret the data in their respective shift registers at the time $\overline{CS}$ is taken high. This means there is no limit to the amount of shift operations as the data passes through each ADC, it is only the last bits shifted into each ADC that matters. Figure 8 shows an example of 24-bit input frame lengths per each ADC to match the 24-bit output data size. See the ADS127L11 data sheet for additional details of the daisy-chain input command format.

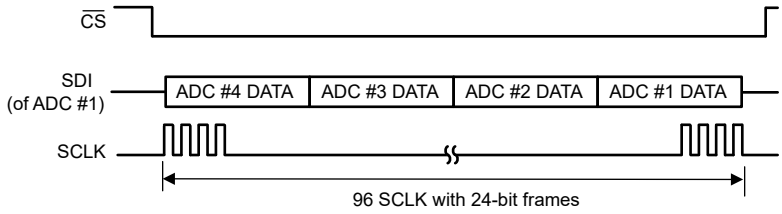

Figure 8. Daisy-Chain Data Input Sequence

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 this case), followed by data from the next device in the chain (ADC #3), and so on (see Figure 9). There is no interruption or gaps in the data stream between the devices. If the ADCs are programed for 24-bit SPI frames, perform 96 shift operations to read data from four devices.

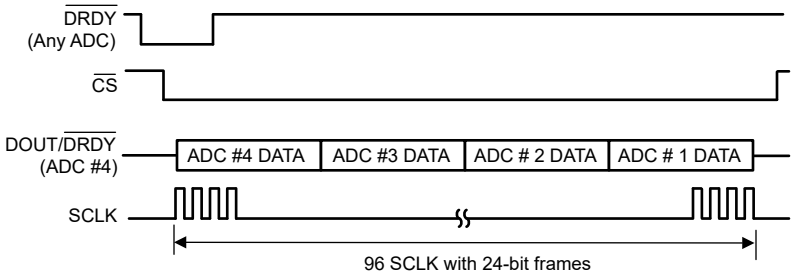

Figure 9. Daisy-Chain Data Output Sequence

When reading register data, two frames are required. The first frame inputs the read register command with the general format shown in Figure 8. The second frame outputs the register data. The first register output data is from device #4 with two pad bytes inserted after the register data byte. The pad bytes fill-out the individual ADC frames to equal the 24-bit conversion data size. Even through the total number of register data bytes is four (one byte from each ADC), 96 shift operations are needed to read the register data from all the ADCs.

System Requirements for Daisy-Chain

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

1. Use the four-wire SPI mode. One $\overline{CS}$ control line can be used to simultaneously select and deselect the ADCs in the chain.
2. The SD0/$\overline{DRDY}$ pin must be remain programmed in the default data-output only mode. The dual function mode of data ready and data output of this pin cannot be used for daisy-chain operation.
3. To reduce the complexity of interfacing to the ADCs, program the ADCs to the same frame length using a parallel write operation (program 16, 24, 32, or 40-bit individual frame length as desired).
4. Use pull-up resistors between the SDI input and SDO/$\overline{DRDY}$ output connection between ADCs. Since $\overline{CS}$ tri-states SDO/$\overline{DRDY}$, the resistor prevents the SDI input from floating in any condition.

Number of Devices in a Daisy-Chain Connection

The maximum number of 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 is also true using the standard SPI cascade connection, because data is also read sequentially in this mode.

The ADS1271L11 supports native SCLK speeds up to 50 MHz. But this requires non-SPI standard same edge clock-out and clock-in which is not supported in daisy-chain connection. When considering that normal SPI operation is opposite edge clock-out and clock-in, the data output pin propagation time (SDO/$\overline{DRDY}$) combined with the data input pin set-up time (SDI), the SCLK speed is limited to approximately 16.5 MHz. Operating IOVDD at 2 V or greater increases the maximum SCLK speed to approximately 20 MHz.

The number of ADCs connected in a single daisy-chain is limited by the SCLK frequency, the data rate and the bits per frame of each ADC, as expressed in Equation 2.

$$\text{Number devices in a daisy-chain connection} = \lfloor f_{SCLK} / (f_{DATA} \cdot \text{bits per frame}) \rfloor \quad (2)$$

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

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

Parallel SDO/DRDY Connection

In systems where multiple ADCs are sampling at high data rates, such as 1 MSPS, the conventional SPI cascade connection and the daisy-chain connection cannot provide the necessary data throughput. In this case, connect the SDO/$\overline{\text{DRDY}}$ pins in parallel to the host controller and operate SCLK in parallel at a minimum of 17 MHz to read 16-bit data, 25 MHz to read 24-bit data, and so on. Note for 25 MHz SCLK operation, observe the SCLK to SDO/$\overline{\text{DRDY}}$ propagation time specification. It may be necessary to use same edge clock-out/clock-in technique and apply one additional SCLK because of the bit offset.

Figure 10 shows parallel SDO/$\overline{\text{DRDY}}$ connections leading to the host. In this example, single $\overline{\text{CS}}$, SCLK, and SDI control lines are shown, therefore requiring the ADCs to have the same device configuration. If the ADCs are to have unique configurations, then use separate $\overline{\text{CS}}$ control lines to enable the ADCs for independent input communication.

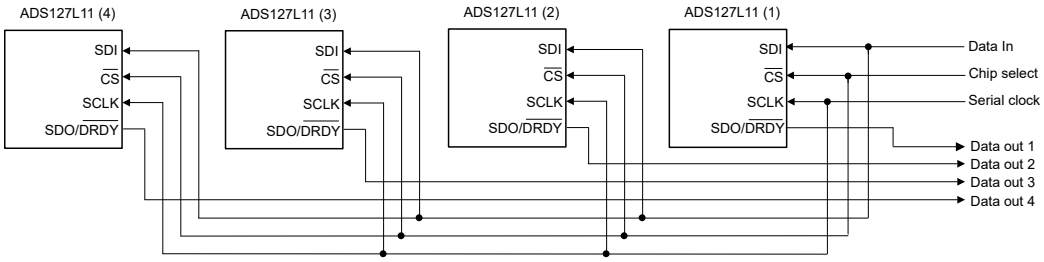

Figure 10. Parallel SDO/$\overline{\text{DRDY}}$ Connection

Data-Ready

Conversion data-ready of the ADC can be determined by several methods.

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

Note 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 may choose to monitor all the $\overline{\text{DRDY}}$ outputs to verify the ADCs are synchronized; or for applications where an intentional phase offset is used between ADCs to verify phasing between devices. For these circumstances, monitor the $\overline{\text{DRDY}}$ outputs from each ADC.

Alternatively, clock cycles can be counted after synchronizing to predict when to read the conversion data. At synchronization, the digital filter restarts, therefore additional time is required for filter settling to produce the first conversion result. The net time delay for the first conversion result is specified as latency time, listed in the ADS127L11 data sheet.

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 (that is, the old data is not repeated).

Reference Voltage

Multiple ADCs can share a single voltage reference with good results. In addition to the large bulk decoupling capacitor at the voltage reference output, use local decoupling capacitors at the ADC REFP and REFN pins. It is recommended to enable the internal REFP buffer to reduce loading of the voltage reference.

Route REFP and REFN as differential pair with the same care as routing the signal. Route a wide PCB trace starting at the voltage reference ground pin to the REFN pins using no intervening ground ties. Use of ground ties to the REFN trace may result in ground noise pick-up in the reference signal. See Figure 11 for an example REFP and REFN routing and reference input pin bypassing at the ADCs.

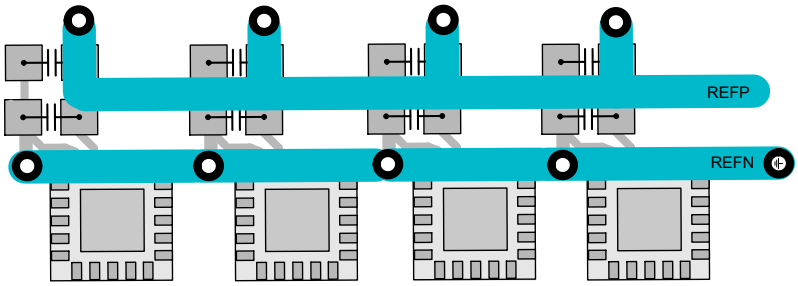

Figure 11. Reference Voltage Routing

Power Supply Bypassing and Grounding

For best performance, it is recommended to follow the same practices of power supply bypassing and ground plane used for a single device for a multiple ADC system. See the ADS127L11 data sheet for power supply bypassing recommendations. A dedicated, unbroken ground plane layer with shared analog and digital grounds for the ADCs in multichannel systems generally provides the best results.

Summary

The clock signal, synchronization and voltage reference are key signals for any ADC. These signals are just as important in a simultaneous-sampling system. The clock signal should be crystal-based with low jitter and routed to the ADCs with care so to not add clock jitter noise and to keep isolated from other clock signals. It is important to synchronize the ADCs after power-on and after the ADC configuration is changed by asserting the START pin high only on the falling edge of clock signal. The reference voltage ground should be tied at a single point at the reference voltage ground terminal. Daisy-chaining the ADCs can be effective in simplifying the number SPI I/O connections between ADCs and the host controller. Parallel SDO/$\overline{\text{DRDY}}$ connections increases data throughput by providing the ability to clock output data in parallel.

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, 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 will 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 (https:www.ti.com/legal/termsofsale.html) 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.

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265