Understanding Serial LVDS Capture in High-Speed ADCs

Application Report

Literature Number: SBAA205
July 2013

Contents

List of Figures

Abstract

Vinod Paliakara, Shantanu Prabhudesai
High-Performance Analog Products

This application note describes various schemes of interfacing serialized low-voltage differential signaling (LVDS) data outputs from high-speed analog-to-digital converters (ADCs) to a field-programmable gate arrays (FPGAs) or other application-specific integrated circuit (ASIC)-based receivers. This note provides an introduction to standard one-wire interfaces and other interface variants (such as two-wire). This note describes in detail the two key points required for reliable data capture: bit clock edge selection and frame alignment. Schemes for capturing data from multiple ADC devices are also detailed. LVDS timing parameters, as found in TI data sheets, are explained along with a detailed introduction to jitter in a source-synchronous interface.

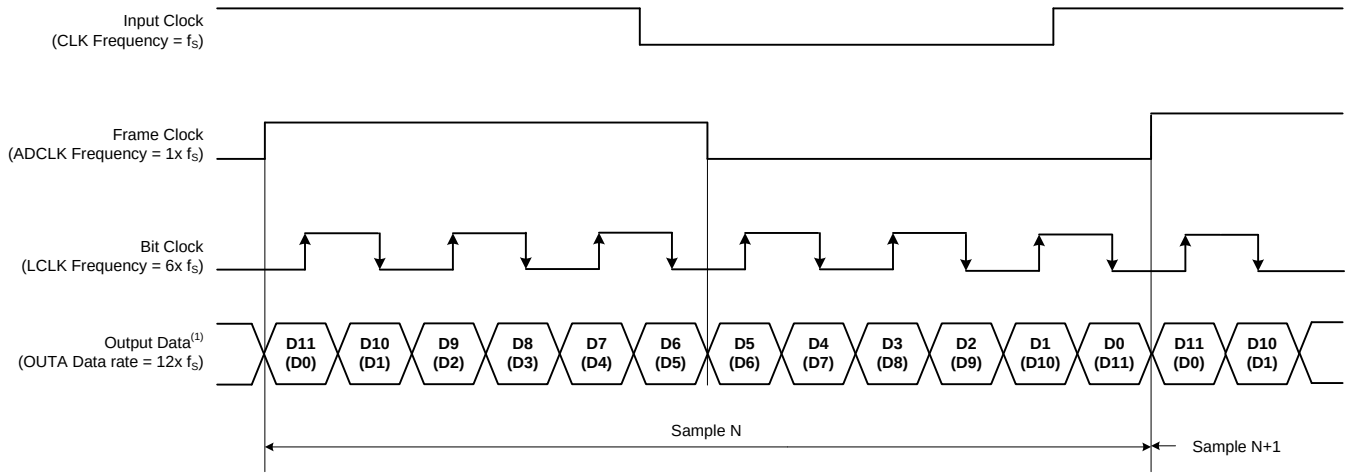
Introduction

Among high-speed ADCs operating at tens of megahertz to hundreds of megahertz, the preferred output data interface is serialized LVDS. This preference is especially true in multichannel ADCs such as TI's ADS528x, ADS529x, and ADS5263 families. The serialized LVDS interface is also seen in analog front-ends (AFEs) that include a multichannel ADC, such as the AFE580x families. The differential signaling protocol of an LVDS pair offers common-mode noise rejection, thus enabling higher data transmission speeds (typically up to 1 Gbps per LVDS pair).

Figure 1-1 shows a typical serialized LVDS interface for one ADC channel operating on a sampling clock frequency (f_S). The characteristics of this interface are:

1. ADC data are output serially from D0 to DN-1 (for an N-bit resolution ADC) at every sampling clock cycle. This output results in a serial data rate of ($f_S \times N$) bits per second.
2. An associated 50% duty cycle bit clock is output with a frequency of ($f_S \times N / 2$). The bit clock is typically center-aligned and both clock edges can be used to latch serial ADC data. Therefore, the bit clock is referred to as a double data rate (DDR) bit clock.
3. An associated 50% duty cycle frame clock is output with a frequency of f_S. As the name suggests, the frame clock rising edge transitions are aligned with the framing ADC data bits (D0 and DN-1). This alignment helps the receiver to correctly load parallel data after de-serialization. Note that the frame clock rising and falling edges are aligned with the transitions of data.

This interface is described as a one-wire, LSB-first interface with N-x serialization.



(1) The upper data bit is the MSB-first mode data bit and the lower data bit is the LSB-first mode data bit.

Figure 1-1. LVDS Output Timing Diagram

Some variations of this interface include:

1. **Two-wire interface:** At high sampling clock frequencies, the serial data rate becomes very high. In such cases, the ADC data are output serially over two LVDS pairs. For example, bits D0 to D(N/2)-1 are output over one LVDS pair and the other bits D(N/2) to DN-1 over the second pair. With half the number of bits transmitted per LVDS pair on each sampling clock cycle, the serial data rate is reduced to ($f_S \times N / 2$) bits per second.

2. **MSB-first:** the ADC data bits are serially transmitted from the MSB bit (DN-1) to the LSB bit, D0.

An advantage of the serial LVDS interface is the savings in pin-count resulting from the use of only two pins (one LVDS pair) per ADC channel. Therefore, this interface is most commonly found in multichannel ADC devices. A typical multichannel device has one (or two) LVDS pairs per ADC channel, one common bit clock output, and one frame clock output. Refer to Figure 1-2 for the LVDS output interface of an 8-channel ADC device.

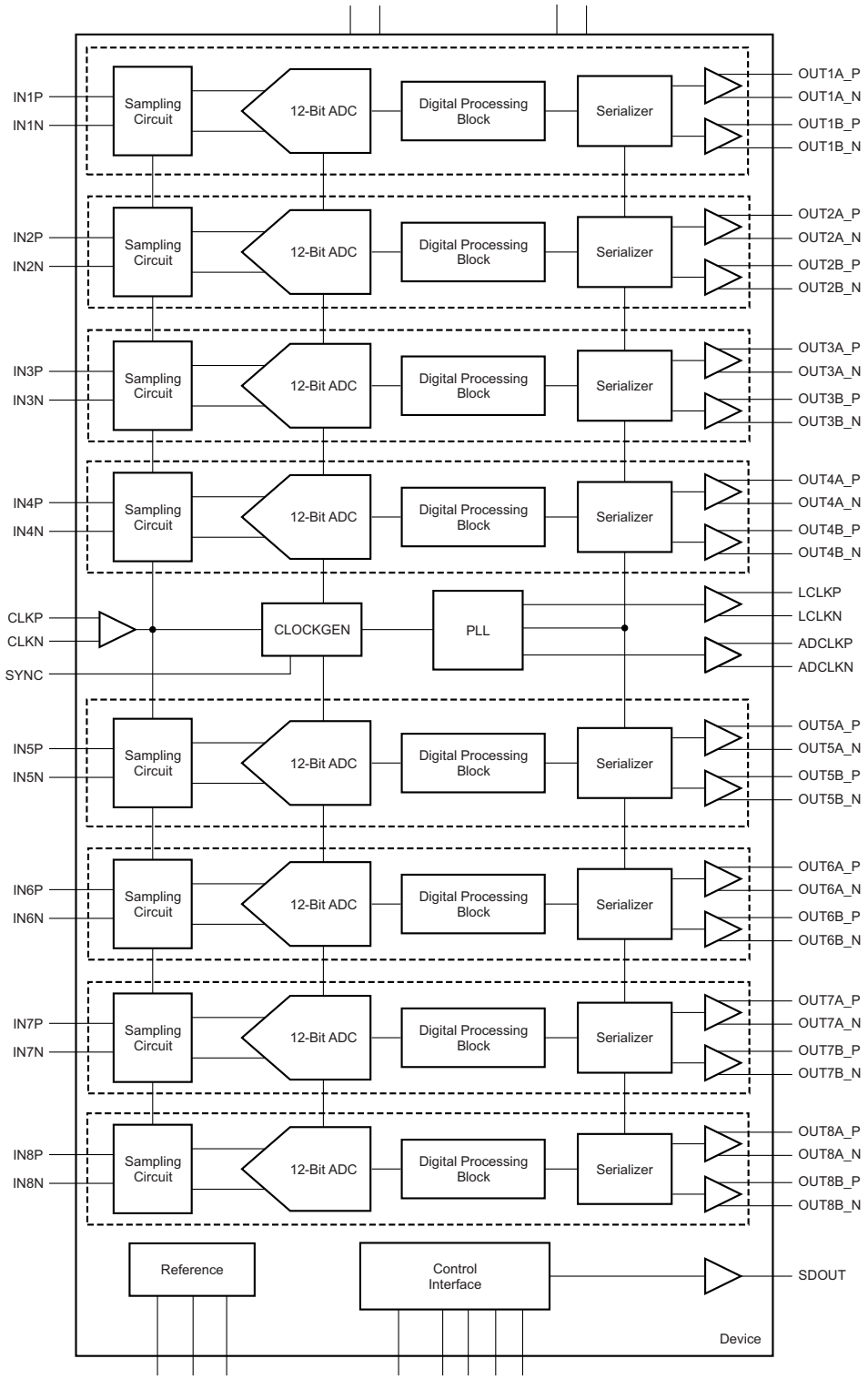


Figure 1-2. Typical Multichannel ADC with a Serial LVDS Interface

Capturing Data from a Serial LVDS Interface: Ideal Case

The simplest capture scheme to receive data from a serial interface consists of a double data rate (DDR) logic block followed by a serial-to-parallel shift register, as shown in Figure 2-1. The length of the shift register chain must be at least (N / 2) flip-flops. At every frame clock rising edge, the parallel data output from both shift register chains are latched. The output is a frame-aligned parallel data word of N-bits (D0 to DN-1).

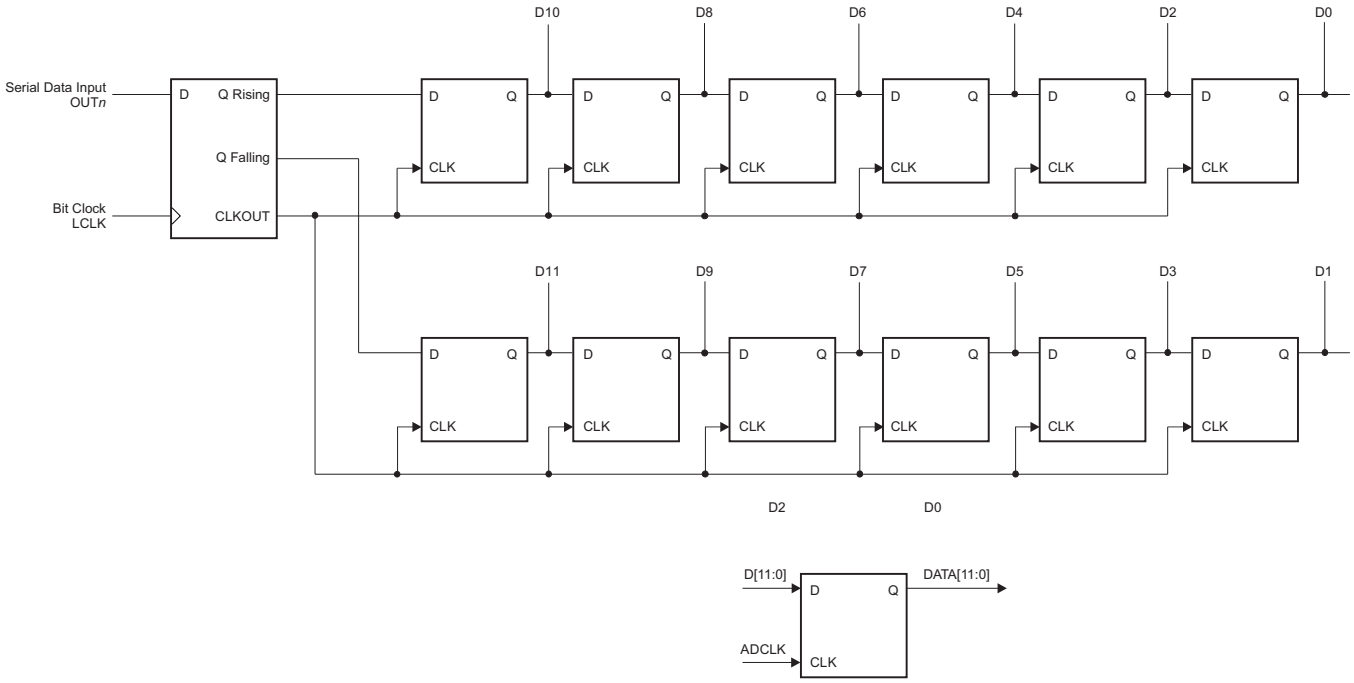


Figure 2-1. Capturing Serial LVDS Data: Simplest Scheme

The DDR block latches input serial data at both edges of the bit clock and outputs two data streams: one corresponding to the rising edge of the bit clock (Q rise) and the other corresponding to the falling edge of the bit clock (Q fall, as shown in Figure 2-2). The rising and falling data streams are then applied to the shift register.

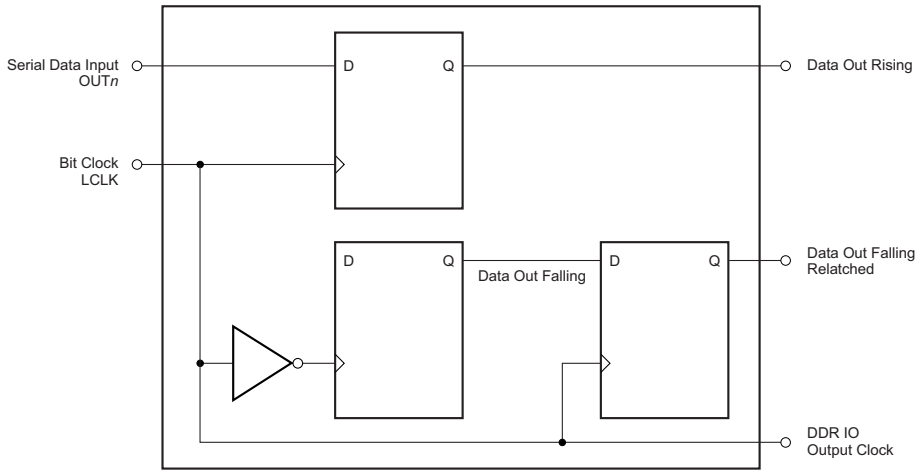


Figure 2-2. Double Data Rate IO

Note that the DDR block re-latches the data stream falling edge at the next bit clock rising edge and outputs the rising and falling edge data streams at the rising bit clock edge, which is used by all the flip-flops in the shift register chain. As a result, data are valid for roughly half a bit clock period at the DDR IO input, whereas data are valid for almost one bit clock period for the remaining flops in the shift register chain. Figure 2-3 shows a DDR IO timing diagram.

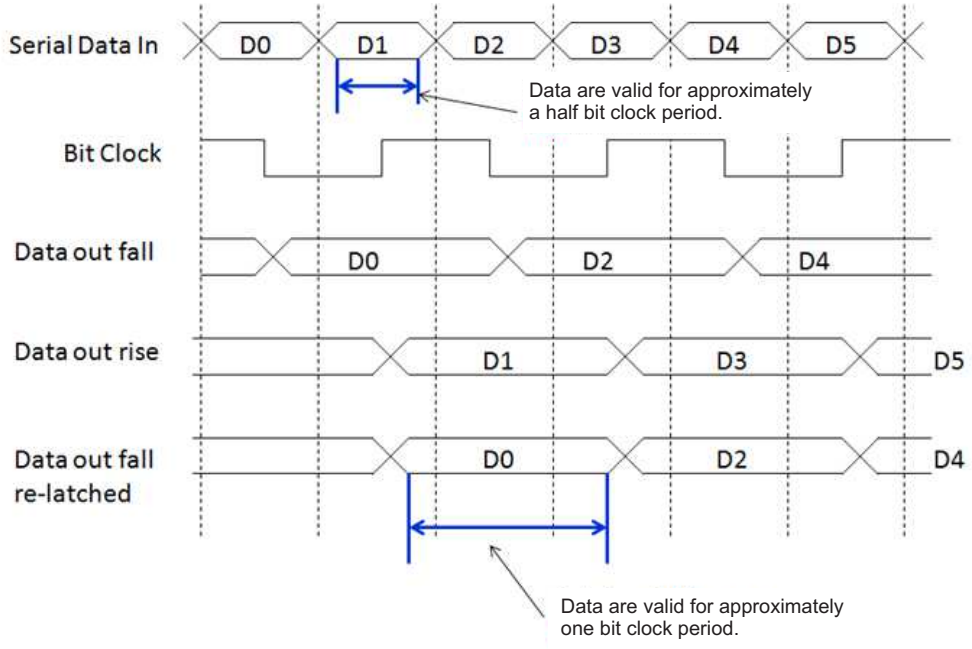


Figure 2-3. DDR IO Timing

This difference implies that the timing constraint for the flip-flops in the DDR block alone is critical compared to the rest of the flip-flops.

Most FPGAs have a DDR flip-flop and register as part of the logic library. The DDR IO element accepts a single clock and registers data at the input data pin at both clock edges. The DDR IO element is also physically located close to the FPGA receiver pins, thus helping minimize any delays caused by routing in the FPGA. Therefore, TI recommends using the DDR IO element to interface to the ADC serial LVDS interface.

Ideally, there is no additional skew between the serial LVDS data, LVDS bit clock, and LVDS frame clock signals caused by delays in the PCB and FPGA internal routing. In a real situation, the effects of trace delays in the PCB and FPGA routing cannot be ignored.

Chapter 3 and Chapter 4 describe various implementations for capturing data from a serialized ADC interface in a real-world situation.

Receiver Capture Schemes

Any capture solution must implement two major functions:

1. Latching the serialized ADC data bits correctly into the serial-to-parallel (S2P) shift register using the ADC bit clock (serialized clock domain), and
2. Aligning the parallel output data from the S2P registers correctly (parallel clock domain).

Various methods of implementing these functions are discussed in the *Latching Serialized ADS Data Bits into the S2P Shift Register*, *Aligning Parallel Output Data from the S2P Shift Register*, *Determining if Captured Data are Aligned Correctly*, and *Frame Alignment Logic Using SYNC Pattern* sections.

3.1 Latching Serialized ADC Data Bits into the S2P Shift Register

As explained previously, a real-world application has delays resulting from PCB traces and FPGA routing. These delays must be taken into account during timing analysis. In many cases (especially for data rates > 600 Mbps), meeting the timing requirements with these effects can be difficult. For such cases, one common solution is using a delayed version of the ADC bit clock for latching data into the shift register. The delay can be accomplished by:

- Using delay elements (external to the ADC and receiver OR built into the receiver itself), and by
- Using phase-locked loops (PLLs)

3.1.1 Using Delay Elements

An important consideration when using this scheme is the delay variation across process, supply voltage, and temperature. Also, ensure that timing is met at both minimum and maximum delay values. With these considerations in mind, using circuits such as delay-locked loops (DLLs) is preferable to generate delays that do not vary with process, supply voltage, and temperature. Many FPGAs have built-in DLL elements that can be used to provide delays within certain ranges (up to a few hundred picoseconds). Figure 3-1 shows a diagram of variable delays used in an FPGA.

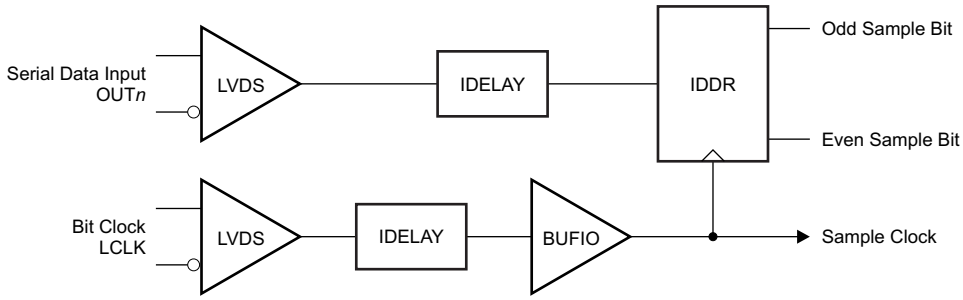


Figure 3-1. Using Variable Delays in an FPGA

3.1.2 Using PLLs

An alternate scheme for deriving a delayed bit clock makes use of a PLL with multiple delayed edges in each clock cycle. Figure 3-2 shows a case where eight equally-spaced clock edges are available within each bit clock cycle. In the receiver, additional logic is required for selecting the correct clock edge. The edge selection logic must run when the ADC interface initializes after power-up and when the bit clock output is stable. Figure 3-3 shows the timing of multiple phase clocks at the PLL output.

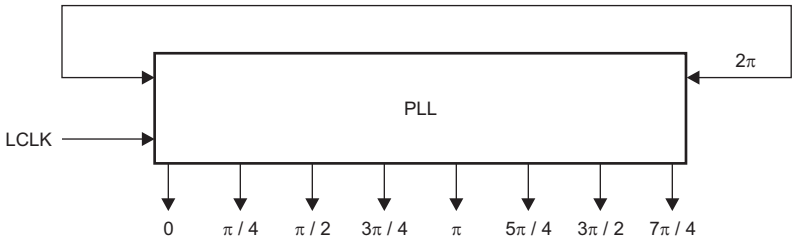


Figure 3-2. PLLs in an FPGA with Multi-Phase Outputs

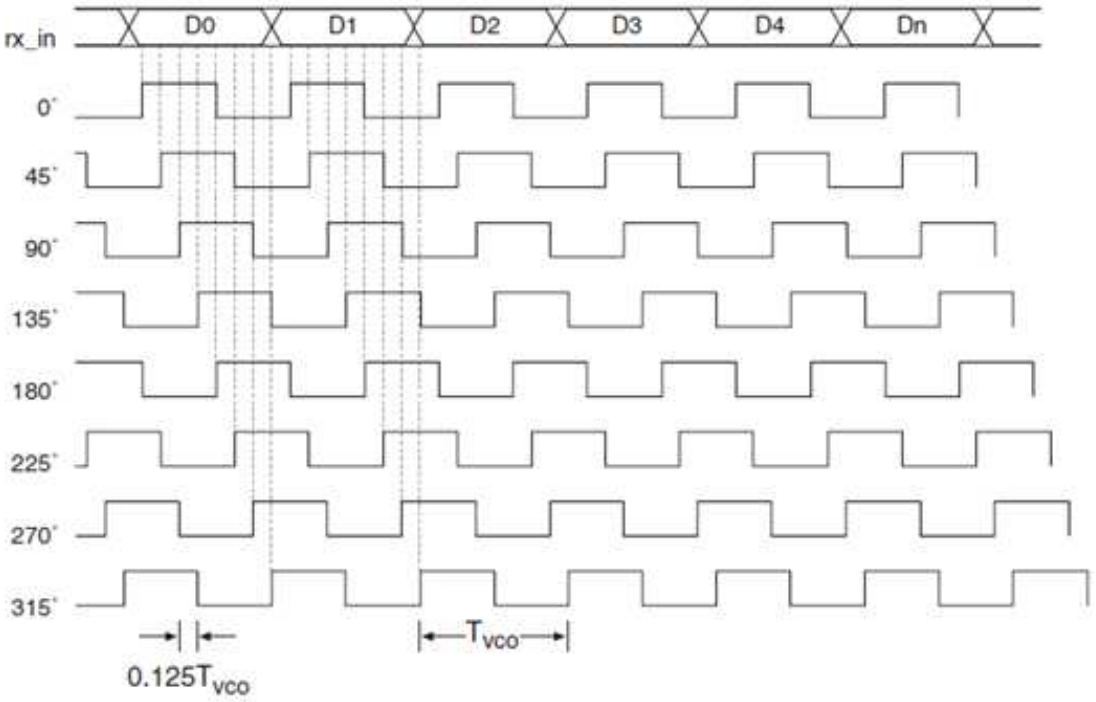


Figure 3-3. Multiple Phase Clocks at the PLL Output

3.1.3 PLL Clock Edge Selection Logic

A typical implementation of the edge selection logic functions by capturing the ADC data for all PLL output clock edges. For data captured at each edge, the logic determines if data are correct or incorrect. If data are correct, the corresponding clock edge can be used for latching serial data.

3.1.4 Determination of Correct Data

How are captured data determined to indeed be correct? All TI high-speed ADCs have a set of test patterns that can be enabled to implement receiver-capture logic. For clock edge selection, a suitable test pattern is termed *deskew*. When enabled, the ADC serial LVDS interface outputs a data bit sequence of alternating '1's and '0's. When captured with the correct clock edge, the expected data are either *101010101010* or *010101010101* for a 12-bit LVDS interface. Figure 3-4 shows an example of captured data for various clock edges. Note how the captured data are correct for a range of clock edges from 3 to 6. The robustness of the scheme can be improved by selecting the clock edge at the center of the good clock edge range, in this case that would be clock edges 4 or 5. This selection provides a margin for any timing changes caused by temperature and supply voltage variations.

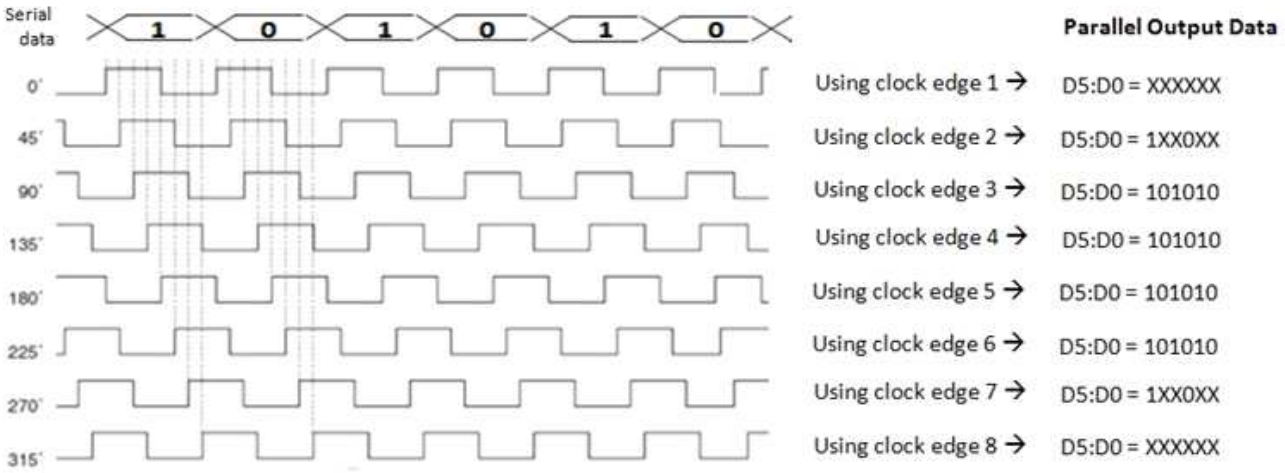


Figure 3-4. Data Captured with Various Clock Edges

Note that whenever the ADC input clock is interrupted, the PLL in the receiver is no longer locked. When the ADC input clock is restarted, the receiver PLL locks again and the output edges stabilize after the PLL lock time. Most PLLs have a reset signal that ensures that clock edge 0 has a fixed relationship with the input clock (either aligned, 90° phase shifted, or so forth). Executing the clock selection logic in the receiver is not required every time the PLL reacquires lock. The same clock edge identified earlier by the logic (clock edge 5 from the example in Figure 3-4) can be reused. TI recommends executing the edge selection logic whenever there is a significant change in temperature, supply voltage, or bit clock frequency.

Other schemes for determining the correct clock edge from captured data also exist. Many FPGAs include a dynamic clock edge selection scheme that runs continuously in the background and does not require any test patterns from the ADC. By running in the background, such a scheme can make dynamic clock edge changes based on temperature and supply variations.

3.2 Aligning Parallel Output Data from the S2P Shift Register

After the serialized ADC data bits are latched correctly by the shift register using one of the previous schemes, the next point to consider is the alignment of the parallel data output of the shift register. As explained in the *Latching Serialized ADS Data Bits into the S2P Shift Register* section, ideally the ADC frame clock output can be directly used to latch the parallel data output of the shift registers. This data output is at the frame clock rising edge because the frame clock rising edge is aligned with the start of the parallel word boundary.

In an actual case, the frame clock can be delayed (or advanced) with respect to the serialized output data outputs. For example, this delay can occur as a result from skews in the PCB traces of the frame and output data lines. However, such skews are reasonably easy to address and minimize during PCB layout. Skews are more likely to occur inside receivers. This occurrence is especially true when using FPGAs as receivers and also when the serial data outputs from multiple ADC devices are routed to one FPGA receiver.

The effect of a skewed frame clock on the final captured data using the ideal receiver scheme is described in Figure 3-5 and Figure 3-6. Note that for skews less than (1/2) a bit clock period, the captured data are correct and include all bits from the same sample (see Figure 3-7). For larger skews, the captured data consist of a mix of bits from two successive samples.

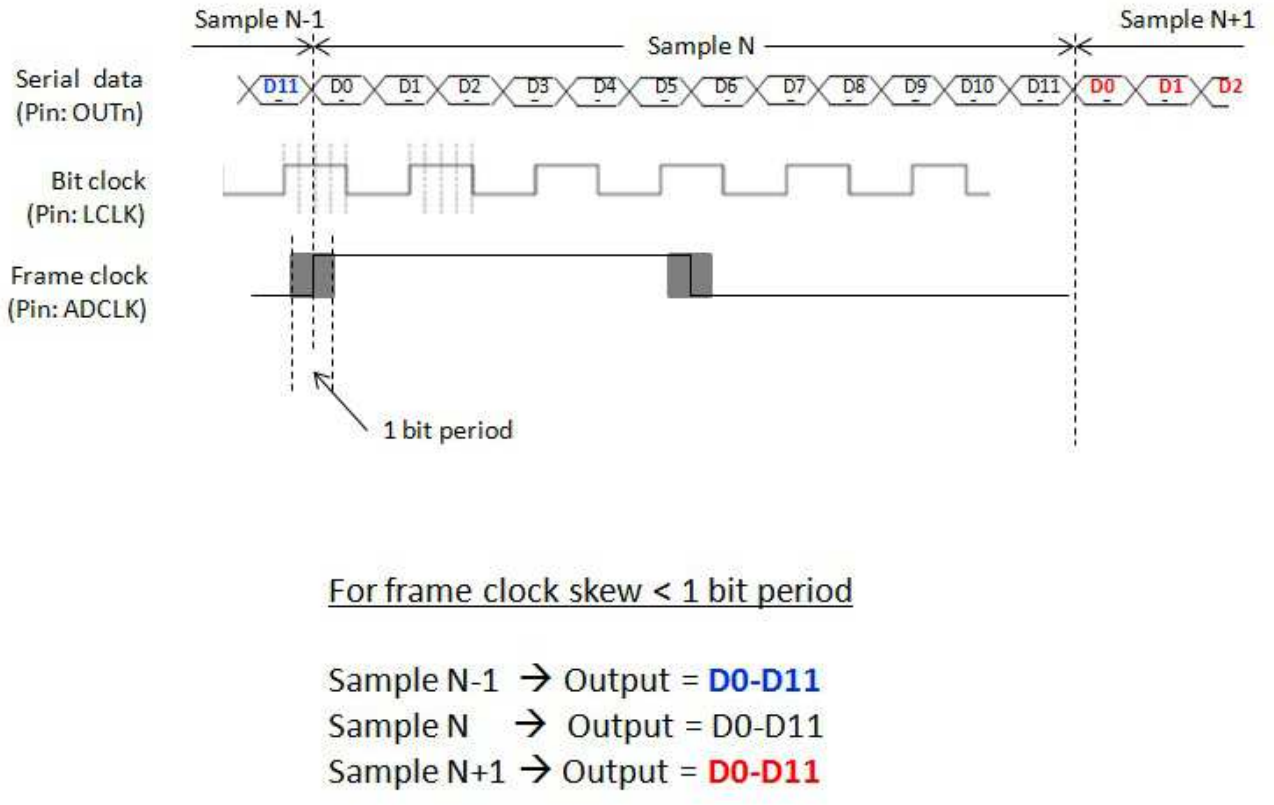


Figure 3-5. Data Alignment with Frame Clock: Ideal Case

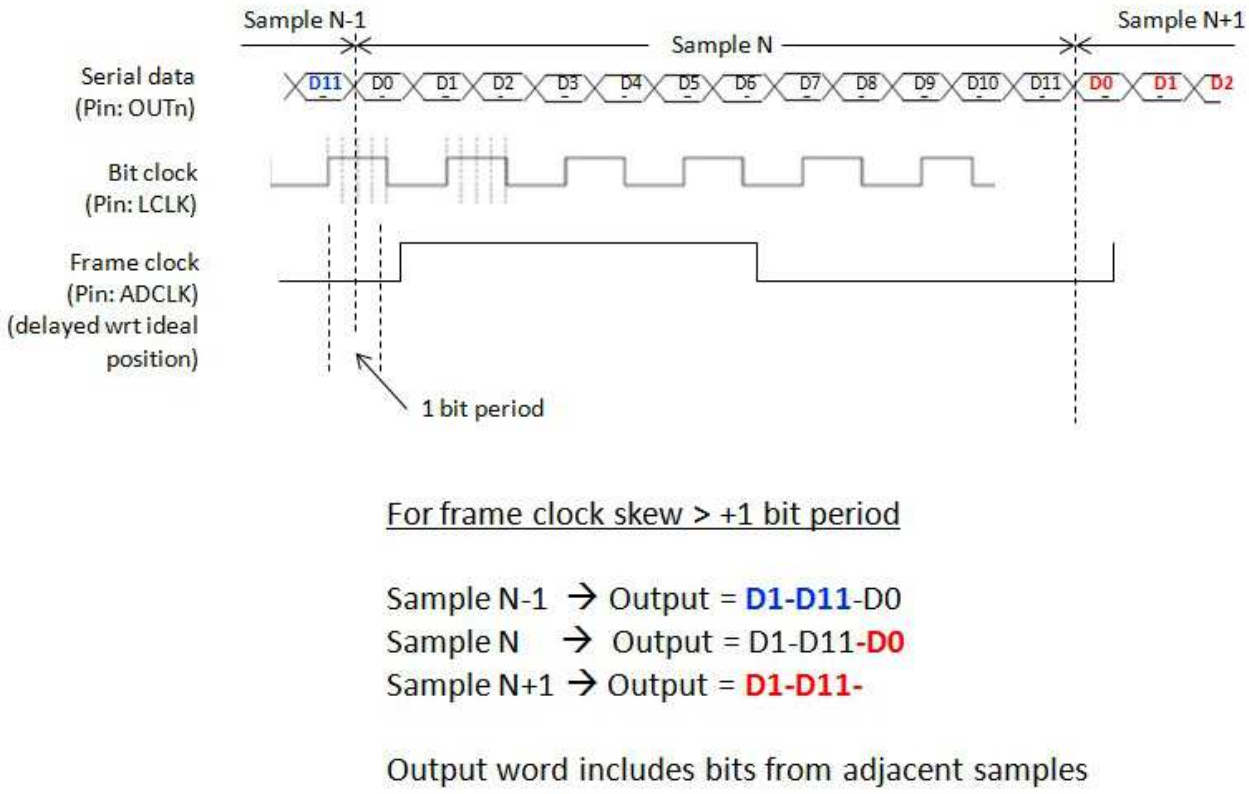


Figure 3-6. Data Misalignment Resulting from a Delayed Frame Clock

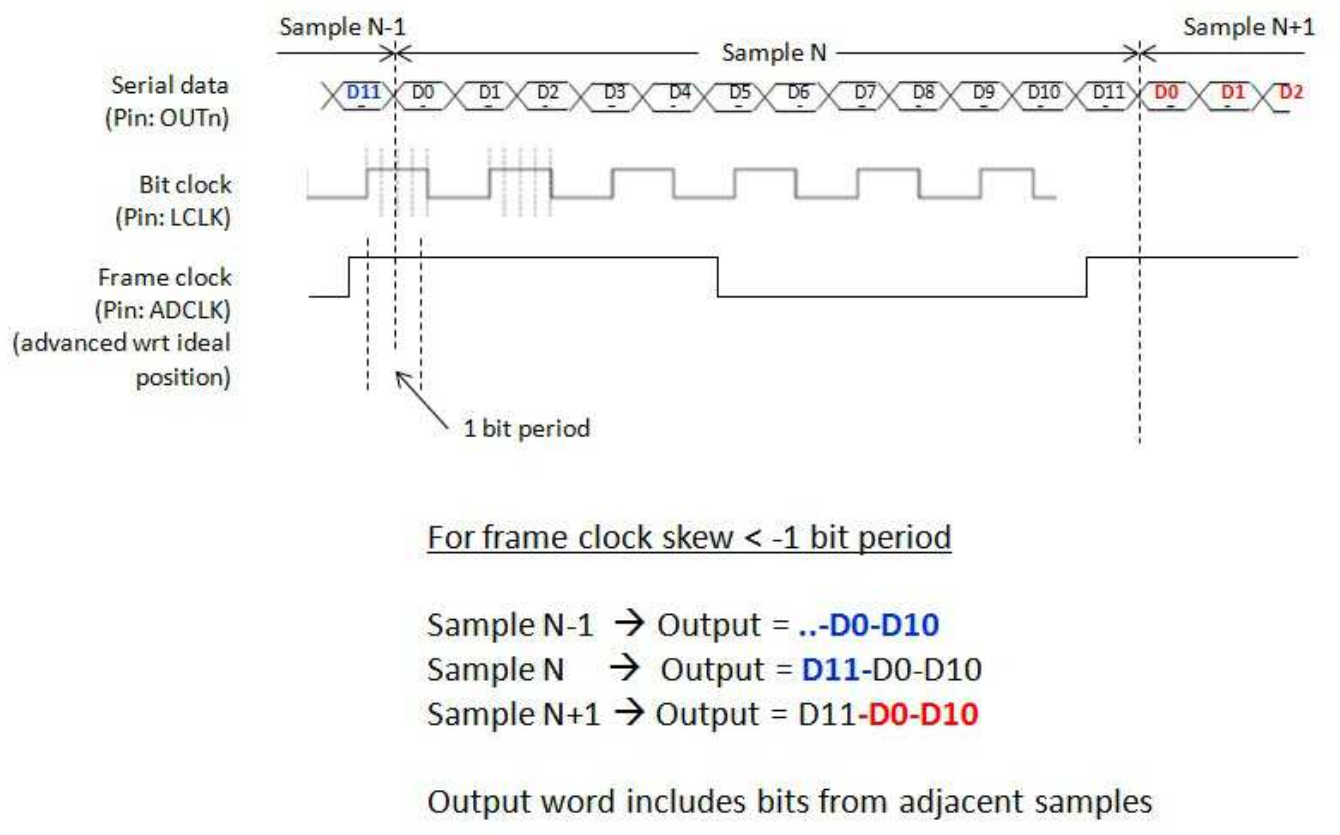


Figure 3-7. Data Misalignment Resulting from an Advanced Frame Clock

3.2.1 Determining if Captured Data are Aligned Correctly

Certain test patterns can be advantageously used to determine if captured data are aligned properly. TI ADCs have a SYNC pattern that consists of a sequence of six '1's followed by six '0's (for a 12-bit serialized ADC). When enabled, each serial data output consists of this pattern. Note that when displayed on an oscilloscope, this pattern resembles a frame clock. Figure 3-8 and Figure 3-9 show the effect of a skewed frame clock when the SYNC test pattern is enabled.

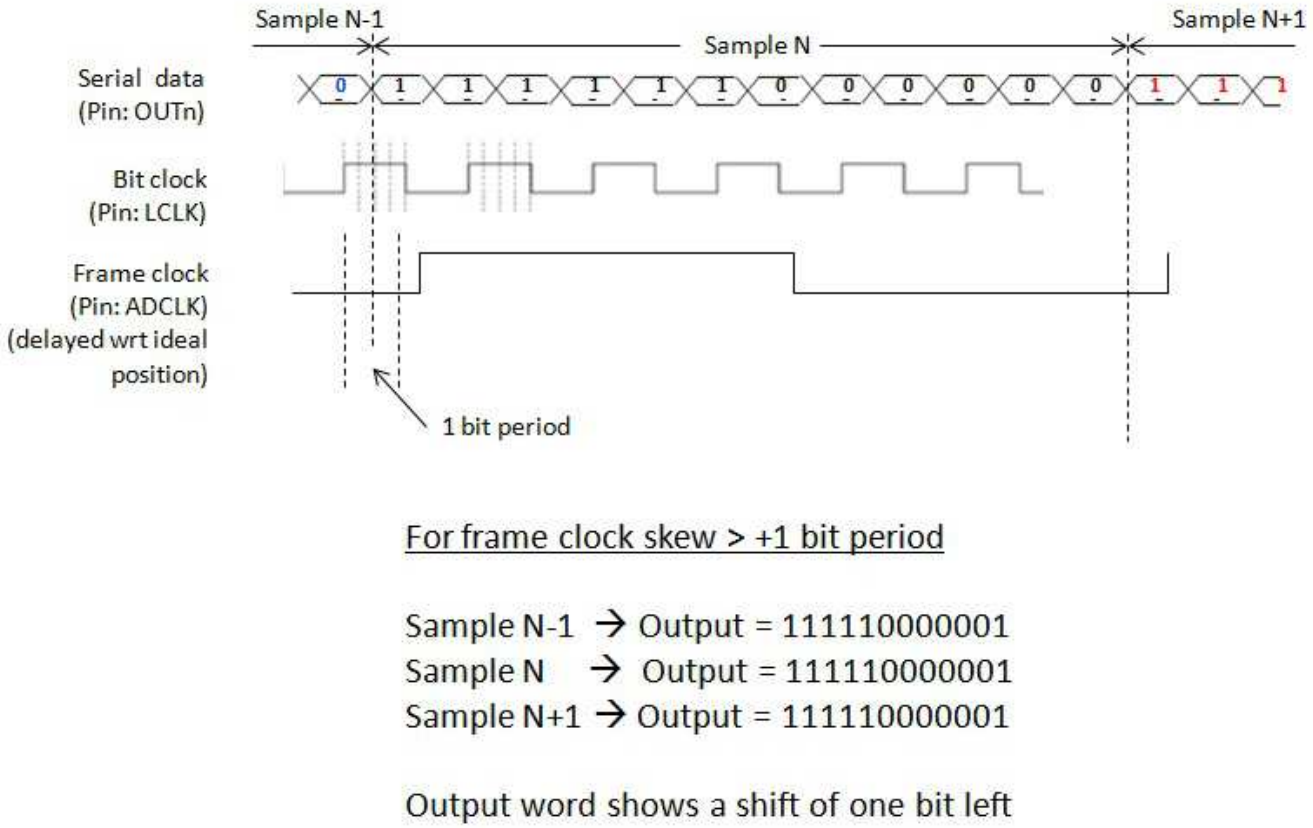


Figure 3-8. Using the SYNC Test Pattern to Determine Data Misalignment: Delayed Frame Clock

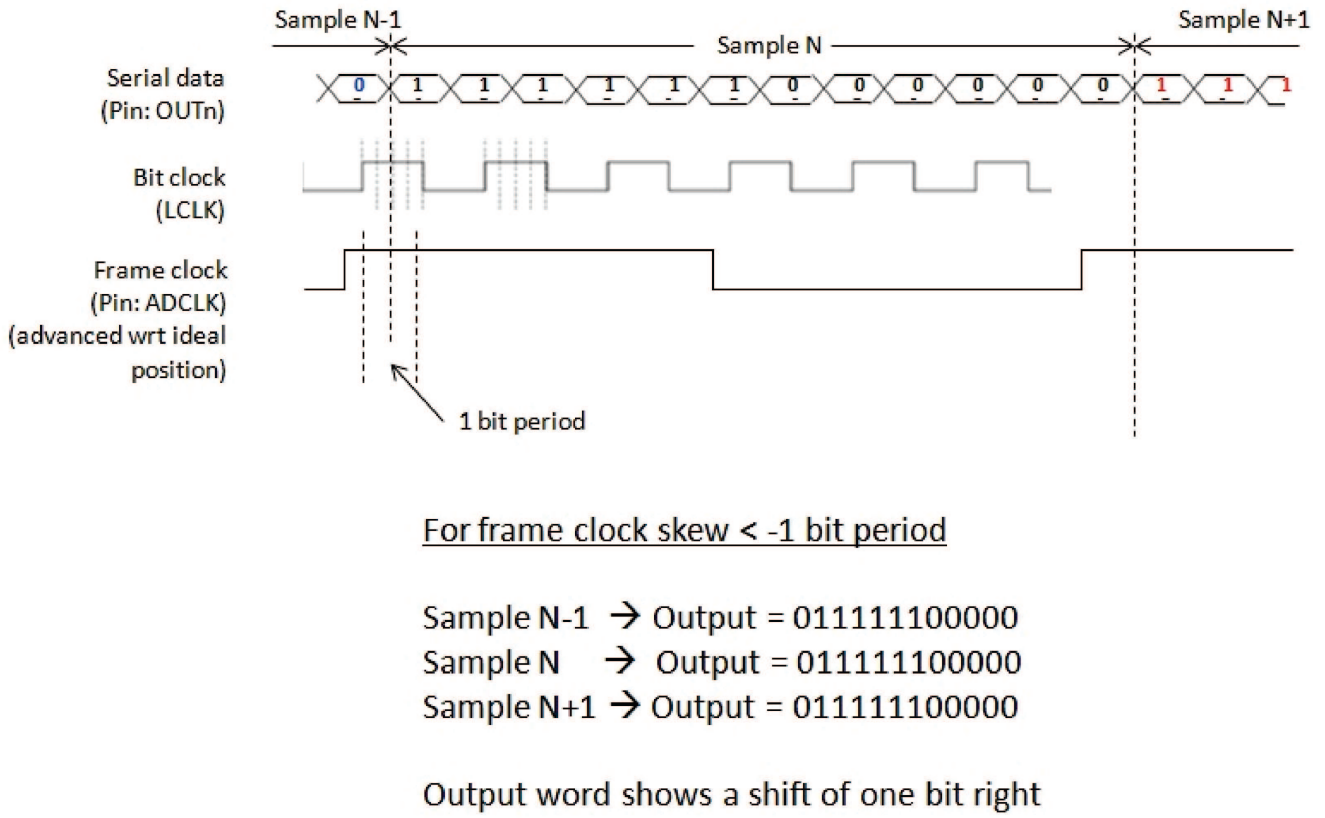


Figure 3-9. Using the SYNC Test Pattern to Determine Data Misalignment: Advanced Frame Clock

Using the SYNC pattern, the *Frame Alignment Logic Using SYNC Pattern* section describes a scheme to correctly align the parallel output of the S2P shift register.

3.2.2 Frame Alignment Logic Using SYNC Pattern

A common logic for aligning to the correct frame boundary is described in this section. Here, the parallel output of the S2P shift registers is first latched using the (skewed) frame clock. This note refers to the latched data as *S2P_PAR_OUT*. The alignment logic requires the SYNC test pattern in the ADC to be enabled. With the SYNC test pattern enabled in the ADC, the S2P_PAR_OUT data are compared with the ideal (or expected) data, which is *111111000000* for a 12-bit serialized ADC. To correct for the effect of the frame clock skew, two successive samples of S2P_PAR_OUT are required. The correction and determination of the final aligned output (termed FINAL_OUT) is described by Figure 3-10.

To summarize Figure 3-10:

- For no skew, FINAL_OUT[11:0] = S2P_PAR_OUT[23:12];
- For one bit clock period positive skew (shown in Figure 3-8),
  - FINAL_OUT[11:0] = S2P_PAR_OUT[22:11]; and
- For two bits clock period positive skew,
  - FINAL_OUT[11:0] = S2P_PAR_OUT[21:10], and so on.
- Similarly, for one bit clock period negative skew (shown in Figure 3-9),
  - FINAL_OUT[11:0] = S2P_PAR_OUT[12:1]: and
- For two bits clock period negative skew,
  - FINAL_OUT[11:0] = S2P_PAR_OUT[13:2], and so on.

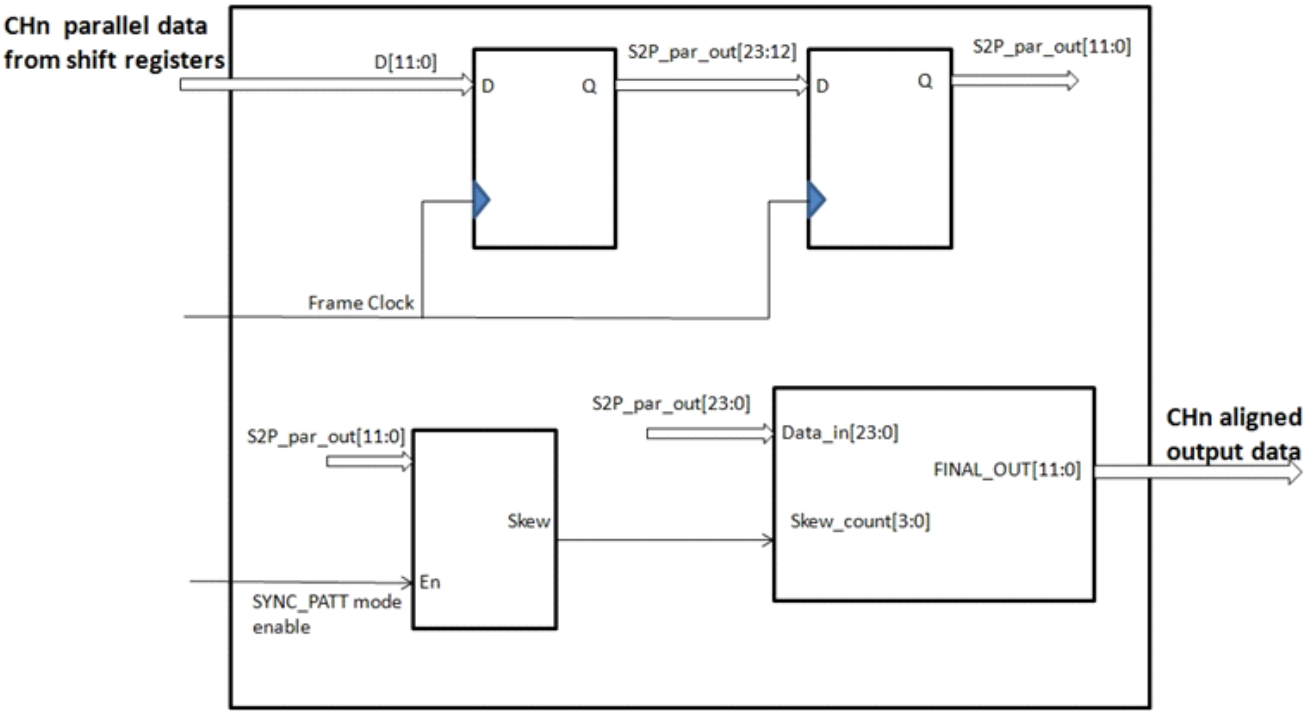


Figure 3-10. Overall Frame Alignment Scheme

Note that the skew between the frame clock and serial ADC data output can be different across channels. The logic shown in Figure 3-10 determines the skew for every channel and corrects for it.

Summary of Capture Schemes

Figure 4-1 and Figure 4-2 show block diagrams of the two capture schemes described in the *Receiver Capture Schemes* section. The block diagram is shown for one channel and is applicable for the one-wire interface.

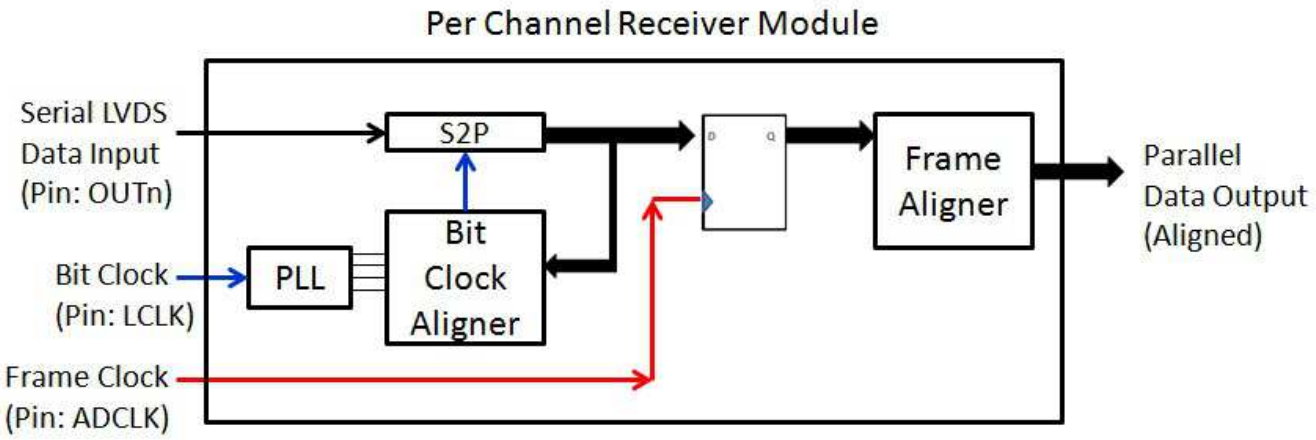


Figure 4-1. Capture Scheme Using PLL: One-Wire Interface

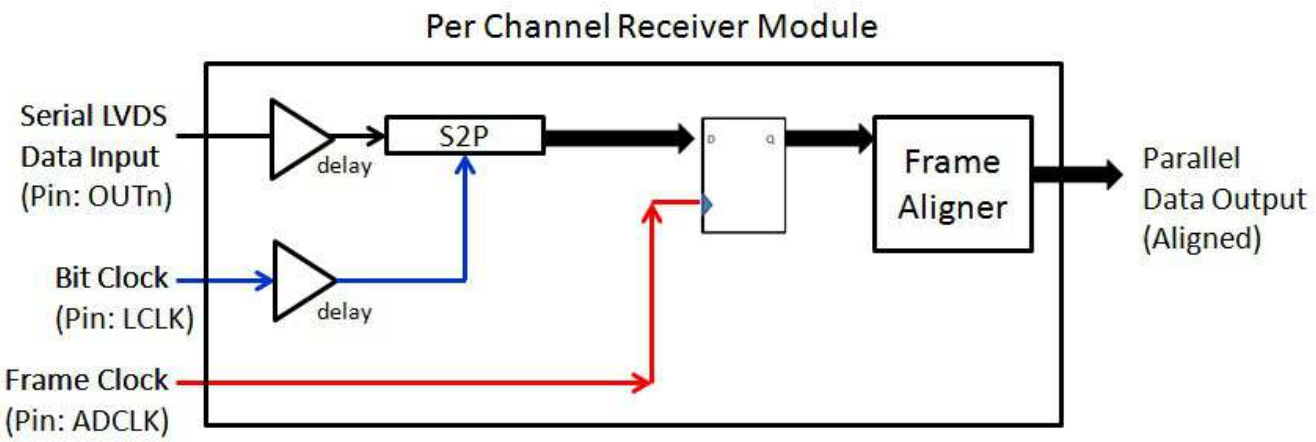


Figure 4-2. Capture Scheme Using Delays: One-Wire Interface

Figure 4-3 shows the block diagram for a two wire interface. Here, the capture scheme for each wire is treated identical to that of a one wire interface. After the data from each wire is frame aligned, the data from both wires are combined and output. Note that while a common PLL can be used for both wires, each wire includes an independent bit align block to account for different skews in each wire.

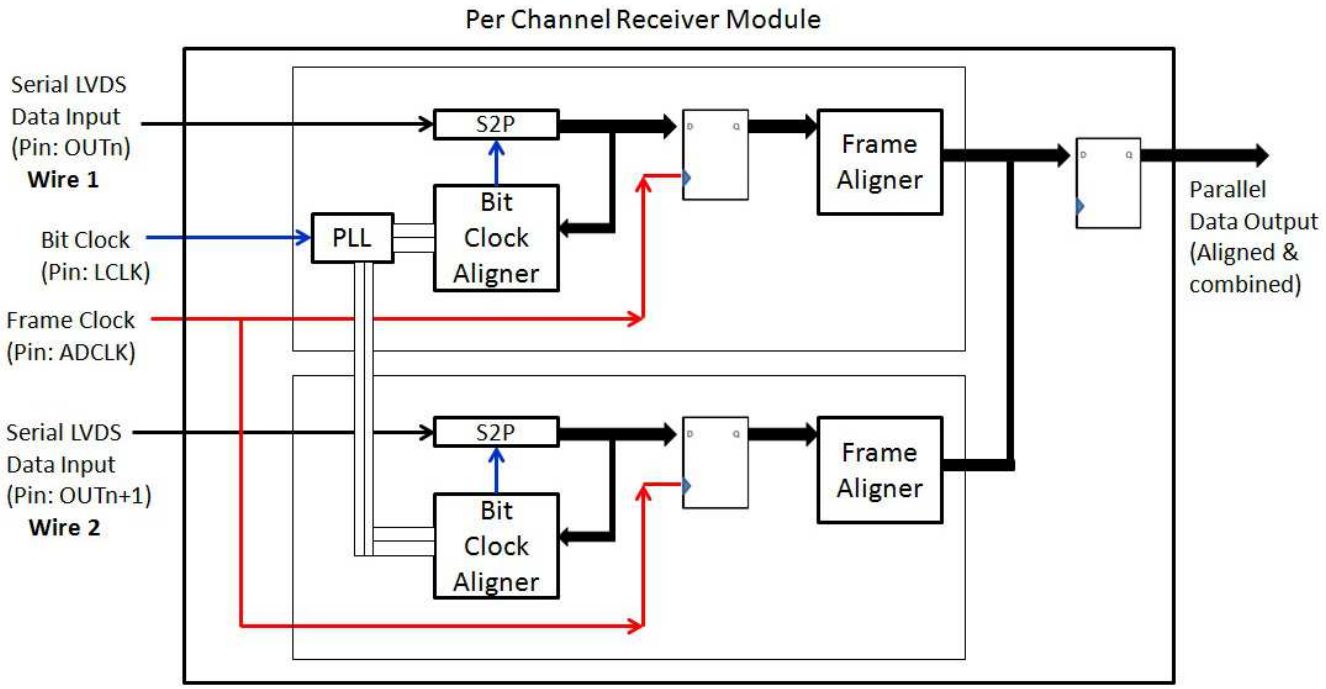


Figure 4-3. Capture Scheme for Two-Wire Interface

The capture schemes in Figure 4-1 to Figure 4-3 are valid even multiple multichannel ADCs are used in the system. Figure 4-4 and Figure 4-5 describe two recommended ways of capturing data from multiple ADC devices. Specifically, note that the individual bit clock output from each ADC is used by the respective capture module. This architecture provides better setup and hold timing compared to using a common bit clock across all devices. (See the *Understanding Source Synchronous Interface* section).

Therefore, the FPGA code consists of a dedicated capture module per ADC device (supporting all eight channels within one device). The output of the capture module is a parallel data bus corresponding to the eight channels of one ADC device, running from a 1x clock derived within the module. For example, this bus is 96-bits wide (12x8 bits) for a 12-bit, 8-channel ADC device.

Note that the 1x clock used within the capture modules are not aligned together. This misalignment is primarily because of the various input clock to output clock delays across the ADC devices.

The parallel data output from all capture modules are then latched using a common system clock that runs at 1x frequency (or at the same frequency as the ADC sample rate). Note that for multiple devices, TI recommends using a common system clock to combine the parallel data from all devices, rather than using a common high-speed clock (bit clock) to combine the serial data from all devices. When multiple devices are used, using the individual bit clock output from each device (as explained in the *Understanding Source Synchronous Interface* section) is advantageous.

While combining the parallel data from all devices, the system clock frequency is at the same frequency (or 1x) as the ADC sample rate. At this low frequency, a sufficient timing margin is available even if the parallel data outputs from multiple devices are misaligned.

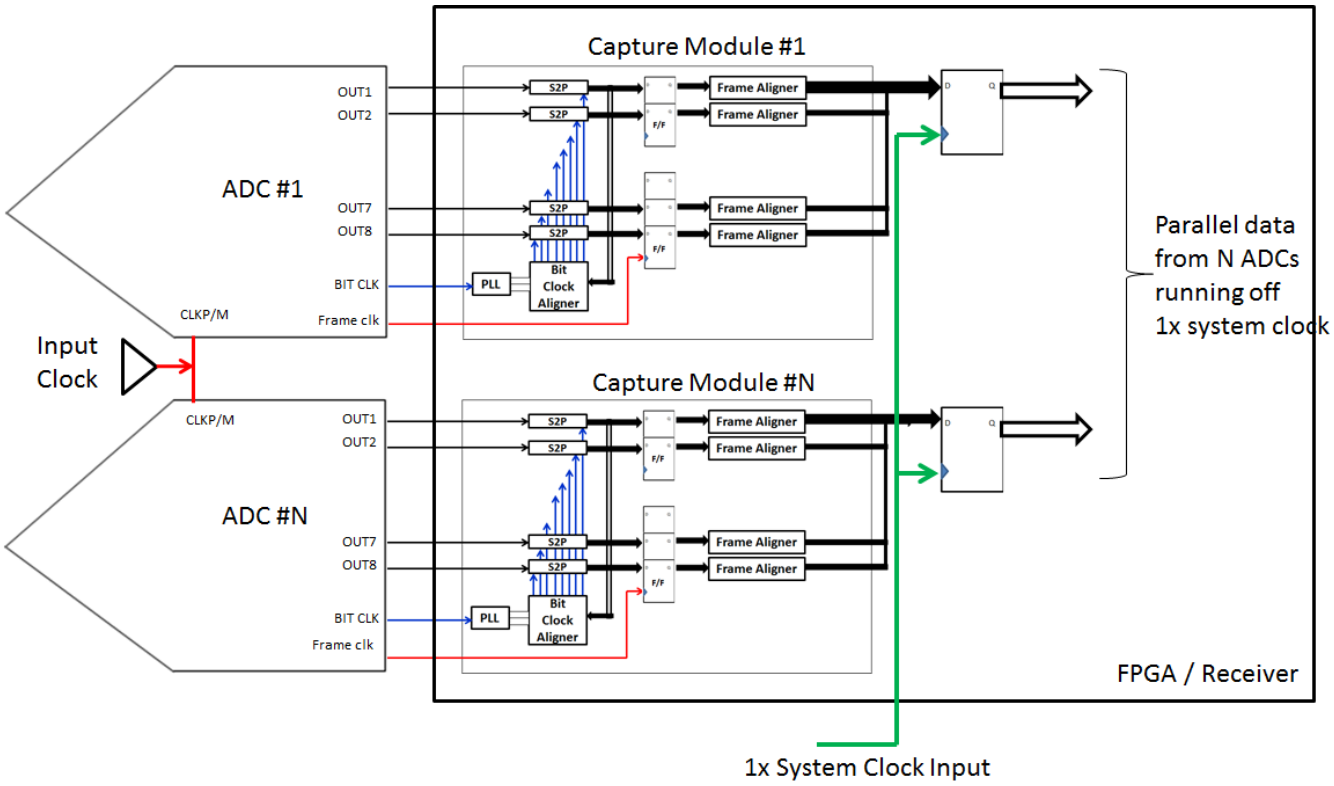


Figure 4-4. Capture Scheme for Multiple ADC Devices using PLLs

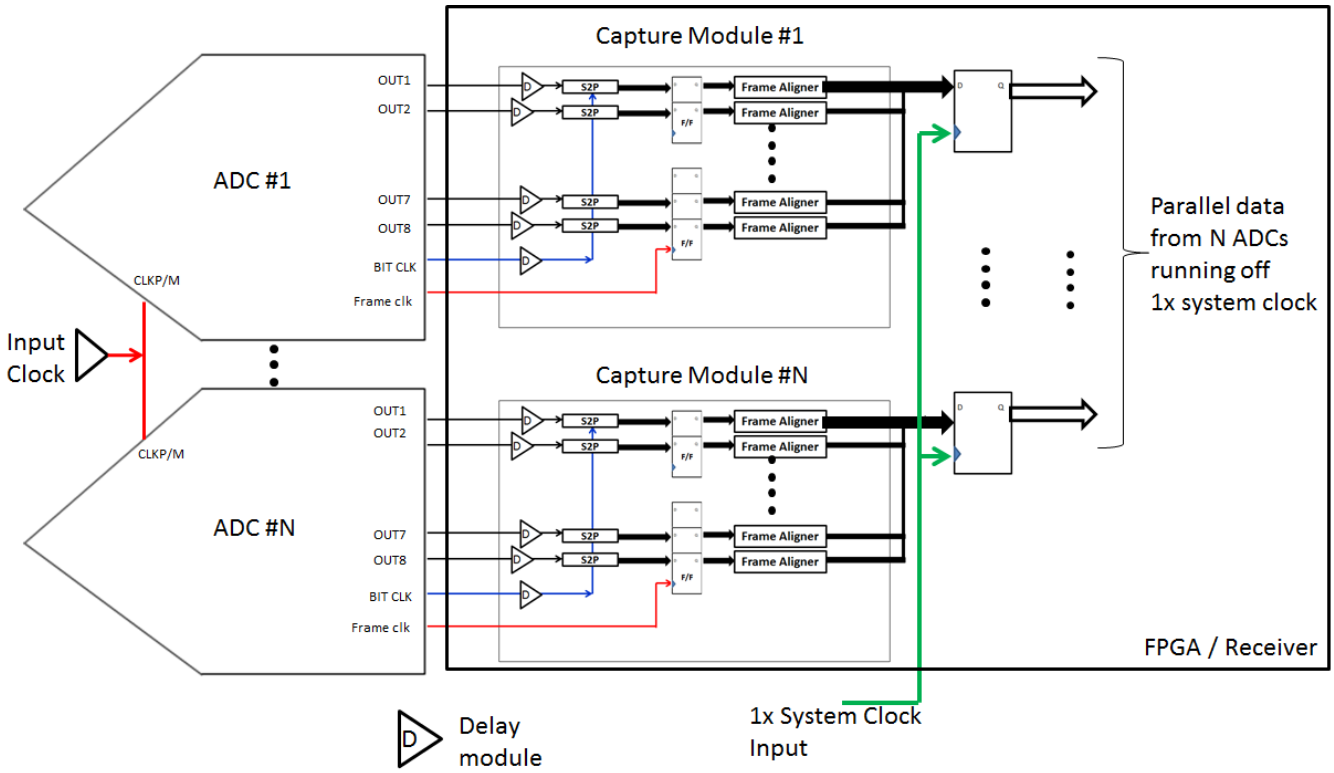
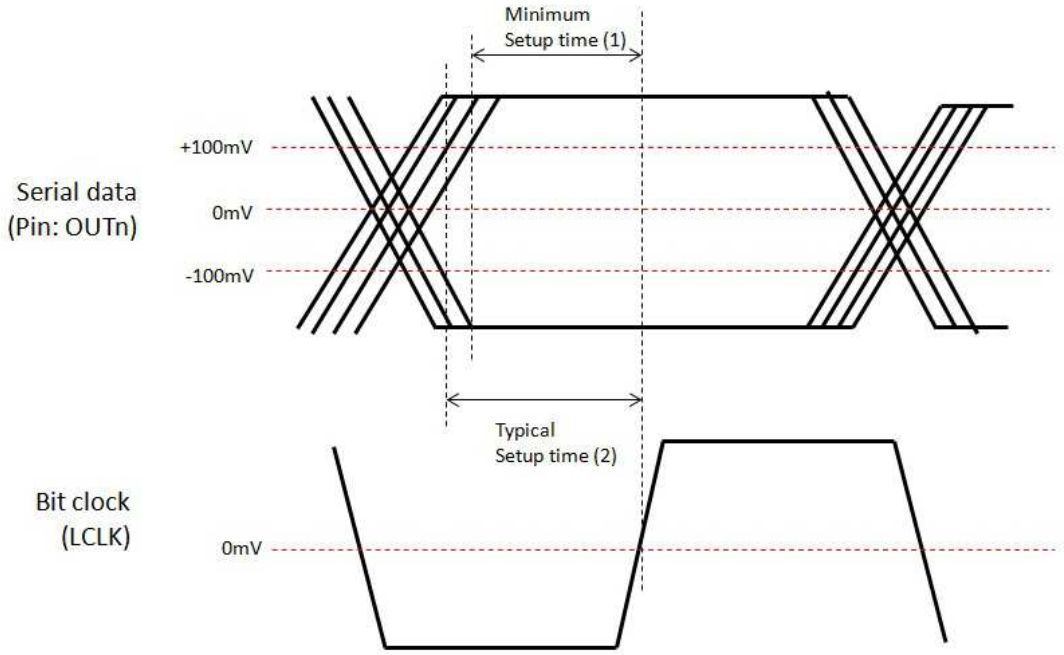


Figure 4-5. Capture Scheme for Multiple ADC Devices using Delays

Understanding ADC Interface Timing Specifications

TI high-speed ADC data sheets specify the LVDS interface setup and hold times measured with respect to the output bit clock. See the *Understanding Source Synchronous Interface* section for further information regarding the advantages of using the output bit clock from each device.

Setup time is the time delay from the instant when data becomes valid to the rising or falling bit clock edge. The minimum value of this time delay is specified as the minimum setup time in the data sheet. Refer to Figure 5-1 for the typical and minimum setup times.

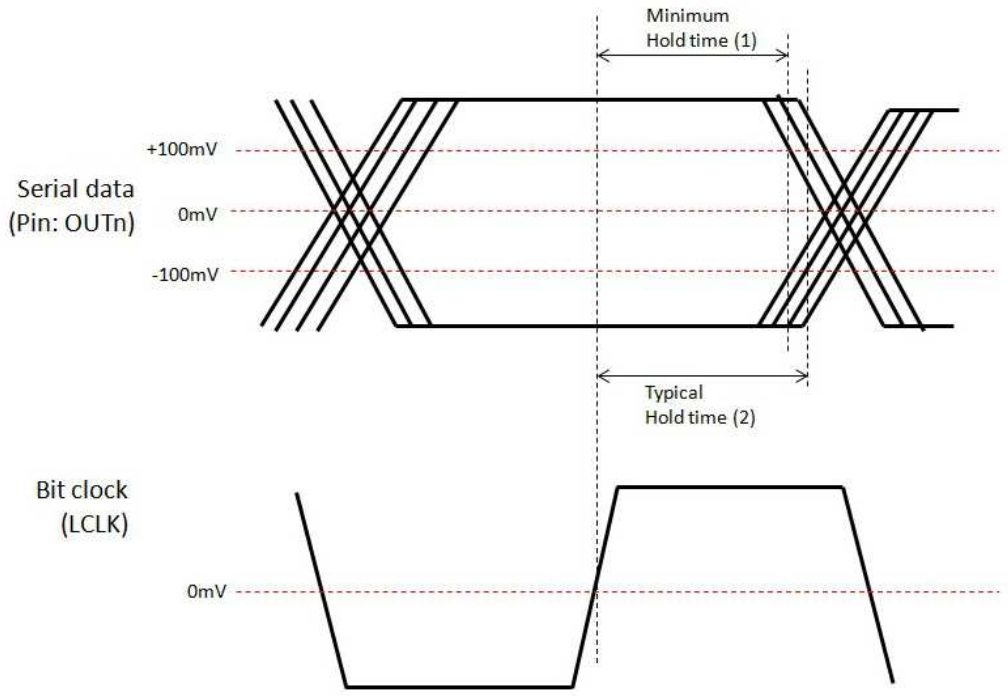


(1) Refer to the t_{SU} minimum specification in the Timing Characteristics table of the device data sheet.

(2) Refer to the t_{SU} typical specification in the Timing Characteristics table of the device data sheet.

Figure 5-1. Setup Time Definition

Hold time is the time delay from the rising or falling bit clock edge to the instant when data becomes invalid. The minimum value of this time delay is specified as the minimum hold time in the data sheet. Refer to Figure 5-2 for the typical and minimum hold times.



(1) Refer to the t_{HO} minimum specification in the Timing Characteristics table of the device data sheet.

(2) Refer to the t_{HO} typical specification in the Timing Characteristics table of the device data sheet.

Figure 5-2. Hold Time Definition

Data eye is the solid blue box in Figure 5-3 and represents the amount of time for which the data logic levels are valid. Having as wide and tall an eye as possible is preferable.

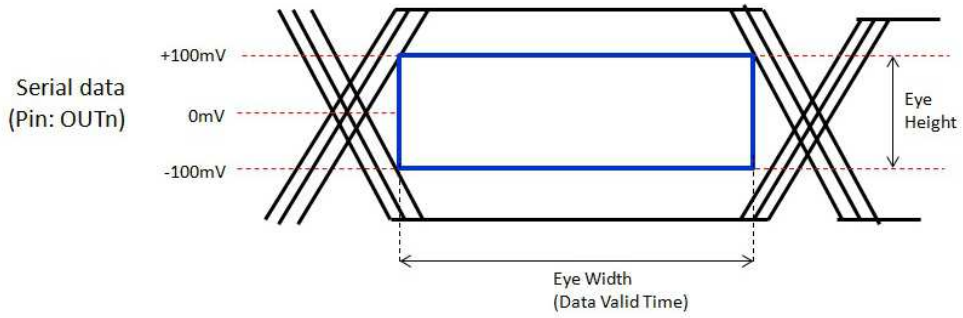


Figure 5-3. Data Valid Time Definition

Referring to Figure 5-1 to Figure 5-3, data are considered valid for logic levels greater than or equal to ±100 mV and are invalid otherwise. The waveforms in Figure 5-1 and Figure 5-2 can be displayed on an oscilloscope by probing the serial output data and bit clock in infinite persistence mode and triggering on the output bit clock. The resulting serial data waveform is also referred to as an *eye*. Note that the serial data waveform crosses the ±100-mV thresholds at multiple points, which implies that the setup time (or hold time) is sometimes large or small at certain times. This effect is a result of jitter in the output data and bit clock. The minimum value of the setup time (or hold time) is considered as the worst-case value and is the specified value in the TI data sheet.

Ideally, the setup time for all LVDS pairs should be identical. Though care is taken to match delays across LVDS outputs within the die, there is skew among LVDS channels caused by mismatches within the die and within the package because of asymmetric pin locations. As a result, the setup and hold times are different across LVDS channels. Figure 5-4 shows an example of an 8-channel ADC with four LVDS outputs. Note that the channel 3 output (OUT3) has the worst setup time among all channels. The minimum setup time on OUT3 is specified in the data sheet. Similarly, note that the channel 4 output (OUT4) has the worst hold time among all channels. The minimum hold time on OUT4 is likewise specified in the data sheet.

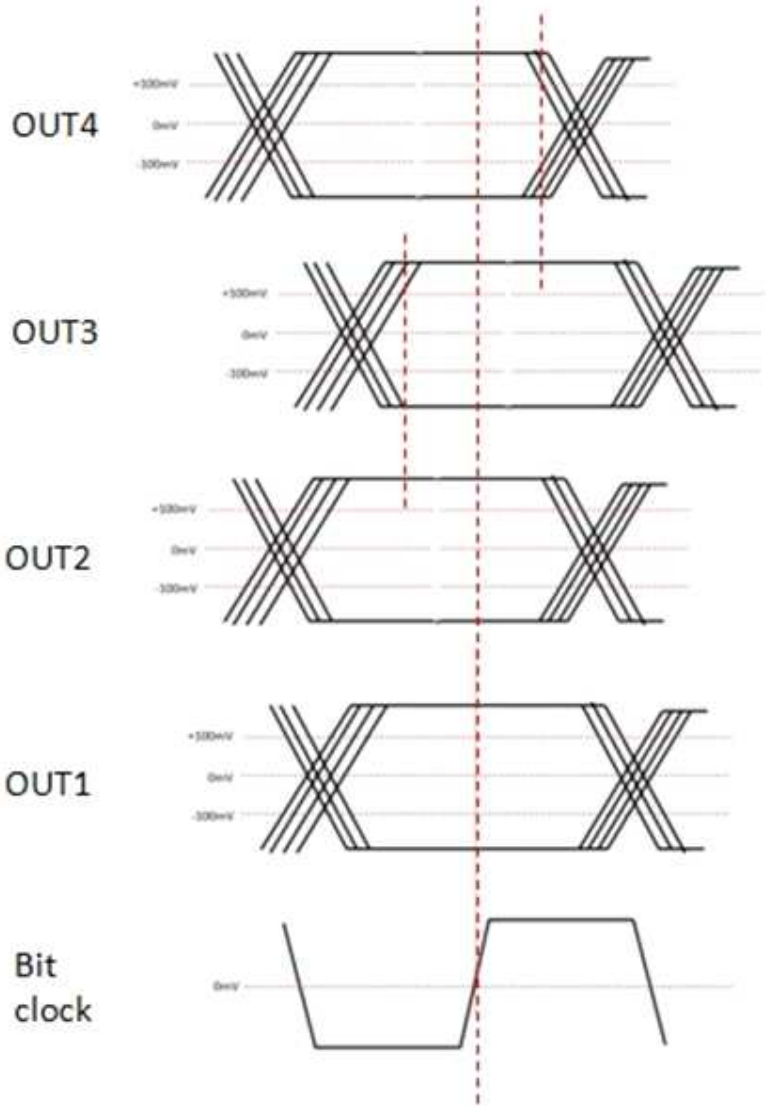


Figure 5-4. Skew Among LVDS Outputs of the ADC

To summarize, the minimum setup and hold specifications in the data sheet already take into account effects of jitter in output data and clock and skew across LVDS outputs.

Achieving Timing Closure in the System

A typical system has multiple ADCs connected to the receiver using one of the schemes described in the *Summary of Capture Schemes* section. Timing analysis methods are used to verify if all data outputs (from the ADCs) satisfy the setup and hold timing specifications of the receiver. Timing analysis takes into account:

- The setup and hold specifications of the transmitter (in this case, the ADC).
- Delays in the data path, which are composed of PCB trace delays and internal delays in the receiver itself.
- Delays in the clock path, which are also composed of PCB trace delays and internal delays in the receiver itself.
- Minimum setup and hold specification of the receiver flip-flop (t_{SU_RX}, min and t_{HO_RX}, min).

Figure 6-1 shows a model for the timing analysis that includes the additional delays resulting from PCB traces and receiver internal routing. For the receiver flip-flop to latch data correctly, the setup and hold times of the input data at the receiver must satisfy Equation 1 and Equation 2.

Actual Setup Time (Equal to Clock Arrival Time – Data Valid Time) > t_{SU_RX}, min (1)

Actual Hold Time (Equal to Data Invalid Time – Clock Arrival Time) > t_{HO_RX}, min (2)

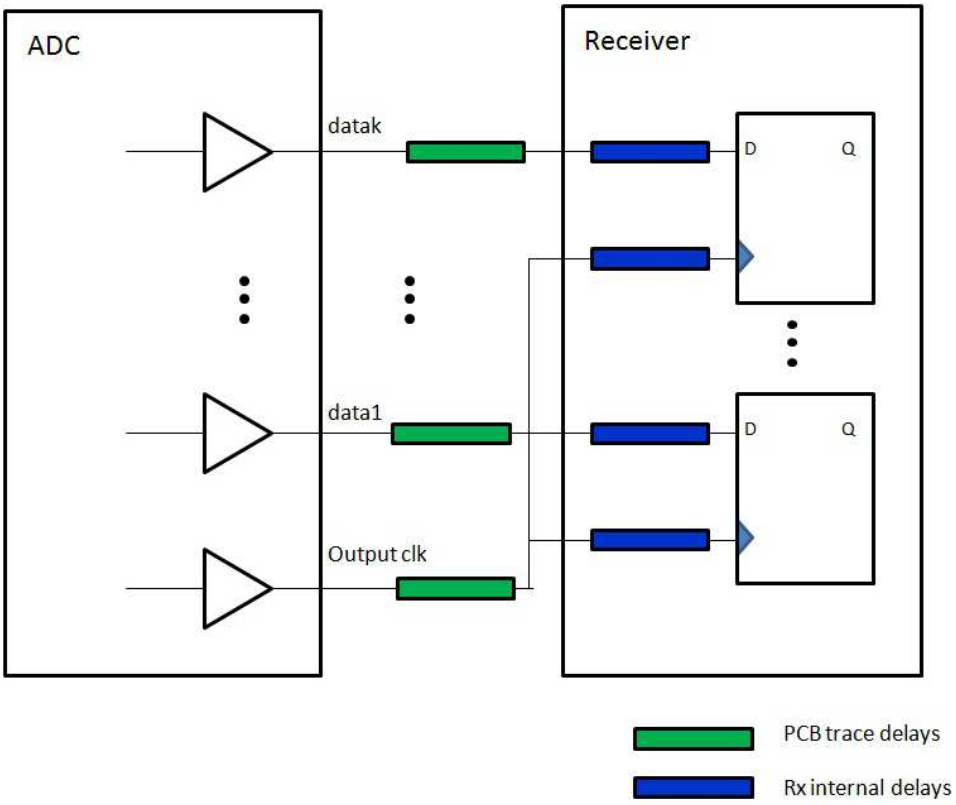


Figure 6-1. Timing Analysis

6.1 Actual Setup Time

The actual setup time can be derived using Equation 3 to Equation 5 and by referring to Figure 6-2.

Clock Arrival Time = $t_{SU\_ADC} + t_{PCB\_CLK} + t_{INT\_CLK} = t_{SU\_ADC} + t_{DEL\_CLK}$ (3)

Data Valid Time = $t_{PCB\_DATA} + t_{INT\_DATA} = t_{DEL\_DATA}$ (4)

Actual Receiver Setup Time = $t_{SU\_ADC} + t_{DEL\_CLK} – t_{DEL\_DATA} = t_{SU\_ADC} – t_{SKEW}$ (5)

The worst-case setup time occurs with minimum ADC setup time, minimum clock delay, and maximum data delay, as described in Equation 6.

Worst-Case Setup Time at Receiver = t_{SU_ADC} (min) – t_{SKEW} (max)

where:

- t_{SKEW} (max) = t_{DEL_DATA} (max) – t_{DEL_CLK} (min) (6)

Usually, data path delays are larger than clock path delays. In such a case, t_{SKEW} is a positive number and Equation 6 shows that the setup time at the receiver is less than the setup time provided by the ADC. Therefore, TI recommends minimizing data path delays (or matching the data and clock path delays) to minimize t_{SKEW}.

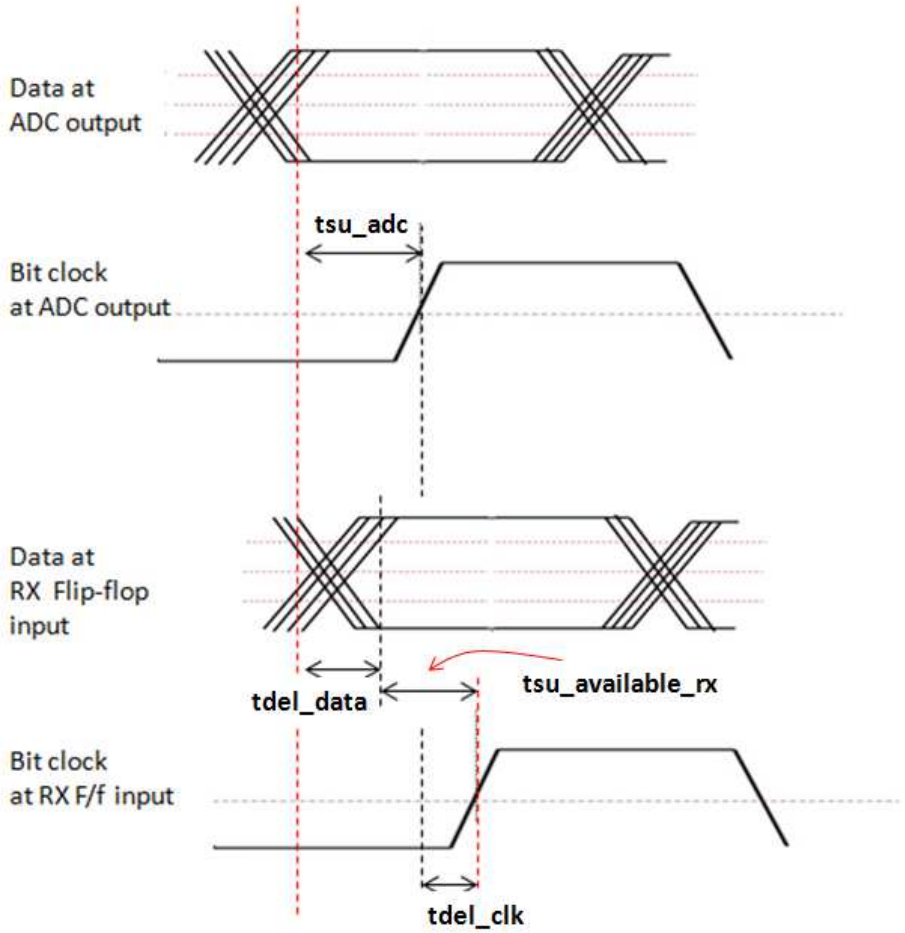


Figure 6-2. Actual Setup Time

6.2 Actual Hold Time

The actual hold time can be derived using Equation 7 to Equation 9 and by referring to Figure 6-3.

Data Invalid Time = $t_{HO\_ADC} + t_{PCB\_DATA} + t_{INT\_DATA} = t_{HO\_ADC} + t_{DEL\_DATA}$ (7)

Clock Arrival Time = $t_{PCB\_CLK} + t_{INT\_CLK} = t_{DEL\_CLK}$ (8)

Actual Hold Time at Receiver = $t_{HO\_ADC} + t_{DEL\_DATA} - t_{DEL\_CLK} = t_{HOD\_ADC} + t_{SKEW}$ (9)

The worst-case hold time occurs with minimum ADC hold time, minimum data delay, and maximum clock delay, as described in Equation 10.

Worst-Case Hold Time at Receiver = t_{HOD_ADC} (min) + t_{SKEW} (min)

where:

- t_{SKEW} (min) = t_{DEL_DATA} (min) – t_{DEL_CLK} (max) (10)

Again, Equation 10 shows that minimizing data path delays (or matching the data and clock path delays) in order to minimize t_{SKEW} is preferable.

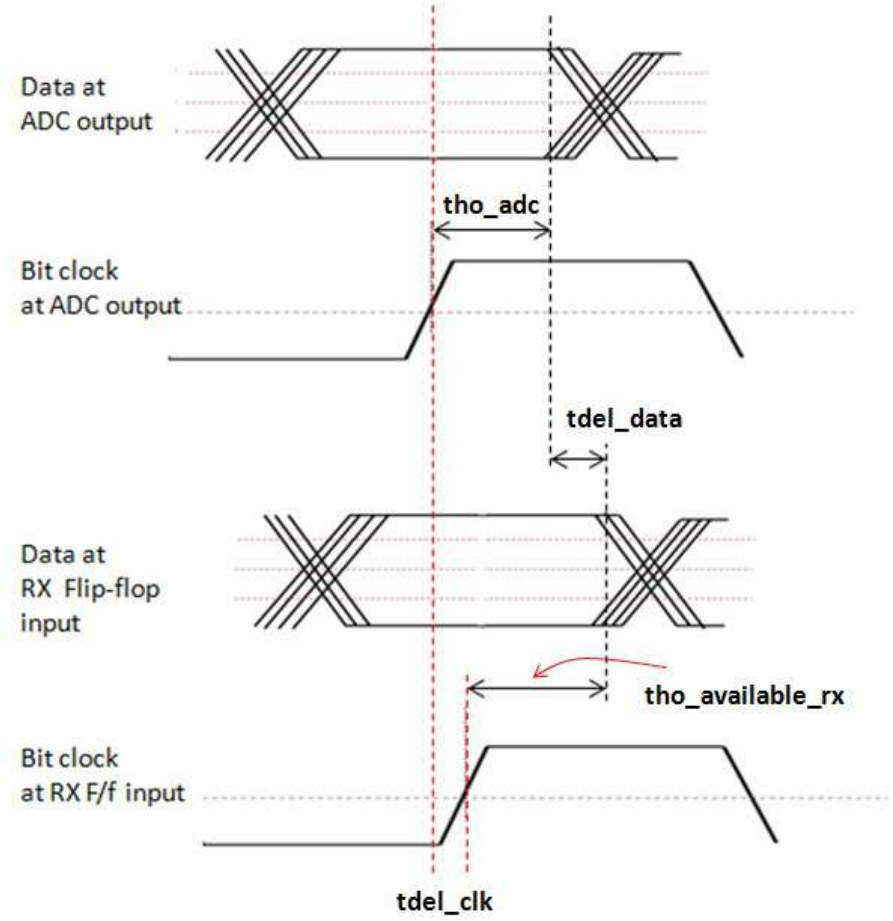


Figure 6-3. Actual Hold Time

The analysis for actual setup and hold times must be done for all the data lines. Timing analysis can be referred to as *closed* when the actual setup (and hold) time at the receiver is greater than the minimum setup (and hold) time specification of the receiver for every data input.

6.3 PCB Skew

As discussed previously, minimizing the skew between the output data and output clock signals is advantageous. One major component of the skew results from PCB trace delays, see Figure 6-1. Minimizing PCB skew is especially important for higher data rates. PCB traces result in 150 ps to 200 ps per inch of trace length depending on the type of traces used (micro-strip versus strip line). For data rates greater than 500 Mbps, TI recommends matching the PCB trace lengths for data and clock signals to within 100 mils of each other. This match ensures that the skew is low (approximately 15 ps to 20 ps) and maximizes the setup and hold times at the receiver.

Understanding Source Synchronous Interface

The serial LVDS interface consists of serial data outputs (one or two outputs per ADC channel) accompanied by a bit clock output (refer to Figure 1-1). This interface, in which the clock is transmitted along with data, is referred to as a *source-synchronous interface*.

Source-synchronous interfaces are especially beneficial at high data rates, whereby matching output data and output clock signal routing, any skews between them can be minimized. Another important benefit of this interface is with respect to the effect of jitter. Jitter is the instantaneous deviation of the rising and falling edges of data and clock signals. Jitter is primarily caused by supply noise, where vertical noise is translated to horizontal or time-instant deviations, depending on the signal rise and fall times, as shown in Figure 7-1.

In a serial LVDS transmitter, both output data and clock signals have jitter. Jitter has two components: correlated and uncorrelated. By carefully matching clock and data paths, the data and clock jitter can be ensured to be largely correlated.

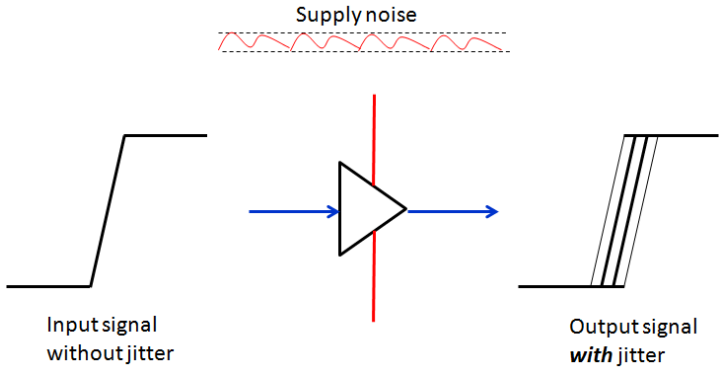


Figure 7-1. Jitter Source

Figure 7-2 shows the effect of correlated jitter where the instantaneous time deviations of the output data and clock signals are identical. Thus, when the data transition is delayed (the red line), the clock edge is also delayed similarly and when the data transition is advanced (the green line), the clock edge also arrives earlier compared to the nominal or ideal time instants (the black lines).

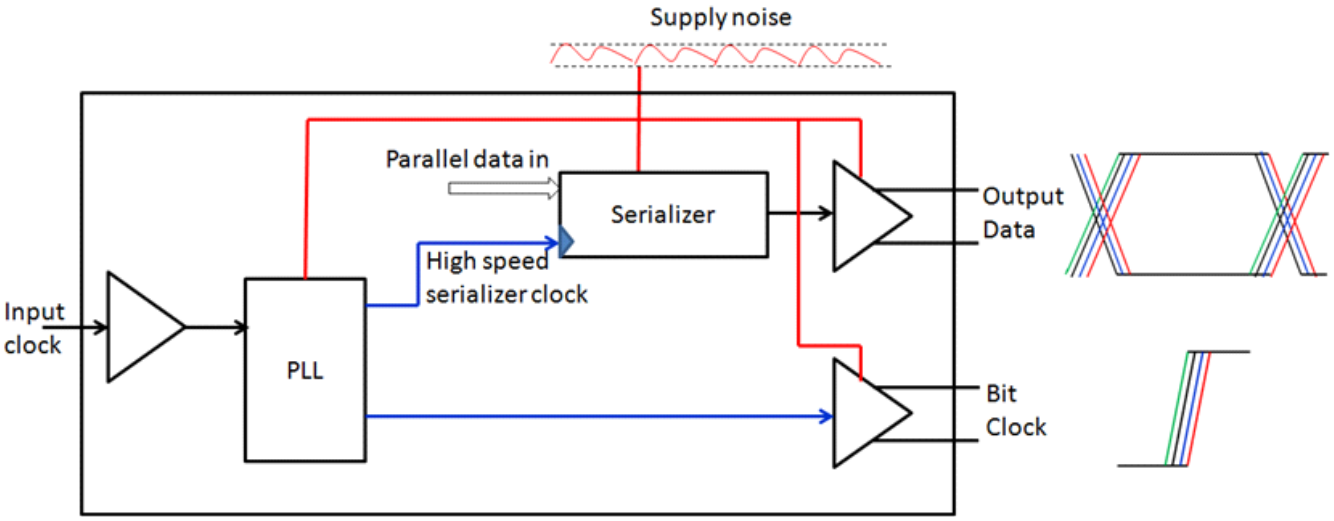


Figure 7-2. Jitter in Data and Clock Paths

When the output data timing is measured relative to the output clock, the correlated (or common) jitter component in the output data and bit clock cancels, resulting in a larger eye (or available valid data time). Effectively, this cancellation results in better setup and hold timing specifications when measured with respect to the output clock.

Figure 7-3 shows the data eye for two cases.

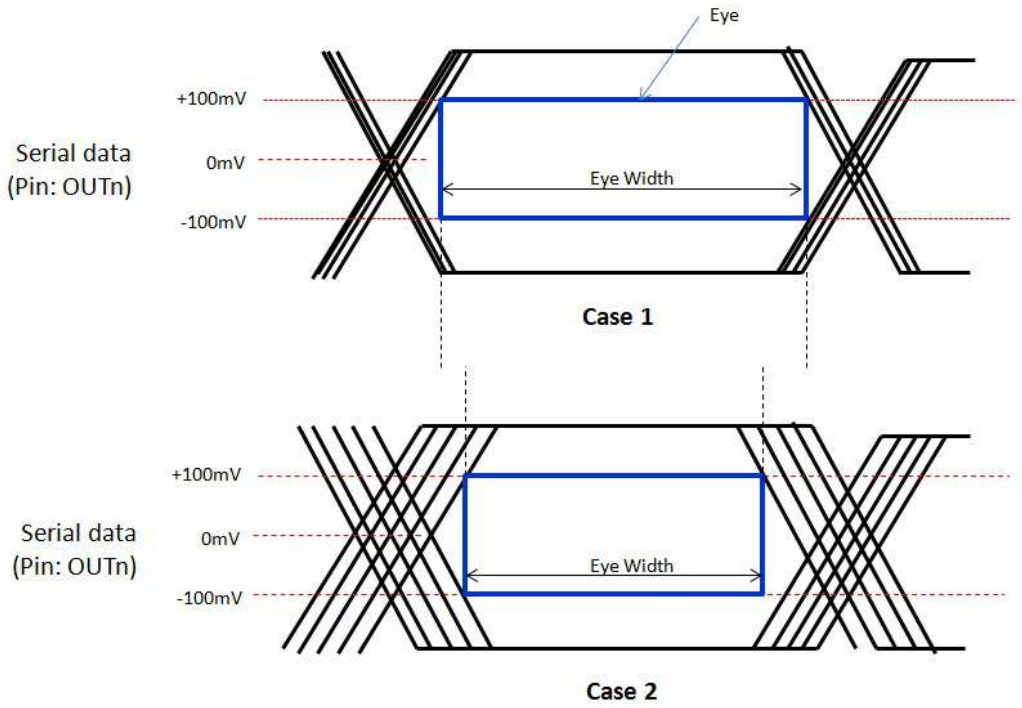


Figure 7-3. Eye Using an Internal Bit Clock (Case 1) and External Bit Clock (Case 2)

Case 1: In this case, the eye is the result when output data are measured by triggering the oscilloscope on the output bit clock of the ADC (see Figure 7-2). As explained before, the jitter is largely correlated and explains the resultant wide eye. This case is the condition under which TI specifies the output LVDS timing parameters of setup and hold times.

Case 2: In this case, the eye is the result when output data are measured by triggering the oscilloscope using an external bit clock that may be generated, for example, using an external PLL from the input clock, as shown in Figure 7-4. In this case, the correlated component of jitter is very small. The eye is noticeably smaller compared to the first case because most of the jitter is uncorrelated. This case effectively results in worse setup and hold timings when output data are measured with respect to the external bit clock.

The main conclusion from this discussion is that using the output bit clock from the ADC to latch the serialized output data is recommended, especially at high data rates, because of the wide eye available with this scheme.

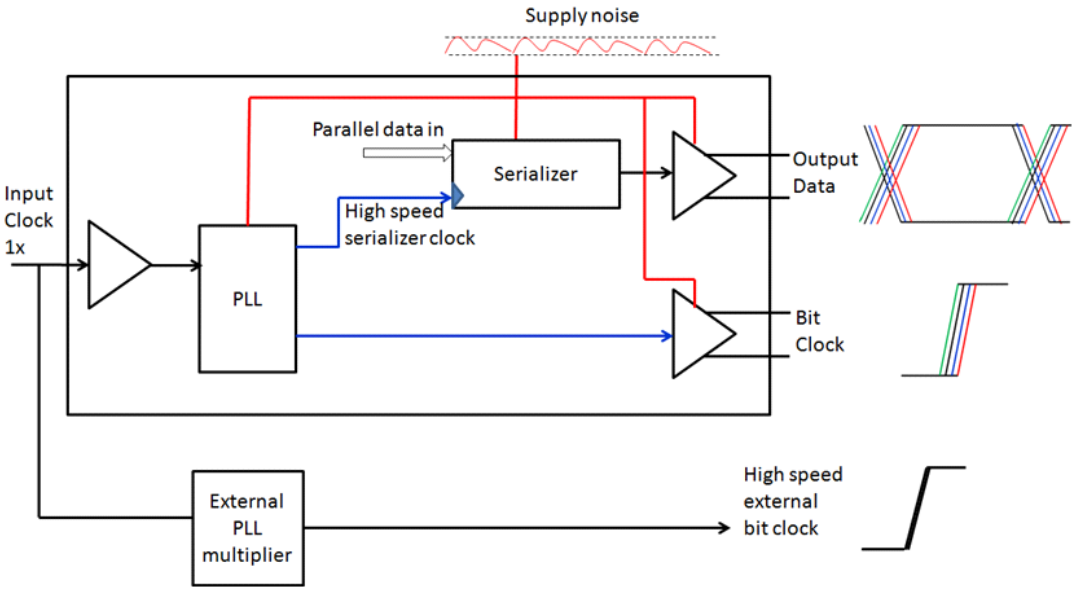


Figure 7-4. Jitter in Data and Clock Paths Compared with an External or *Ideal* Clock

IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, enhancements, improvements and other changes to its semiconductor products and services per JESD46, latest issue, and to discontinue any product or service per JESD48, latest issue. Buyers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All semiconductor products (also referred to herein as “components”) are sold subject to TI’s terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its components to the specifications applicable at the time of sale, in accordance with the warranty in TI’s terms and conditions of sale of semiconductor products. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by applicable law, testing of all parameters of each component is not necessarily performed.

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

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

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

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

Buyer acknowledges and agrees that it is solely responsible for compliance with all legal, regulatory and safety-related requirements concerning its products, and any use of TI components in its applications, notwithstanding any applications-related information or support that may be provided by TI. Buyer represents and agrees that it has all the necessary expertise to create and implement safeguards which anticipate dangerous consequences of failures, monitor failures and their consequences, lessen the likelihood of failures that might cause harm and take appropriate remedial actions. Buyer will fully indemnify TI and its representatives against any damages arising out of the use of any TI components in safety-critical applications.

In some cases, TI components may be promoted specifically to facilitate safety-related applications. With such components, TI’s goal is to help enable customers to design and create their own end-product solutions that meet applicable functional safety standards and requirements. Nonetheless, such components are subject to these terms.

No TI components are authorized for use in FDA Class III (or similar life-critical medical equipment) unless authorized officers of the parties have executed a special agreement specifically governing such use.

Only those TI components which TI has specifically designated as military grade or “enhanced plastic” are designed and intended for use in military/aerospace applications or environments. Buyer acknowledges and agrees that any military or aerospace use of TI components which have ***not*** been so designated is solely at the Buyer's risk, and that Buyer is solely responsible for compliance with all legal and regulatory requirements in connection with such use.

TI has specifically designated certain components as meeting ISO/TS16949 requirements, mainly for automotive use. In any case of use of non-designated products, TI will not be responsible for any failure to meet ISO/TS16949.

Products		**Applications**	
Audio	www.ti.com/audio	Automotive and Transportation	www.ti.com/automotive
Amplifiers	amplifier.ti.com	Communications and Telecom	www.ti.com/communications
Data Converters	dataconverter.ti.com	Computers and Peripherals	www.ti.com/computers
DLP® Products	www.dlp.com	Consumer Electronics	www.ti.com/consumer-apps
DSP	dsp.ti.com	Energy and Lighting	www.ti.com/energy
Clocks and Timers	www.ti.com/clocks	Industrial	www.ti.com/industrial
Interface	interface.ti.com	Medical	www.ti.com/medical
Logic	logic.ti.com	Security	www.ti.com/security
Power Mgmt	power.ti.com	Space, Avionics and Defense	www.ti.com/space-avionics-defense
Microcontrollers	microcontroller.ti.com	Video and Imaging	www.ti.com/video
RFID	www.ti-rfid.com		
OMAP Applications Processors	www.ti.com/omap	**TI E2E Community**	e2e.ti.com
Wireless Connectivity	www.ti.com/wirelessconnectivity		

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