5. High-Speed Differential I/O Interfaces in Stratix Devices

S52005-3.1

Introduction

To achieve high data transfer rates, Stratix® devices support True-LVDS™ differential I/O interfaces which have dedicated serializer/deserializer (SERDES) circuitry for each differential I/O pair. Stratix SERDES circuitry transmits and receives up to 840 megabits per second (Mbps) per channel. The differential I/O interfaces in Stratix devices support many high-speed I/O standards, such as LVDS, LVPECL, PCML, and HyperTransport™ technology. Stratix device high-speed modules are designed to provide solutions for many leading protocols such as SPI-4 Phase 2, SFI-4, 10G Ethernet XSBI, RapidIO, HyperTransport technology, and UTOPIA-4.

The SERDES transmitter is designed to serialize 4-, 7-, 8-, or 10-bit wide words and transmit them across either a cable or printed circuit board (PCB). The SERDES receiver takes the serialized data and reconstructs the bits into a 4-, 7-, 8-, or 10-bit-wide parallel word. The SERDES contains the necessary high-frequency circuitry, multiplexer, demultiplexer, clock, and data manipulation circuitry. You can use double data rate I/O (DDRIO) circuitry to transmit or receive differential data in by-one (×1) or by-two (×2) modes.

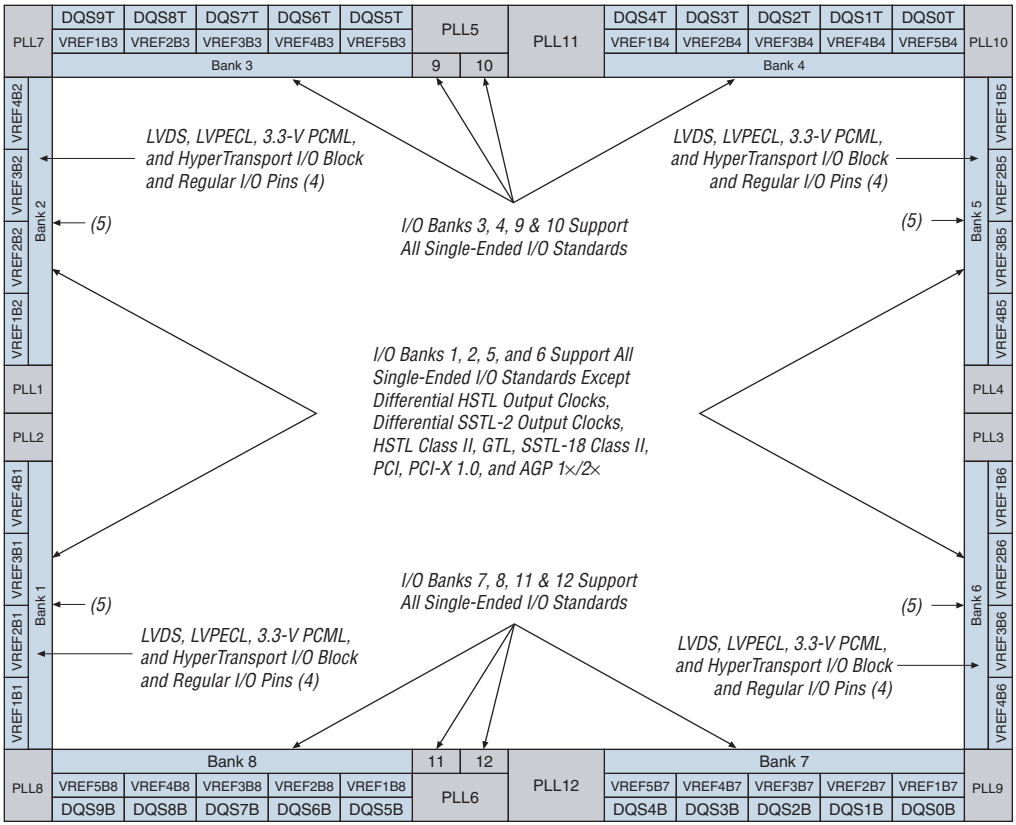
Contact Altera Applications for more information on other *B* values that the Stratix devices support and using ×7-mode in the Quartus® II software. Stratix devices currently only support *B* = 1 and B = 7 in ×7 mode.

This chapter describes the high-speed differential I/O capabilities of Stratix programmable logic devices (PLDs) and provides guidelines for their optimal use. You should use this document in conjunction with the *Stratix Device Family Data Sheet* section of the *Stratix Device Handbook, Volume 1*. Consideration of the critical issues of controlled impedance of traces and connectors, differential routing, termination techniques, and DC balance gets the best performance from the device. Therefore, an elementary knowledge of high-speed clock-forwarding techniques is also helpful.

Stratix I/O Banks

Stratix devices contain eight I/O banks, as shown in Figure 5–1. The two I/O banks on each side contain circuitry to support high-speed LVDS, LVPECL, PCML, HSTL class I and II, SSTL-2 class I and II, and HyperTransport inputs and outputs.

Figure 5–1. Stratix I/O Banks *Notes (1), (2), (3)*



Notes to Figure 5–1:

(1) Figure 5–1 is a top view of the Stratix silicon die, which corresponds to a top-down view of non-flip-chip packages and a bottom-up view of flip-chip packages.

(2) Figure 5–1 is a graphic representation only. See the pin list and the Quartus II software for exact locations.

(3) Banks 9 through 12 are enhanced PLL external clock output banks.

(4) If the high-speed differential I/O pins are not used for high-speed differential signaling, they can support all of the I/O standards except HSTL class I and II, GTL, SSTL-18 Class II, PCI, PCI-X 1.0, and AGP 1×/2×.

(5) See "Differential Pad Placement Guidelines" on page 4–30. You can only place single-ended output/bidirectional pads five or more pads away from a differential pad. Use the **Show Pads** view in the Quartus II Floorplan Editor to locate these pads. The Quartus II software gives an error message for illegal output or bidirectional pin placement next to a high-speed differential I/O pin.

Stratix Differential I/O Standards

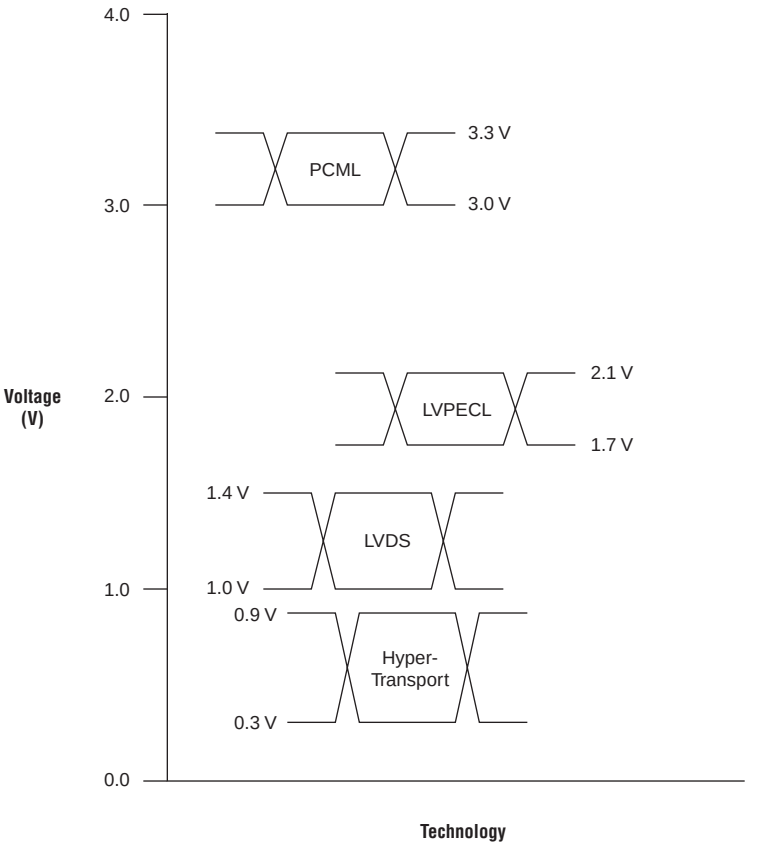
Stratix devices provide a multi-protocol interface that allows communication between a variety of I/O standards, including LVDS, HyperTransport technology, LVPECL, PCML, HSTL class I and II, and

SSTL-2 class I and II. This feature makes the Stratix device family ideal for applications that require multiple I/O standards, such as a protocol translator.

For more information on termination for Stratix I/O standards, see "Differential I/O Termination" on page 5–46.

Figure 5–2 compares the voltage levels between differential I/O standards supported in all the Stratix devices.

Figure 5–2. Differential I/O Standards Supported by Stratix Devices



LVDS

The LVDS I/O standard is a differential high-speed, low-voltage swing, low-power, general-purpose I/O interface standard requiring a 3.3-V V_{CCIO}. This standard is used in applications requiring high-bandwidth data transfer, backplane drivers, and clock distribution. The ANSI/TIA/EIA-644 standard specifies LVDS transmitters and receivers capable of operating at recommended maximum data signaling rates of 655 Mbps. However, devices can operate at slower speeds if needed, and there is a theoretical maximum of 1.923 Gbps. Stratix devices meet the ANSI/TIA/EIA-644 standard.

Due to the low voltage swing of the LVDS I/O standard, the electromagnetic interference (EMI) effects are much smaller than CMOS, transistor-to-transistor logic (TTL), and PECL. This low EMI makes LVDS ideal for applications with low EMI requirements or noise immunity requirements. The LVDS standard specifies a differential output voltage range of $0.25\ V \times V_{OD} \leq 0.45\ V$. The LVDS standard does not require an input reference voltage, however, it does require a 100-Ω termination resistor between the two signals at the input buffer. Stratix devices include an optional differential termination resistor within the device. See the Stratix Device Family Data Sheet section of the *Stratix Device Handbook*, Volume 1 for the LVDS parameters.

HyperTransport Technology

The HyperTransport technology I/O standard is a differential high-speed, high-performance I/O interface standard requiring a 2.5-V VCCIO. This standard is used in applications such as high-performance networking, telecommunications, embedded systems, consumer electronics, and Internet connectivity devices. The HyperTransport technology I/O standard is a point-to-point standard in which each HyperTransport technology bus consists of two point-to-point unidirectional links. Each link is 2 to 32 bits. See the *Stratix Device Family Data Sheet* section of the *Stratix Device Handbook, Volume 1* for the HyperTransport parameters.

LVPECL

The LVPECL I/O standard is a differential interface standard requiring a 3.3-V $V_{CCIO.}$ The standard is used in applications involving video graphics, telecommunications, data communications, and clock distribution. The high-speed, low-voltage swing LVPECL I/O standard uses a positive power supply and is similar to LVDS, however, LVPECL has a larger differential output voltage swing than LVDS. See the *Stratix Device Family Data Sheet* section of the *Stratix Device Handbook, Volume 1* for the LVPECL signaling characteristics.

PCML

The PCML I/O standard is a differential high-speed, low-power I/O interface standard used in applications such as networking and telecommunications. The standard requires a 3.3-V V_{CCIO}. The PCML I/O standard achieves better performance and consumes less power than the LVPECL I/O standard. The PCML standard is similar to LVPECL, but PCML has a reduced voltage swing, which allows for a faster switching time and lower power consumption.See the *Stratix Device Family Data Sheet* section of the *Stratix Device Handbook, Volume 1* for the PCML signaling characteristics.

Differential HSTL (Class I & II)

The differential HSTL I/O standard is used for applications designed to operate in the 0.0- to 1.5-V HSTL logic switching range such as quad data rate (QDR) memory clock interfaces. The differential HSTL specification is the same as the single ended HSTL specification. The standard specifies an input voltage range of $-0.3\ V \leq V_I \leq V_{CCIO} + 0.3\ V$. The differential HSTL I/O standard is only available on the input and output clocks. See the *Stratix Device Family Data Sheet* section of the *Stratix Device Handbook, Volume 1* for the HSTL signaling characteristics

Differential SSTL-2 (Class I & II)

The differential SSTL-2 I/O standard is a 2.5-V memory bus standard used for applications such as high-speed double data rate (DDR) SDRAM interfaces. This standard defines the input and output specifications for devices that operate in the SSTL-2 logic switching range of 0.0 to 2.5 V. This standard improves operation in conditions where a bus must be isolated from large stubs. The SSTL-2 standard specifies an input voltage range of $-0.3\ V \leq V_I \leq V_{CCIO} + 0.3\ V$. Stratix devices support both input and output levels. The differential SSTL-2 I/O standard is only available on output clocks. See the *Stratix Device Family Data Sheet* section of the *Stratix Device Handbook, Volume 1* for the SSTL-2 signaling characteristics.

Stratix Differential I/O Pin Location

The differential I/O pins are located on the I/O banks on the right and left side of the Stratix device. Table 5–1 shows the location of the Stratix device high-speed differential I/O buffers. When the I/O pins in the I/O banks that support differential I/O standards are not used for high-speed

signaling, you can configure them as any of the other supported I/O standards. DDRIO capabilities are detailed in "SERDES Bypass DDR Differential Signaling" on page 5–42.

Table 5–1. I/O Pin Locations on Each Side of Stratix Devices

Device Side *(1)*	Differential Input	Differential Output	DDRIO
Left	✓	✓	✓
Right	✓	✓	✓
Top			✓
Bottom			✓

Note to Table 5–1:
(1) Device sides are relative to pin A1 in the upper left corner of the device (top view of the package).

Principles of SERDES Operation

Stratix devices support source-synchronous differential signaling up to 840 Mbps. Serial data is transmitted and received along with a low-frequency clock. The PLL can multiply the incoming low-frequency clock by a factor of 1 to 10. The SERDES factor *J* can be 4, 7, 8, or 10 and does not have to equal the clock multiplication value. ×1 and ×2 operation is also possible by bypassing the SERDES; it is explained in "SERDES Bypass DDR Differential Interface Review" on page 5–42.

On the receiver side, the high-frequency clock generated by the PLL shifts the serial data through a shift register (also called deserializer). The parallel data is clocked out to the logic array synchronized with the low-frequency clock. On the transmitter side, the parallel data from the logic array is first clocked into a parallel-in, serial-out shift register synchronized with the low-frequency clock and then transmitted out by the output buffers.

There are four dedicated fast PLLs in EP1S10 to EP1S25 devices, and eight in EP1S30 to EP1S80 devices. These PLLs are used for the SERDES operations as well as general-purpose use.

The differential channels and the high-speed PLL layout in Stratix devices are described in the "Differential I/O Interface & Fast PLLs" section on page 5–16.

Stratix Differential I/O Receiver Operation

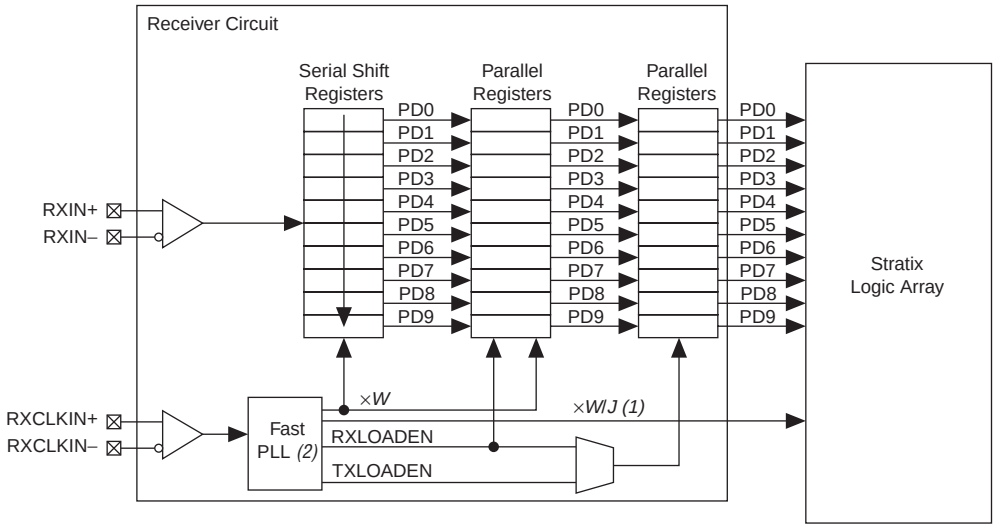
You can configure any of the Stratix differential input channels as a receiver channel (see Figure 5–3). The differential receiver deserializes the incoming high-speed data. The input shift register continuously clocks the incoming data on the negative transition of the high-frequency clock generated by the PLL clock (×*W*).

The data in the serial shift register is shifted into a parallel register by the RXLOADEN signal generated by the fast PLL counter circuitry on the third falling edge of the high-frequency clock. However, you can select which falling edge of the high frequency clock loads the data into the parallel register, using the data-realignment circuit. For more information on the data-realignment circuit, see "Data Realignment Principles of Operation" on page 5–25.

In normal mode, the enable signal RXLOADEN loads the parallel data into the next parallel register on the second rising edge of the low-frequency clock. You can also load data to the parallel register through the TXLOADEN signal when using the data-realignment circuit.

Figure 5–3 shows the block diagram of a single SERDES receiver channel. Figure 5–4 shows the timing relationship between the data and clocks in Stratix devices in ×10 mode. *W* is the low-frequency multiplier and *J* is data parallelization division factor.

Figure 5–3. Stratix High-Speed Interface Deserialized in ×10 Mode

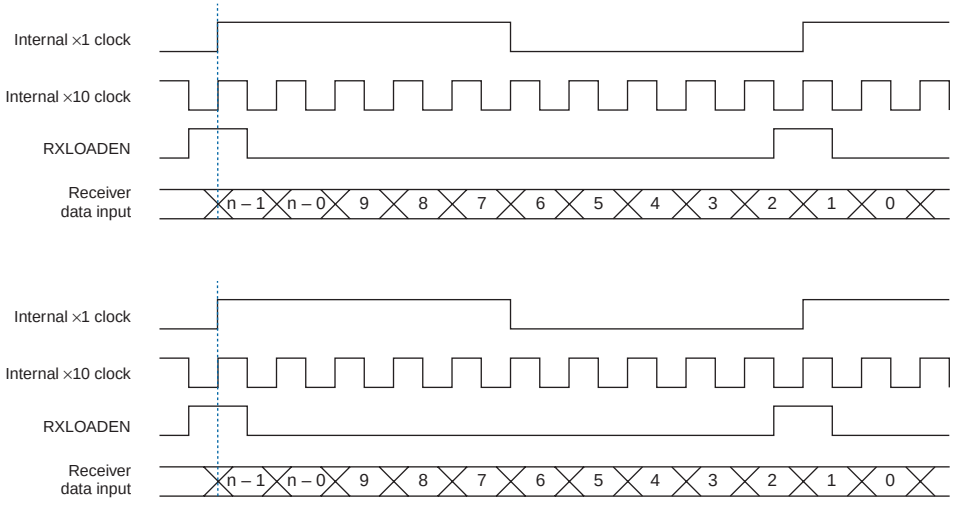


Notes to Figure 5–3:

(1) $W = 1, 2, 4, 7, 8,$ or 10.
$J = 4, 7, 8,$ or 10.
W does not have to equal J. When $J = 1$ or 2, the deserializer is bypassed. When $J = 2$, the device uses DDRIO registers.

(2) This figure does not show additional circuitry for clock or data manipulation.

Figure 5–4. Receiver Timing Diagram



Stratix Differential I/O Transmitter Operation

You can configure any of the Stratix differential output channels as a transmitter channel. The differential transmitter is used to serialize outbound parallel data.

The logic array sends parallel data to the SERDES transmitter circuit when the TXLOADEN signal is asserted. This signal is generated by the high-speed counter circuitry of the logic array low-frequency clock's rising edge. The data is then transferred from the parallel register into the serial shift register by the TXLOADEN signal on the third rising edge of the high-frequency clock.

Figure 5–5 shows the block diagram of a single SERDES transmitter channel and Figure 5–6 shows the timing relationship between the data and clocks in Stratix devices in ×10 mode. *W* is the low-frequency multiplier and *J* is the data parallelization division factor.

Figure 5–5. Stratix High-Speed Interface Serialized in ×10 Mode

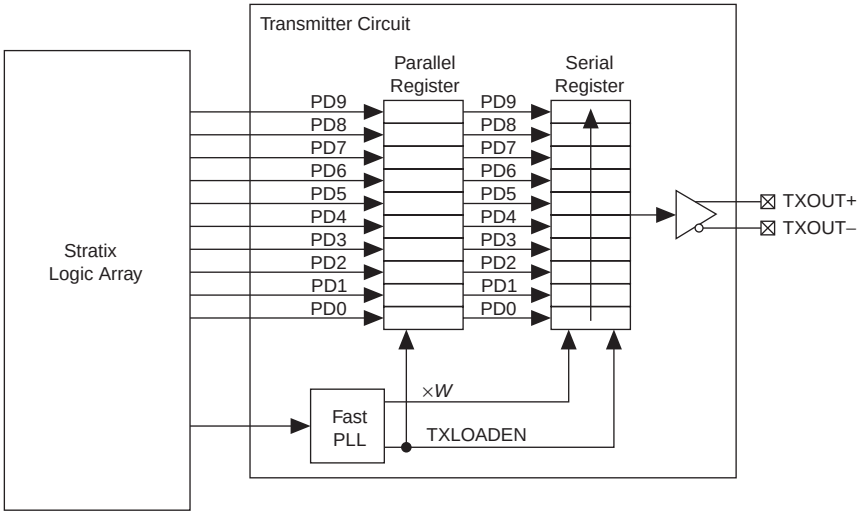
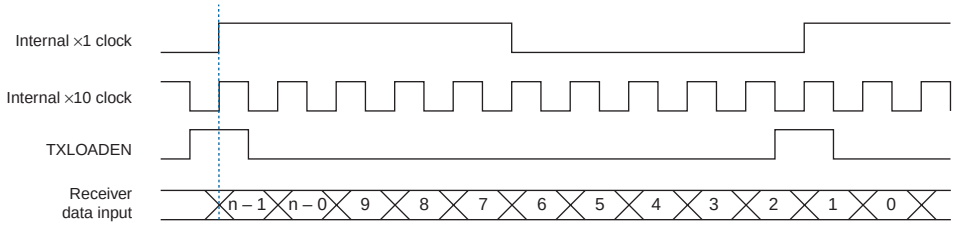


Figure 5–6. Transmitter Timing Diagram



Transmitter Clock Output

Different applications and protocols call for various clocking schemes. Some applications require you to center-align the rising or falling clock edge with the data. Other applications require a divide version of the transmitted clock, or the clock and data to be at the same high-speed frequency. The Stratix device transmitter clock output is versatile and easily programmed for all such applications.

Stratix devices transmit data using the source-synchronous scheme, where the clock is transmitted along with the serialized data to the receiving device. Unlike APEX™ 20KE and APEX II devices, Stratix devices do not have a fixed transmitter clock output pin. The Altera® Quartus II software generates the transmitter clock output by using a fast clock to drive a transmitter `dataout` channel. Therefore, you can place the transmitter clock pair close to the data channels, reducing clock-to-data skew and increasing system margins. This approach is more flexible, as any channel can drive a clock, not just specially designated clock pins.

Divided-Down Transmitter Clock Output

You can divide down the high-frequency clock by 2, 4, 8, or 10, depending on the system requirements. The various options allow Stratix devices to accommodate many different types of protocols. The divided-down clock is generated by an additional transmitting data channel.

Table 5–2 shows the divided-down version of the high-frequency clock and the selected serialization factor *J* (described in pervious sections). The Quartus II software automatically generates the data input to the additional transmitter data channel.

Table 5–2. Differential Transmitter Output Clock Division

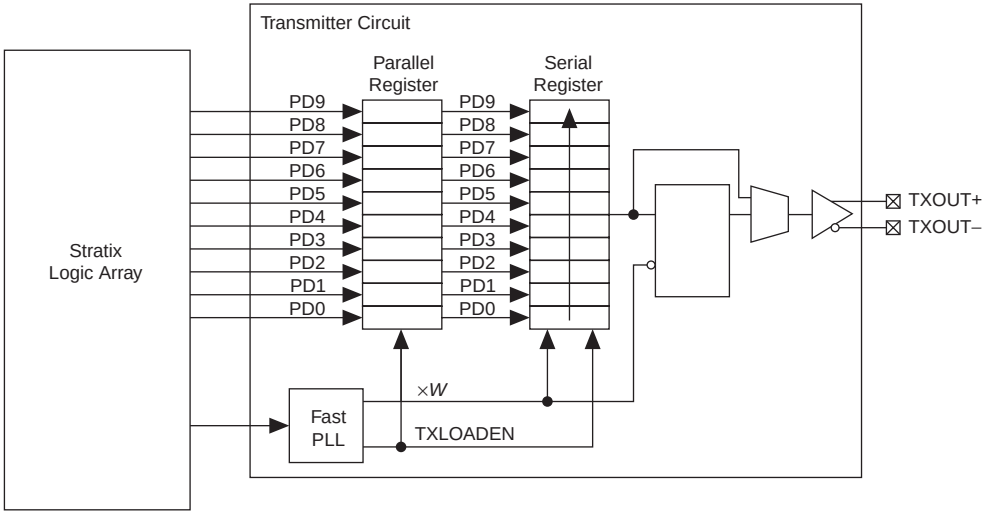
J	Data Input	Output Clock Divided By *(1)*
4	1010	2
4	0011	4
8	10101010	2
8	00110011	4
8	11000011	8
10	1010101010	2
10	1110000011	10

Note to Table 5–2:
(1) This value is usually referred to as *B*.

Center-Aligned Transmitter Clock Output

A negative-edge-triggered D flipflop (DFF) register is located between the serial register of each data channel and its output buffer, as show in Figure 5–7. The negative-edge-triggered DFF register is used when center-aligned data is required. For center alignment, the DFF only shifts the output from the channel used as the transmitter clock out. The transmitter data channels bypass the negative-edge DFF. When you use the DFF register, the data is transmitted at the negative edge of the multiplied clock. This delays the transmitted clock output relative to the data channels by half the multiplied clock cycle. This is used for HyperTransport technology, but can also be used for any interface requiring center alignment.

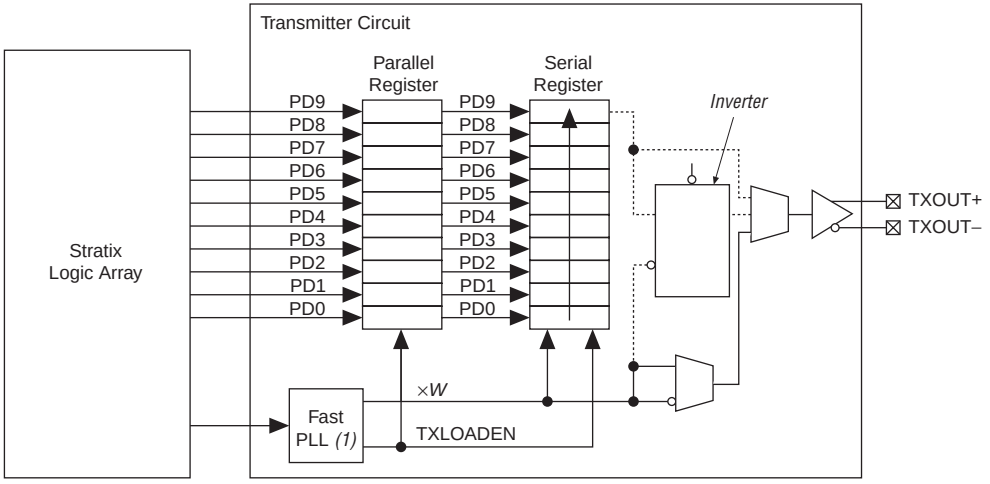
Figure 5–7. Stratix Programmable Transmitter Clock



SDR Transmitter Clock Output

You can route the high-frequency clock internally generated by the PLL out as a transmitter clock output on any of the differential channels. The high-frequency clock output allows Stratix devices to support applications that require a 1-to-1 relationship between the clock and data. The path of the high-speed clock is shown in Figure 5–8. A programmable inverter allows you to drive the signal out on either the negative edge of the clock or 180° out of phase with the streaming data.

Figure 5–8. High-Speed 1-to-1 Transmitter Clock Output



Note to Figure 5–8:
(1) This figure does not show additional circuitry for clock or data manipulation.

Using SERDES to Implement DDR

Some designs require a 2-to-1 data-to-clock ratio. These systems are usually based on Rapid I/O, SPI-4 Phase 2 (POS_PHY Level 4), or HyperTransport interfaces, and support various data rates. Stratix devices meet this requirement for such applications by providing a variable clock division factor. The SERDES clock division factor is set to 2 for double data rate (DDR).

An additional differential channel (as described in "Transmitter Clock Output" on page 5–10) is automatically configured to produce the transmitter clock output signal with half the frequency of the data.

For example, when a system is required to transmit 6.4 Gbps with a 2-to-1 clock-to-data ratio, program the SERDES with eight high-speed channels running at 800 Mbps each. When you set the output clock division factor (2 for this example), the Quartus II software automatically assigns a ninth channel as the transmitter clock output. You can edge- or center-align the transmitter clock by selecting the default PLL phase or selecting the negative-edge transmitter clock output. On the receiver side, the clock signal is connected to the receiver PLL's clock.

The multiplication factor *W* is also calculated automatically. The data rate divides by the input clock frequency to calculate the *W* factor. The deserialization factor (*J*) may be 4, 7, 8, or 10.

Figure 5–9 shows a DDR clock-to-data timing relationship with the clock center-aligned with respect to data. Figure 5–10 shows the connection between the receiver and transmitter circuits.

Figure 5–9. DDR Clock-to-Data Relationship

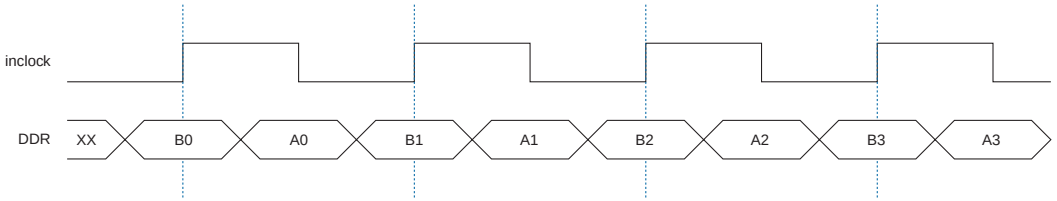
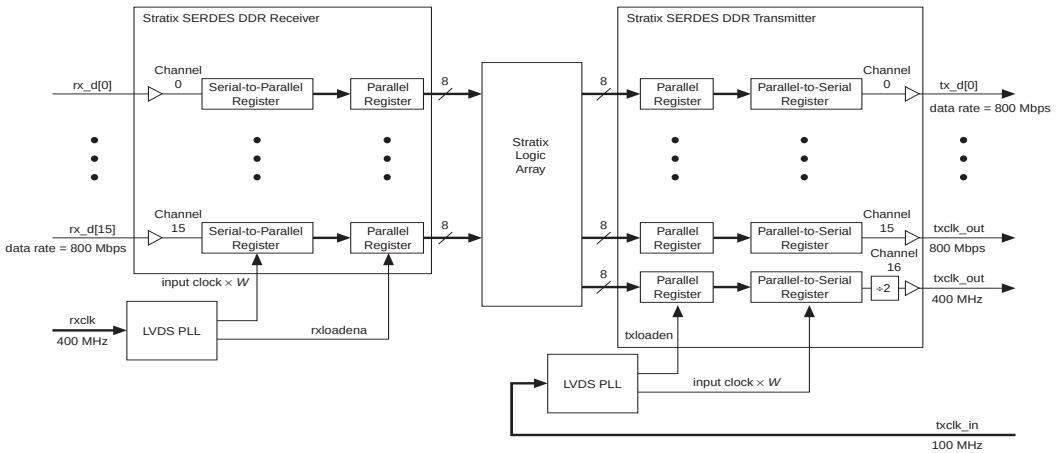


Figure 5–10. DDR Receiver & Transmitter Circuit Connection



Using SERDES to Implement SDR

Stratix devices support systems based on single data rate (SDR) operations applications by allowing you to directly transmit out the multiplied clock (as described in "SDR Transmitter Clock Output" on page 5–12). These systems are usually based on Utopia-4, SFI-4, or XSBI interfaces, and support various data rates.

An additional differential channel is automatically configured to produce the transmitter clock output signal and is transmitted along with the data.

For example, when a system is required to transmit 10 Gbps with a 1-to-1 clock-to-data ratio, program the SERDES with sixteen high-speed channels running at 624 Mbps each. The Quartus II software

automatically assigns a seventeenth channel as the transmitter clock output. You can edge- or center-align the transmitter clock output by selecting the default PLL phase or selecting the 90° phase of the PLL output. On the receiver side, the clock signal is connected to the receiver PLL's clock input, and you can assign identical clock-to-data alignment.

The multiplication factor *W* is calculated automatically. The data rate is dividing by the input clock frequency to calculate the *W* factor. The deserialization factor *J* may be 4, 7, 8, or 10.

Figure 5–11 shows an SDR clock-to-data timing relationship, with clock center aligned with respect to data. Figure 5–12 shows the connection between the receiver and transmitter circuits.

Figure 5–11. SDR Clock-to-Data Relationship

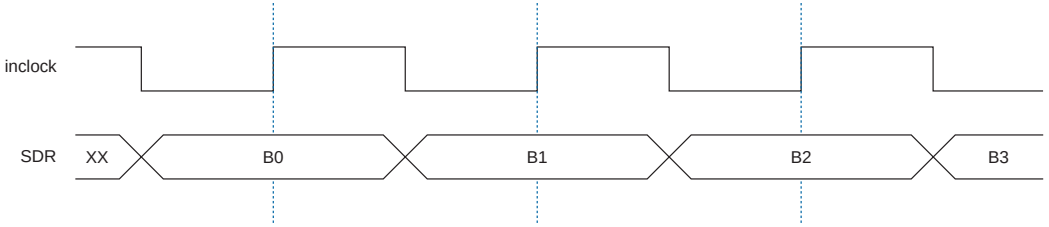
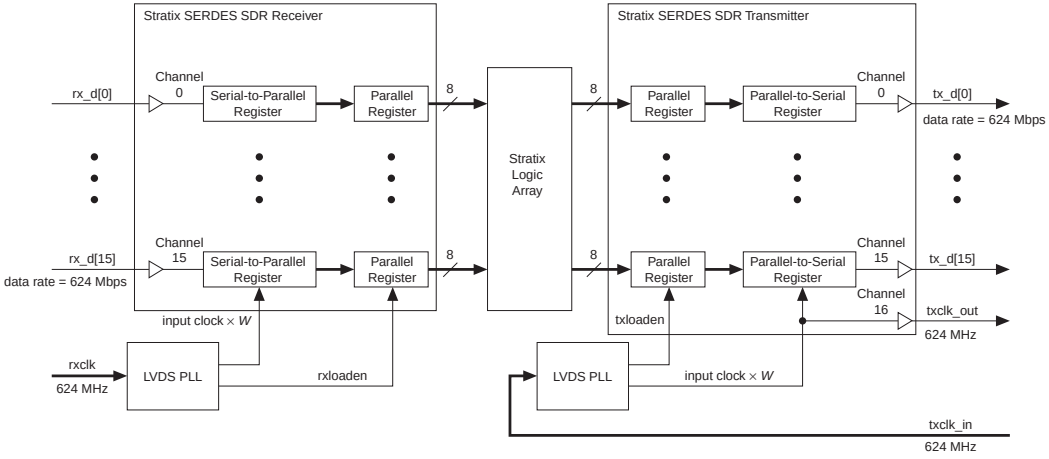


Figure 5–12. SDR Receiver & Transmitter Circuit Connection



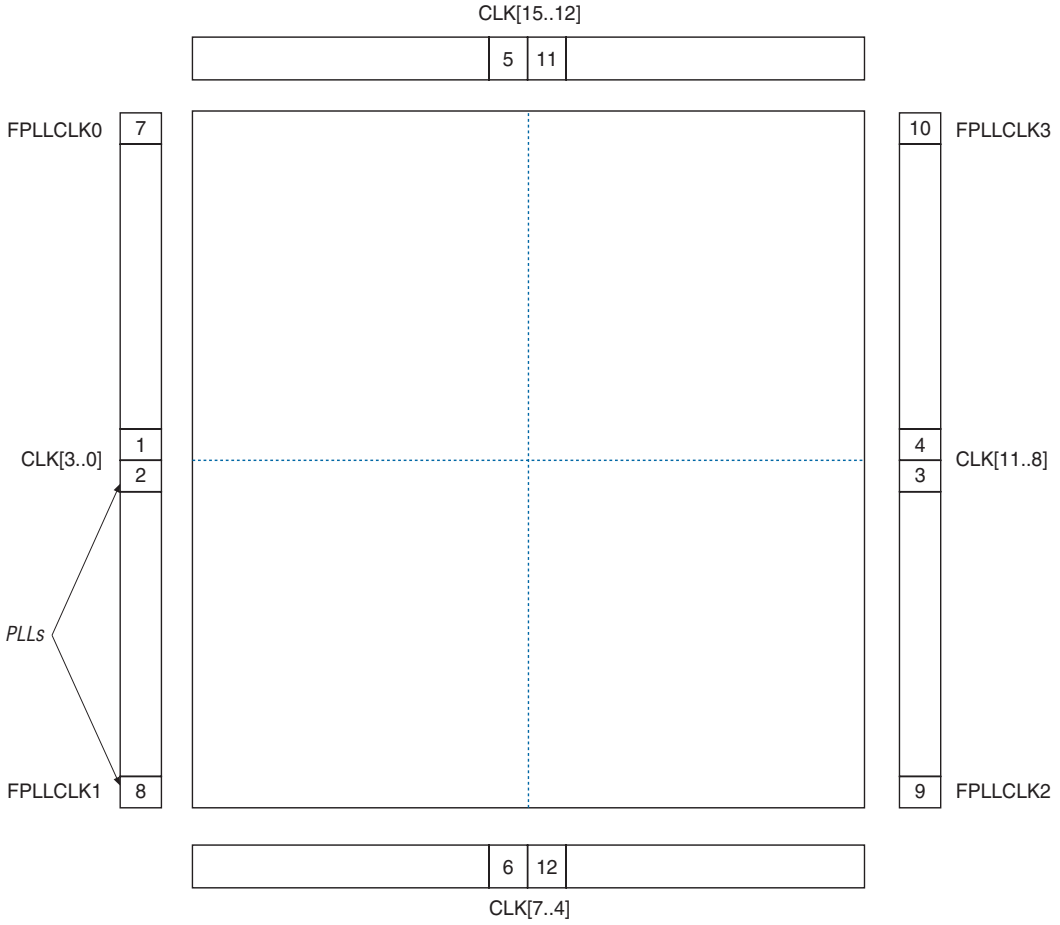
Differential I/O Interface & Fast PLLs

Stratix devices provide 16 dedicated global clocks, 8 dedicated fast regional I/O pins, and up to 16 regional clocks (four per device quadrant) that are fed from the dedicated global clock pins or PLL outputs. The 16 dedicated global clocks are driven either by global clock input pins that support all I/O standards or from enhanced and fast PLL outputs.

Stratix devices use the fast PLLs to implement clock multiplication and division to support the SERDES circuitry. The input clock is either multiplied by the *W* feedback factor and/or divided by the *J* factor. The resulting clocks are distributed to SERDES, local, or global clock lines.

Fast PLLs are placed in the center of the left and right sides for EP1S10 to EP1S25 devices. For EP1S30 to EP1S80 devices, fast PLLs are placed in the center of the left and right sides, as well as the device corners (see Figure 5–13). These fast PLLs drive a dedicated clock network to the SERDES in the rows above and below or top and bottom of the device as shown in Figure 5–13.

Figure 5–13. Stratix Fast PLL Positions & Clock Naming Convention Note (1)



Notes to Figure 5–13:

(1) Dedicated clock input pins on the right and left sides do not support PCI or PCI-X 1.0.

(2) PLLs 7, 8, 9, and 10 are not available on the EP1S30 device in the 780-pin FineLine BGA® package.

Clock Input & Fast PLL Output Relationship

Table 5–3 summarizes the PLL interface to the input clocks and the enable signal (ENA). Table 5–4 summarizes the clock networks each fast PLL can connect to across all Stratix family devices.

Table 5–3. Fast PLL Clock Inputs (Including Feedback Clocks) & Enables Note (1)

Input Pin	All Stratix Devices				EP1S30 to EP1S80 Devices Only			
	PLL 1	PLL 2	PLL 3	PLL 4	PLL 7	PLL 8	PLL 9	PLL 10
CLK0 (2)	✓				✓ (3)			
CLK1	✓							
CLK2 (2)		✓				✓ (3)		
CLK3		✓						
CLK4								
CLK5								
CLK6								
CLK7								
CLK8			✓				✓ (3)	
CLK9 (2)			✓					
CLK10				✓				✓ (3)
CLK11 (2)				✓				
CLK12								
CLK13								
CLK14								
CLK15								
ENA	✓	✓	✓	✓	✓	✓	✓	✓
FPLL7CLK					✓			
FPLL8CLK						✓		
FPLL9CLK							✓	
FPLL10CLK								✓

Notes to Table 5–3:

(1) PLLs 5, 6, 11, and 12 are not fast PLLs.

(2) Clock pins CLK0, CLK2, CLK9, CLK11, and pins FPLL[7..10]CLK do not support differential on-chip termination.

(3) Either a FPLLCLK pin or a CLK pin can drive the corner fast PLLs (PLL7, PLL8, PLL9, and PLL10) when used for general purpose. CLK pins cannot drive these fast PLLs in high-speed differential I/O mode.

Table 5–4. Fast PLL Relationship with Stratix Clock Networks (Part 1 of 2) *Notes (1), (2)*

Output Signal	All Stratix Devices				EP1S30 to EP1S80 Devices Only			
	PLL 1	PLL 2	PLL 3	PLL 4	PLL 7	PLL 8	PLL 9	PLL 10
GCLK0	✓							
GCLK1	✓							
GCLK2		✓						
GCLK3		✓						
GCLK4			✓					
GCLK9			✓					
GCLK10				✓				
GCLK11				✓				
RCLK1	✓	✓			✓			
RCLK2	✓	✓			✓			
RCLK3	✓	✓				✓		
RCLK4	✓	✓				✓		
RCLK9			✓	✓			✓	
RCLK10			✓	✓			✓	
RCLK11			✓	✓				✓
RCLK12			✓	✓				✓
DIFFIOCLK1	✓							
DIFFIOCLK2	✓							
DIFFIOCLK3		✓						
DIFFIOCLK4		✓						
DIFFIOCLK5			✓					
DIFFIOCLK6			✓					
DIFFIOCLK7				✓				
DIFFIOCLK8				✓				
DIFFIOCLK9					✓			
DIFFIOCLK10					✓			
DIFFIOCLK11						✓		
DIFFIOCLK12						✓		
DIFFIOCLK13							✓	

Table 5–4. Fast PLL Relationship with Stratix Clock Networks (Part 2 of 2) Notes (1), (2)

Output Signal	All Stratix Devices				EP1S30 to EP1S80 Devices Only			
	PLL 1	PLL 2	PLL 3	PLL 4	PLL 7	PLL 8	PLL 9	PLL 10
DIFFIOCLK14							✓	
DIFFIOCLK15								✓
DIFFIOCLK16								✓

Notes to Table 5–4:

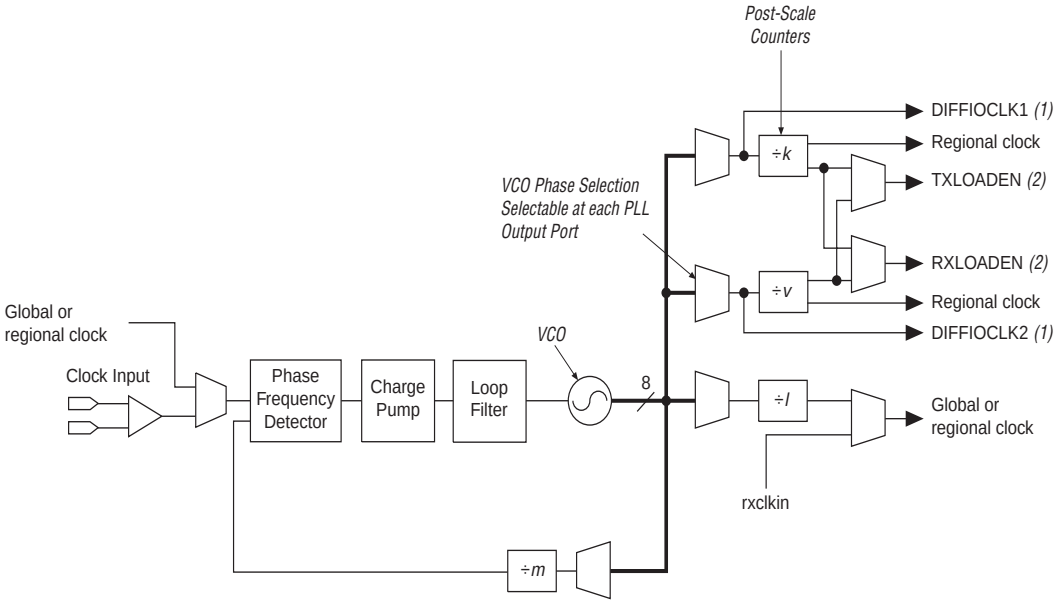
(1) PLLs 5, 6, 11, and 12 are not fast PLLs.

(2) The input clock for PLLs used to clock receiver the `rx_inclock` port on the `altlvds_rx` megafunction must be driven by a dedicated clock pin (`CLK[3..0]` and `CLK[8..11]`) or the corner pins that clock the corner PLLs (`FPLL[10..7]CLK`).

Fast PLL Specifications

You can drive the fast PLLs by an external pin or any one of the sectional clocks [21..0]. You can connect the clock input directly to local or global clock lines, as shown in Figure 5–14. You cannot use the sectional-clock inputs to the fast PLL's input multiplexer for the receiver PLL. You can only use the sectional clock inputs in the transmitter only mode or as a general purpose PLL.

Figure 5–14. Fast PLL Block Diagram



Notes to Figure 5–14:

(1) In high-speed differential I/O mode, the high-speed PLL clock feeds the SERDES. Stratix devices only support one rate of data transfer per fast PLL in high-speed differential I/O mode.

(2) Control signal for high-speed differential I/O SERDES.

You can multiply the input clock by a factor of 1 to 16. The multiplied clock is used for high-speed serialization or deserialization operations. Fast PLL specifications are shown in the *Stratix Device Family Data Sheet* section of the *Stratix Device Handbook, Volume 1*. The voltage controlled oscillators (VCOs) are designed to operate within the frequency range of 300 to 840 MHz, to provide data rates of up to 840 Mbps.

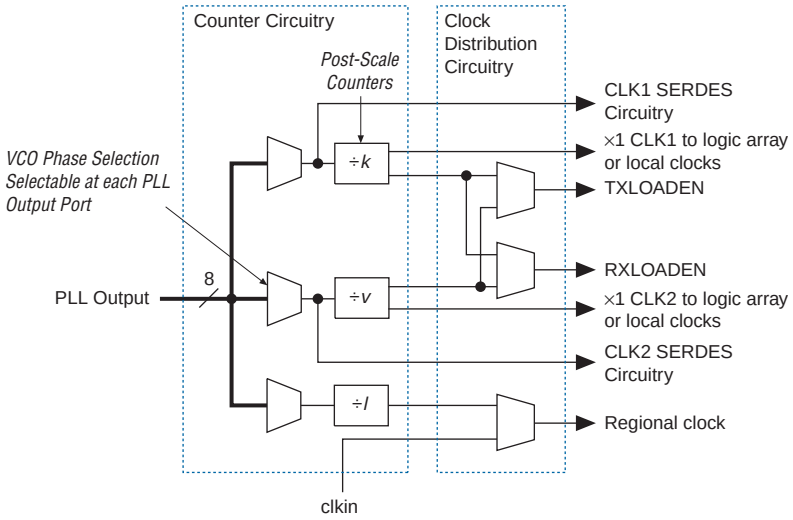
High-Speed Phase Adjust

There are eight phases of the multiplied clock at the PLL output, each delayed by 45° from the previous clock and synchronized with the original clock. The three multiplexers (shown in Figure 5–14) select one of the delayed, multiplied clocks. The PLL output drives the three counters *k*, *v*, and *l*. You can program the three individual post scale counters (*k*, *v*, and *l*) independently for division ratio or phase. The selected PLL output is used for the serialization or deserialization process in SERDES.

Counter Circuitry

The multiplied clocks bypass the counter taps *k* and *v* to directly feed the SERDES serial registers. These two taps also feed to the quadrant local clock network and the dedicated RXLOADENA or TXLOADENA pins, as shown in Figure 5–15. Both *k* and *v* are utilized simultaneously during the data-realignment procedure. When the design does not use the data realignment, both TXLOADEN and RXLOADEN pins use a single counter.

Figure 5–15. Fast PLL Connection to Logic Array



The Stratix device fast PLL has another GCLK connection for general-purpose applications. The third tap *l* feeds the quadrant local clock as well as the global clock network. You can use the *l* counter's multiplexer for applications requiring the device to connect the incoming clock directly to the local or global clocks. You can program the multiplexer to connect the RXCLKIN signal directly to the local or global clock lines. Figure 5–15 shows the connection between the incoming clock, the *l* tap, and the local or global clock lines.

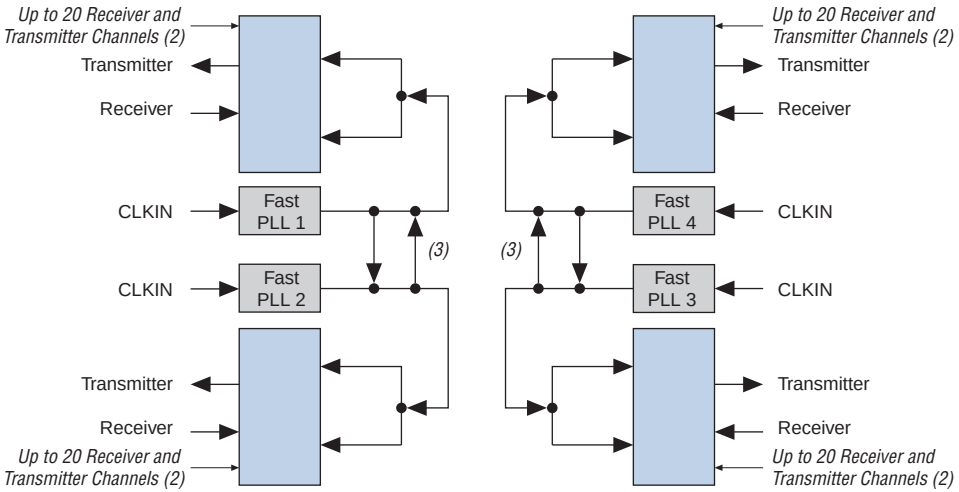
The differential clock selection is made per differential bank. Since the length of the clock tree limits the performance, each fast PLL should drive only one differential bank.

Fast PLL SERDES Channel Support

The Quartus II MegaWizard Plug-In Manager only allows you to implement up to 20 receiver or 20 transmitter channels for each fast PLL. These channels operate at up to 840 Mbps. For more information on implementing more than 20 channels, see "Fast PLLs" on page 5–52. The receiver and transmitter channels are interleaved such that each I/O bank on the left and right side of the device has one receiver channel and one transmitter channel per row. Figure 5–16 shows the fast PLL and channel layout in EP1S10, EP1S20, and EP1S25 devices. Figure 5–17 shows the fast PLL and channel layout in EP1S30 to EP1S80 devices.

For more the number of channels in each device, see Tables 5–10 through 5–14.

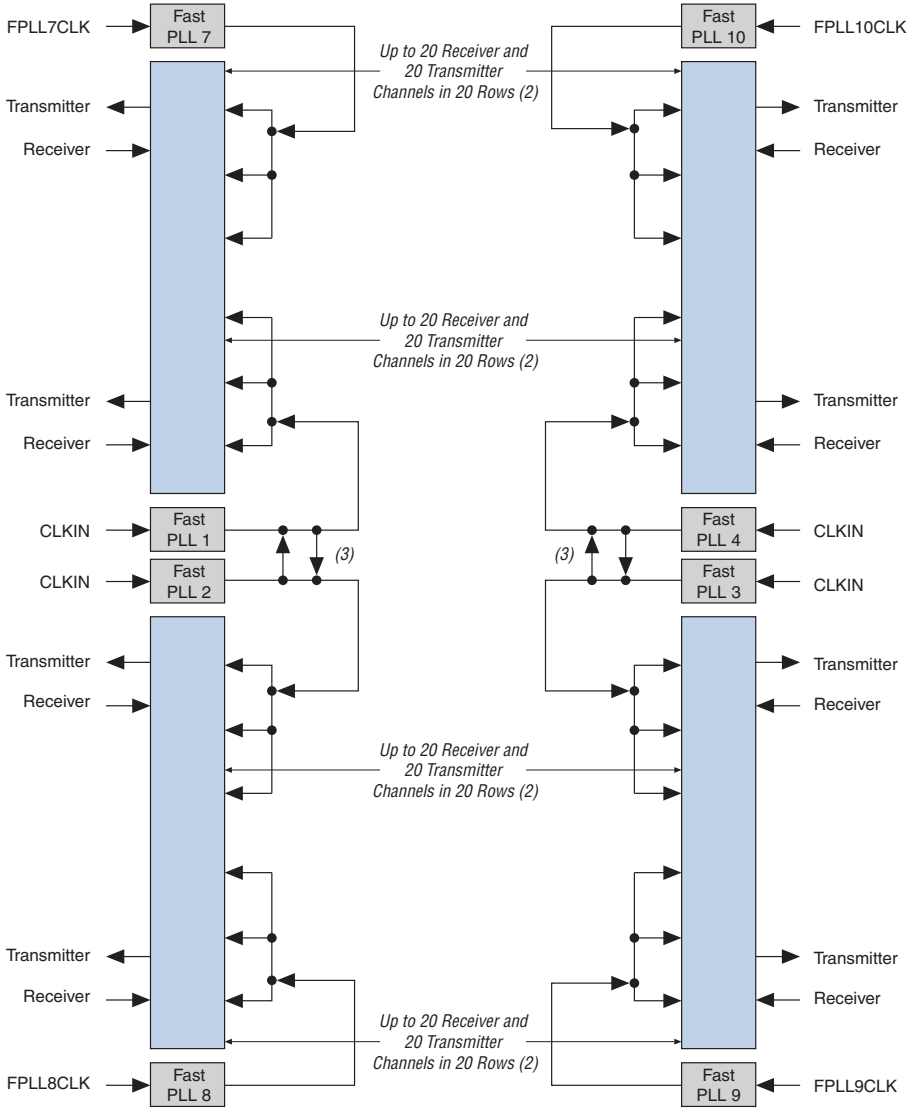
Figure 5–16. Fast PLL & Channel Layout in EP1S10, EP1S20 & EP1S25 Devices *Note (1)*



Notes to Figure 5–16:

(1) Wire-bond packages only support up to 624 Mbps until characterization shows otherwise.
(2) See Tables 5–10 through 5–14 for the exact number of channels each package and device density supports.
(3) There is a multiplexer here to select the PLL clock source. If a PLL uses this multiplexer to clock channels outside of its bank quadrant (e.g., if PLL 2 clocks PLL 1's channel region), those clocked channels support up to 840 Mbps.

Figure 5–17. Fast PLL & Channel Layout in EP1S30 to EP1S80 Devices Note (1)



Notes to Figure 5–17:

(1) Wire-bond packages only support up to 624-Mbps until characterization shows otherwise.

(2) See Tables 5–10 through 5–14 for the exact number of channels each package and device density supports.

(3) There is a multiplexer here to select the PLL clock source. If a PLL uses this multiplexer to clock channels outside of its bank quadrant (e.g., if PLL 2 clocks PLL 1's channel region), those clocked channels support up to 840 Mbps.

Advanced Clear & Enable Control

There are several control signals for clearing and enabling PLLs and their outputs. You can use these signals to control PLL resynchronization and to gate PLL output clocks for low-power applications.

The PLLENABLE pin is a dedicated pin that enables and disables Stratix device enhanced and fast PLLs. When the PLLENABLE pin is low, the clock output ports are driven by GND and all the PLLs go out of lock. When the PLLENABLE pin goes high again, the PLLs relock and resynchronize to the input clocks.

The reset signals are reset/resynchronization inputs for each enhanced PLL. Stratix devices can drive these input signals from an input pin or from LEs. When driven high, the PLL counters reset, clearing the PLL output and placing the PLL out of lock. When driven low again, the PLL resynchronizes to its input as it relocks.

Receiver Data Realignment

Most systems using serial differential I/O data transmission require a certain data-realignment circuit. Stratix devices contain embedded data-realignment circuitry. While normal I/O operation guarantees that data is captured, it does not guarantee the parallelization boundary, as this point is randomly determined based on the power-up of both communicating devices. The data-realignment circuitry corrects for bit misalignments by shifting, or delaying, data bits.

Data Realignment Principles of Operation

Stratix devices use a realignment and clock distribution circuitry (described in "Counter Circuitry" on page 5–22) for data realignment.

Set the internal rx_data_align node end high to assert the data-realignment circuitry. When this node is switched from a low to a high state, the realignment circuitry is activated and the data is delayed by one bit. To ensure the rising edge of the rx_data_align node end is latched into the PLL, the rx_data_align node end should stay high for at least two low-frequency clock cycles.

An external circuit or an internal custom-made state machine using LEs can generate the signal to pull the rx_data_align node end to a high state.

When the data realignment circuitry is activated, it generates an internal pulse Sync S1 or Sync S2 that disables one of the two counters used for the SERDES operation (described in "Counter Circuitry" on page 5–22). One counter is disabled for one high-frequency clock cycle, delaying the

RXLOADEN signal and dropping the first incoming bit of the serial input data stream located in the first serial register of the SERDES circuitry (shown in Figure 5–3 on page 5–8).

Figure 5–18 shows the function-timing diagram of a Stratix SERDES in normal ×8 mode, and Figure 5–19 shows the function-timing diagrams of a Stratix SERDES when data realignment is used.

Figure 5–18. SERDES Function Timing Diagram in Normal Operation

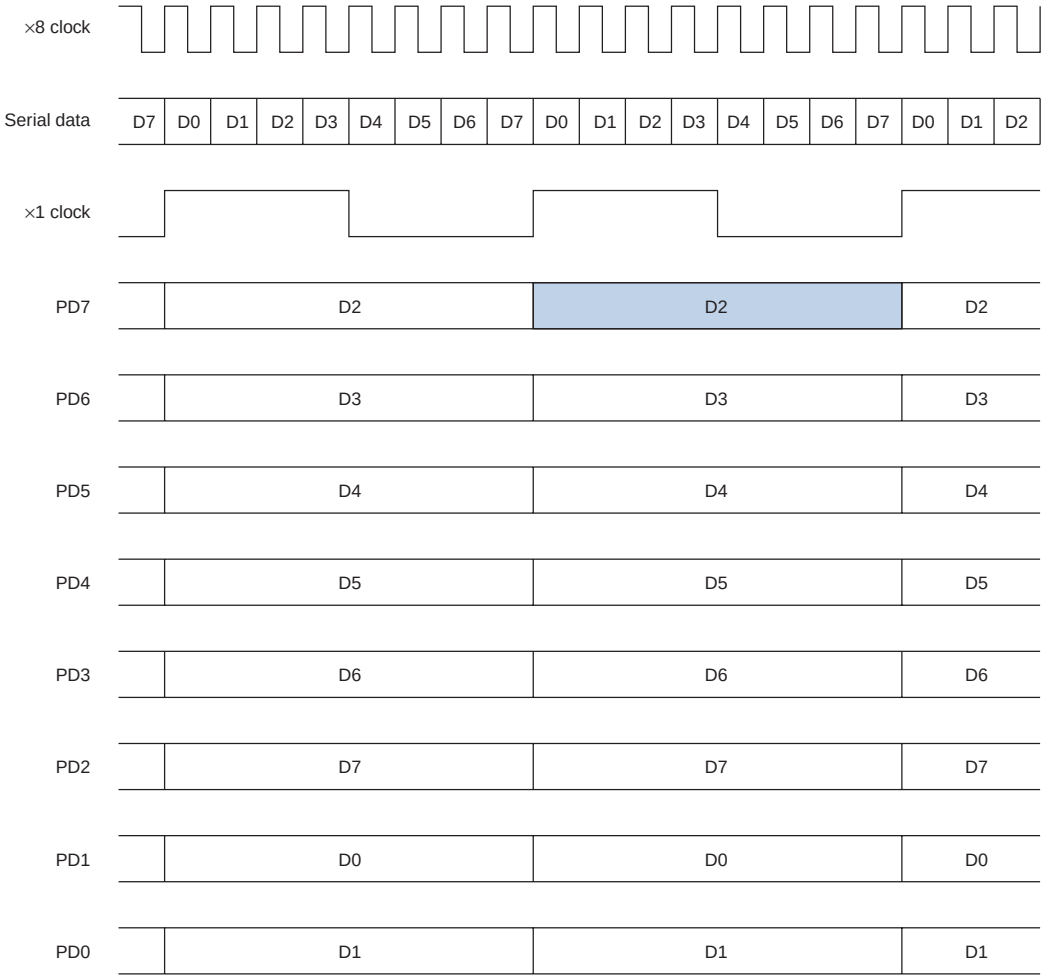
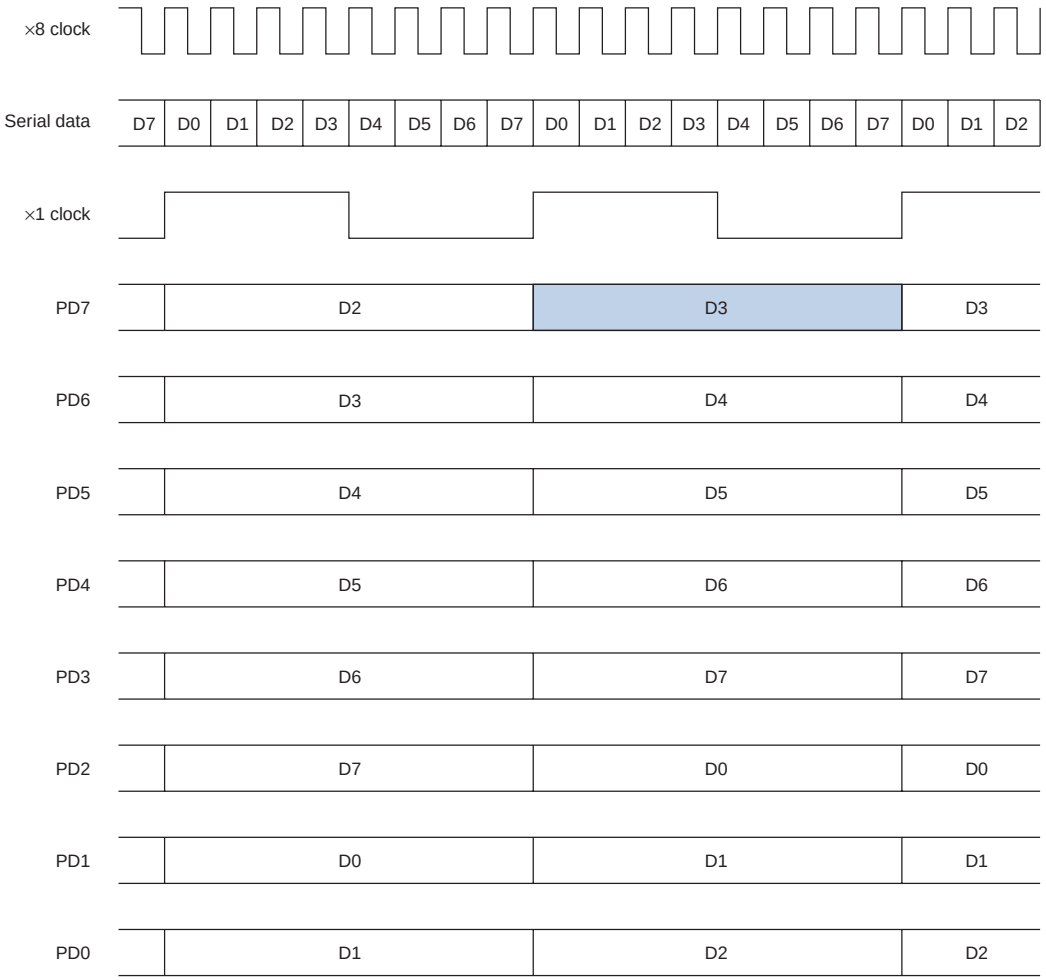


Figure 5–19. SERDES Function Timing Diagram with Data-Realignment Operation



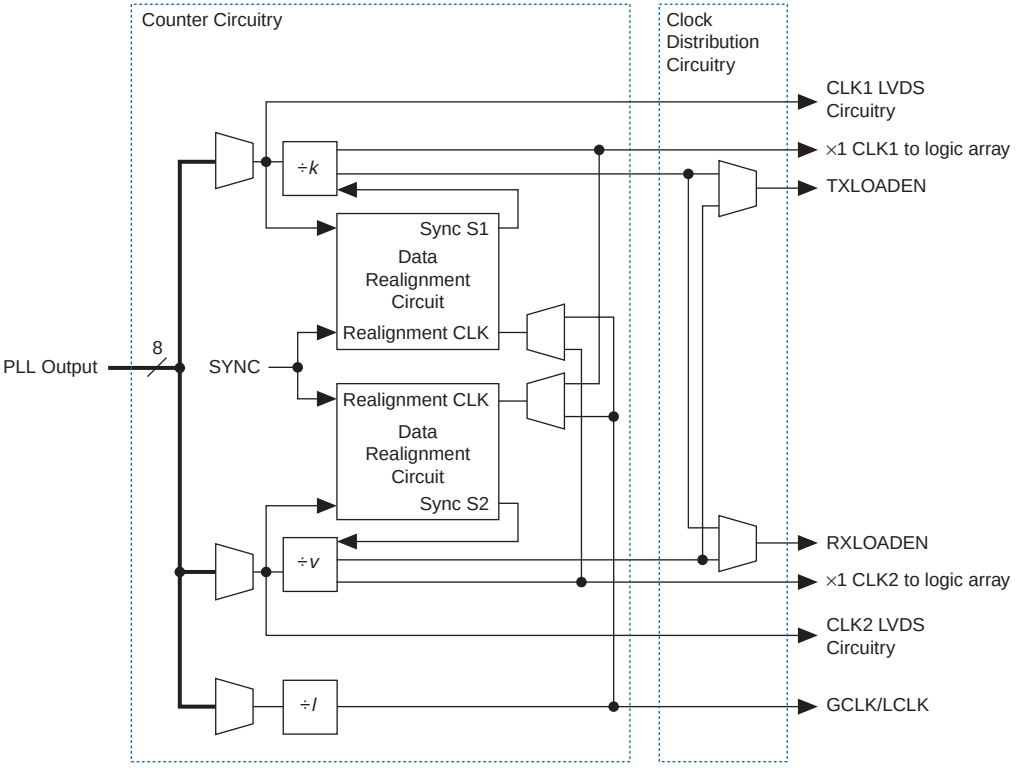
Generating TXLOADEN Signal

The TXLOADEN signal controls the transfer of data between the SERDES circuitry and the logic array when data realignment is used. To prevent the interruption of the TXLOADEN signal during data realignment, both *k* and *v* counter are used.

In normal operation the TXLOADEN signal is generated by the *k* counter. However, during the data-realignment operation this signal is generated by either counter. When the *k* counter is used for realignment, the

TXLOADEN signal is generated by the *v* counter, and when the *v* counter is used for realignment, the TXLOADEN signal is generated by the *k* counter, as shown in Figure 5–20.

Figure 5–20. Realignment Circuit TXLOADEN Signal Control *Note (1)*

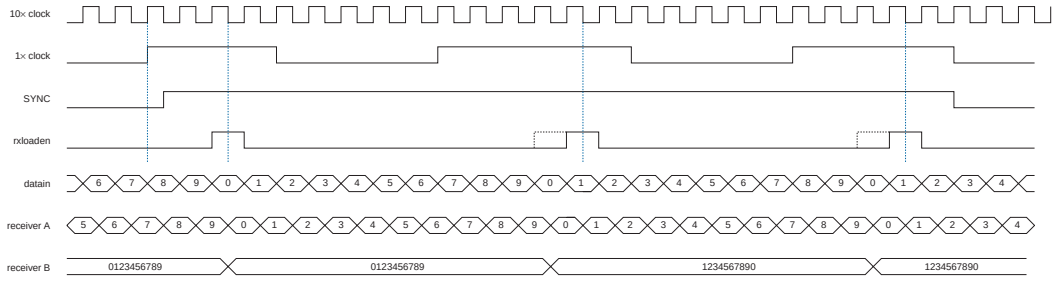


Note to Figure 5–20:
(1) This figure does not show additional realignment circuitry.

Realignment Implementation

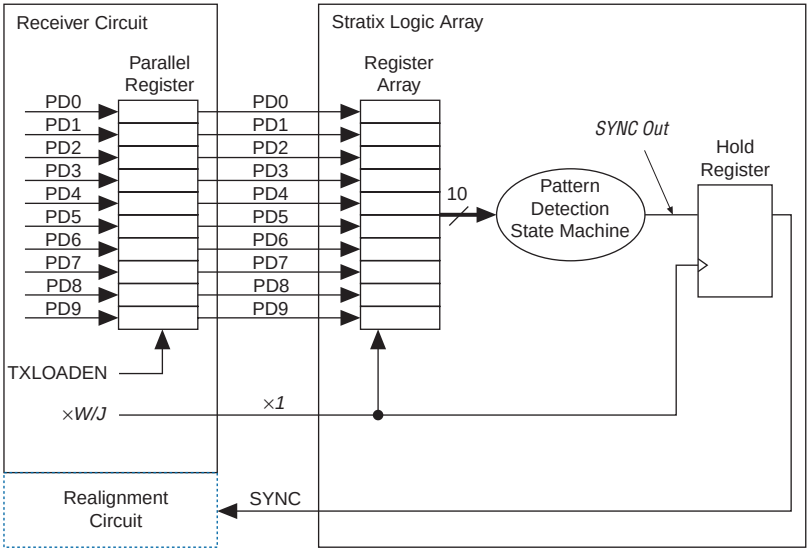
The realignment signal (SYNC) is used for data realignment and reframing. An external pin (RX_DATA_ALIGN) or an internal signal controls the rx_data_align node end. When the rx_data_align node end is asserted high for at least two low-frequency clock cycles, the RXLOADEN signal is delayed by one high-frequency clock period and the parallel bits shift by one bit. Figure 5–21 shows the timing relationship between the high-frequency clock, the RXLOADEN signal, and the parallel data.

Figure 5–21. Realignment by rx_data_align Node End



A state machine can generate the realignment signal to control the alignment procedure. Figure 5–22 shows the connection between the realignment signal and the `rx_data_align` node end.

Figure 5–22. SYNC Signal Path to Realignment Circuit



To guarantee that the `rx_data_align` signal generated by a user state machine is latched correctly by the counters, the user circuit must meet certain requirements.

- The design must include an input synchronizing register to ensure that data is synchronized to the ×1 clock.

- After the pattern detection state machine, use another synchronizing register to capture the generated SYNC signal and synchronize it to the ×1 clock.
- Since the skew in the path from the output of this synchronizing register to the PLL is undefined, the state machine must generate a pulse that is high for two ×1 clock periods.
- Since the SYNC generator circuitry only generates a single fast clock period pulse for each SYNC pulse, you cannot generate additional SYNC pulses until the comparator signal is reset low.
- To guarantee the pattern detection state machine does not incorrectly generate multiple SYNC pulses to shift a single bit, the state machine must hold the SYNC signal low for at least three ×1 clock periods between pulses.

Source-Synchronous Timing Budget

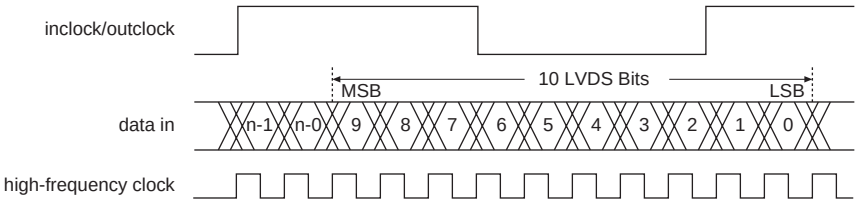
This section discusses the timing budget, waveforms, and specifications for source-synchronous signaling in Stratix devices. LVDS, LVPECL, PCML, and HyperTransport I/O standards enable high-speed data transmission. This high data-transmission rate results in better overall system performance. To take advantage of fast system performance, you must understand how to analyze timing for these high-speed signals. Timing analysis for the differential block is different from traditional synchronous timing analysis techniques.

Rather than focusing on clock-to-output and setup times, source-synchronous timing analysis is based on the skew between the data and the clock signals. High-speed differential data transmission requires you to use timing parameters provided by IC vendors and to consider board skew, cable skew, and clock jitter. This section defines the source-synchronous differential data orientation timing parameters, and timing budget definitions for Stratix devices, and explains how to use these timing parameters to determine a design's maximum performance.

Differential Data Orientation

There is a set relationship between an external clock and the incoming data. For operation at 840 Mbps and $W = 10$, the external clock is multiplied by 10 and phase-aligned by the PLL to coincide with the sampling window of each data bit. The third falling edge of high-frequency clock is used to strobe the incoming high-speed data. Therefore, the first two bits belong to the previous cycle. Figure 5–23 shows the data bit orientation of the ×10 mode as defined in the Quartus II software.

Figure 5–23. Bit Orientation in the Quartus II Software



Differential I/O Bit Position

Data synchronization is necessary for successful data transmission at high frequencies. Figure 5–24 shows the data bit orientation for a receiver channel operating in ×8 mode. Similar positioning exists for the most significant bits (MSBs) and least significant bits (LSBs) after deserialization, as listed in Table 5–5.

Figure 5–24. Bit Order for One Channel of Differential Data

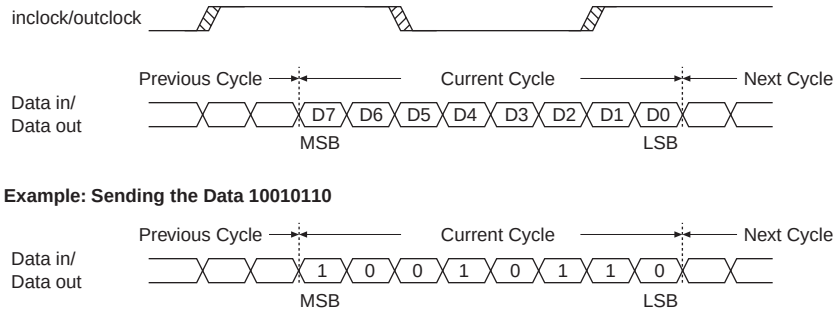


Table 5–5 shows the conventions for differential bit naming for 18 differential channels. However, the MSB and LSB are increased with the number of channels used in a system.

Table 5–5. LVDS Bit Naming

Receiver Data Channel Number	Internal 8-Bit Parallel Data	
	MSB Position	LSB Position
1	7	0
2	15	8
3	23	16
4	31	24
5	39	32
6	47	40
7	55	48
8	63	56
9	71	64
10	79	72
11	87	80
12	95	88
13	103	96
14	111	104
15	119	112
16	127	120
17	135	128
18	143	136

Timing Definition

The specifications used to define high-speed timing are described in Table 5–6.

Table 5–6. High-Speed Timing Specifications & Terminology (Part 1 of 2)

High-Speed Timing Specification	Terminology
t_C	High-speed receiver/transmitter input and output clock period.
f_{HSCLK}	High-speed receiver/transmitter input and output clock frequency.
t_{RISE}	Low-to-high transmission time.

Table 5–6. High-Speed Timing Specifications & Terminology (Part 2 of 2)

High-Speed Timing Specification	Terminology
t_{FALL}	High-to-low transmission time.
Timing unit interval (TUI)	The timing budget allowed for skew, propagation delays, and data sampling window. (TUI = 1/(Receiver Input Clock Frequency × Multiplication Factor) = t_C/w).
f_{HSDR}	Maximum LVDS data transfer rate (f_{HSDR} = 1/TUI).
Channel-to-channel skew (TCCS)	The timing difference between the fastest and slowest output edges, including t_{CO} variation and clock skew. The clock is included in the TCCS measurement.
Sampling window (SW)	The period of time during which the data must be valid in order for you to capture it correctly. The setup and hold times determine the ideal strobe position within the sampling window. SW = t_{SW} (max) – t_{SW} (min).
Input jitter (peak-to-peak)	Peak-to-peak input jitter on high-speed PLLs.
Output jitter (peak-to-peak)	Peak-to-peak output jitter on high-speed PLLs.
t_{DUTY}	Duty cycle on high-speed transmitter output clock.
t_{LOCK}	Lock time for high-speed transmitter and receiver PLLs.

Tables 5–7 and 5–8 show the high-speed I/O timing for Stratix devices

Table 5–7. High-Speed I/O Specifications for Flip-Chip Packages (Part 1 of 3) *Notes (1), (2)*

Symbol	Conditions	-5 Speed Grade			-6 Speed Grade			-7 Speed Grade			-8 Speed Grade			Unit
		Min	Typ	Max	Min	Typ	Max	Min	Typ	Max	Min	Typ	Max	
f_{HSCLK} (Clock frequency) (LVDS, LVPECL, HyperTransport technology) $f_{HSCLK} = f_{HSDR} / W$	*W* = 4 to 30	10		210	10		210	10		156	10		115.5	MHz
	W = 2 (Serdes bypass)	50		231	50		231	50		231	50		231	MHz
	W = 2 (Serdes used)	150		420	150		420	150		312	150		231	MHz
	W = 1 (Serdes bypass)	100		462	100		462	100		462	100		462	MHz
	W = 1 (Serdes used)	300		717	300		717	300		624	300		462	MHz
f_{HSDR} Device operation (LVDS, LVPECL, HyperTransport technology)	*J* = 10	300		840	300		840	300		624	300		462	Mbps
	J = 8	300		840	300		840	300		624	300		462	Mbps
	J = 7	300		840	300		840	300		624	300		462	Mbps
	J = 4	300		840	300		840	300		624	300		462	Mbps
	J = 2	100		462	100		462	100		462	100		462	Mbps
	J = 1 (LVDS and LVPECL only)	100		462	100		462	100		462	100		462	Mbps
f_{HSCLK} (Clock frequency) (PCML) $f_{HSCLK} = f_{HSDR} / W$	*W* = 4 to 30 (Serdes used)	10		100	10		100	10		77.75	10		77.75	MHz
	W = 2 (Serdes bypass)	50		200	50		200	50		150	50		150	MHz
	W = 2 (Serdes used)	150		200	150		200	150		155.5	150		155.5	MHz
	W = 1 (Serdes bypass)	100		250	100		250	100		200	100		200	MHz
	W = 1 (Serdes used)	300		400	300		400	300		311	300		311	MHz

Table 5–7. High-Speed I/O Specifications for Flip-Chip Packages (Part 2 of 3) *Notes (1), (2)*

Symbol	Conditions	-5 Speed Grade			-6 Speed Grade			-7 Speed Grade			-8 Speed Grade			Unit
		Min	Typ	Max	Min	Typ	Max	Min	Typ	Max	Min	Typ	Max	
f_{HSDR} Device operation (PCML)	J = 10	300		400	300		400	300		311	300		311	Mbps
	J = 8	300		400	300		400	300		311	300		311	Mbps
	J = 7	300		400	300		400	300		311	300		311	Mbps
	J = 4	300		400	300		400	300		311	300		311	Mbps
	J = 2	100		400	100		400	100		300	100		300	Mbps
	J = 1	100		250	100		250	100		200	100		200	Mbps
TCCS	All			200			200			300			300	ps
SW	PCML (J = 4, 7, 8, 10)	750			750			800			800			ps
	PCML (J = 2)	900			900			1,200			1,200			ps
	PCML (J = 1)	1,500			1,500			1,700			1,700			ps
	LVDS and LVPECL (J = 1)	500			500			550			550			ps
	LVDS, LVPECL, HyperTransport technology (J = 2 through 10)	440			440			500			500			ps
Input jitter tolerance (peak-to-peak)	All			250			250			250			250	ps
Output jitter (peak-to-peak)	All			160			160			200			200	ps
Output t_{RISE}	LVDS	80	110	120	80	110	120	80	110	120	80	110	120	ps
	HyperTransport technology	110	170	200	110	170	200	120	170	200	120	170	200	ps
	LVPECL	90	130	150	90	130	150	100	135	150	100	135	150	ps
	PCML	80	110	135	80	110	135	80	110	135	80	110	135	ps

Table 5–7. High-Speed I/O Specifications for Flip-Chip Packages (Part 3 of 3) Notes (1), (2)

Symbol	Conditions	-5 Speed Grade			-6 Speed Grade			-7 Speed Grade			-8 Speed Grade			Unit
		Min	Typ	Max	Min	Typ	Max	Min	Typ	Max	Min	Typ	Max	
Output t_{FALL}	LVDS	80	110	120	80	110	120	80	110	120	80	110	120	ps
	HyperTransport technology	110	170	200	110	170	200	110	170	200	110	170	200	ps
	LVPECL	90	130	160	90	130	160	100	135	160	100	135	160	ps
	PCML	105	140	175	105	140	175	110	145	175	110	145	175	ps
t_{DUTY}	LVDS (*J* = 2 through 10)	47.5	50	52.5	47.5	50	52.5	47.5	50	52.5	47.5	50	52.5	%
	LVDS (*J* =1) and LVPECL, PCML, HyperTransport technology	45	50	55	45	50	55	45	50	55	45	50	55	%
t_{LOCK}	All			100			100			100			100	µs

Notes to Table 5–7:
(1) When J = 4, 7, 8, and 10, the SERDES block is used.
(2) When J = 2 or J = 1, the SERDES is bypassed.

Table 5–8. High-Speed I/O Specifications for Wire-Bond Packages (Part 1 of 3)

Symbol	Conditions	-6 Speed Grade			-7 Speed Grade			-8 Speed Grade			Unit
		Min	Typ	Max	Min	Typ	Max	Min	Typ	Max	
f_{HSCLK} (Clock frequency) (LVDS,LVPECL, HyperTransport technology) $f_{HSCLK} = f_{HSDR} / W$	*W* = 4 to 30 (Serdes used)	10		156	10		115.5	10		115.5	MHz
	W = 2 (Serdes bypass)	50		231	50		231	50		231	MHz
	W = 2 (Serdes used)	150		312	150		231	150		231	MHz
	W = 1 (Serdes bypass)	100		311	100		270	100		270	MHz
	W = 1 (Serdes used)	300		624	300		462	300		462	MHz

Table 5–8. High-Speed I/O Specifications for Wire-Bond Packages (Part 2 of 3)

Symbol	Conditions	-6 Speed Grade			-7 Speed Grade			-8 Speed Grade			Unit
		Min	Typ	Max	Min	Typ	Max	Min	Typ	Max	
f_{HSDR} Device operation, (LVDS,LVPECL, HyperTransport technology)	J = 10	300		624	300		462	300		462	Mbps
	J = 8	300		624	300		462	300		462	Mbps
	J = 7	300		624	300		462	300		462	Mbps
	J = 4	300		624	300		462	300		462	Mbps
	J = 2	100		462	100		462	100		462	Mbps
	J = 1 (LVDS and LVPECL only)	100		311	100		270	100		270	Mbps
f_{HSCLK} (Clock frequency) (PCML) $f_{HSCLK} = f_{HSDR} / W$	*W* = 4 to 30 (Serdes used)	10		77.75							MHz
	W = 2 (Serdes bypass)	50		150	50		77.5	50		77.5	MHz
	W = 2 (Serdes used)	150		155.5							MHz
	W = 1 (Serdes bypass)	100		200	100		155	100		155	MHz
	W = 1 (Serdes used)	300		311							MHz
Device operation, f_{HSDR} (PCML)	J = 10	300		311							Mbps
	J = 8	300		311							Mbps
	J = 7	300		311							Mbps
	J = 4	300		311							Mbps
	J = 2	100		300	100		155	100		155	Mbps
	J = 1	100		200	100		155	100		155	Mbps
TCCS	All			400			400			400	ps
SW	PCML (J = 4, 7, 8, 10) only	800			800			800			ps
	PCML (J = 2) only	1,200			1,200			1,200			ps
	PCML (J = 1) only	1,700			1,700			1,700			ps
	LVDS and LVPECL (J = 1) only	550			550			550			ps
	LVDS, LVPECL, HyperTransport technology (J = 2 through 10) only	500			500			500			ps

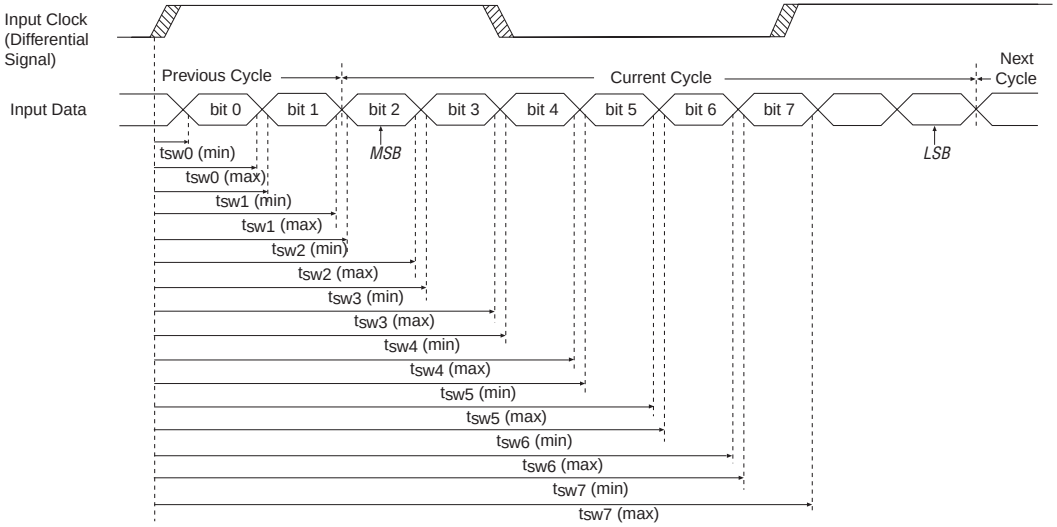
Table 5–8. High-Speed I/O Specifications for Wire-Bond Packages (Part 3 of 3)

Symbol	Conditions	-6 Speed Grade			-7 Speed Grade			-8 Speed Grade			Unit
		Min	Typ	Max	Min	Typ	Max	Min	Typ	Max	
Input jitter tolerance (peak-to-peak)	All			250			250			250	ps
Output jitter (peak-to-peak)	All			200			200			200	ps
Output t_{RISE}	LVDS	80	110	120	80	110	120	80	110	120	ps
	HyperTransport technology	120	170	200	120	170	200	120	170	200	ps
	LVPECL	100	135	150	100	135	150	100	135	150	ps
	PCML	80	110	135	80	110	135	80	110	135	ps
Output t_{FALL}	LVDS	80	110	120	80	110	120	80	110	120	ps
	HyperTransport	110	170	200	110	170	200	110	170	200	ps
	LVPECL	100	135	160	100	135	160	100	135	160	ps
	PCML	110	145	175	110	145	175	110	145	175	ps
t_{DUTY}	LVDS (J =2..10) only	47.5	50	52.5	47.5	50	52.5	47.5	50	52.5	%
	LVDS (J =1) and LVPECL, PCML, HyperTransport technology	45	50	55	45	50	55	45	50	55	%
t_{LOCK}	All			100			100			100	µs

Input Timing Waveform

Figure 5–25 illustrates the essential operations and the timing relationship between the clock cycle and the incoming serial data. For a functional description of the SERDES, see "Principles of SERDES Operation" on page 5–6.

Figure 5–25. Input Timing Waveform *Note (1)*

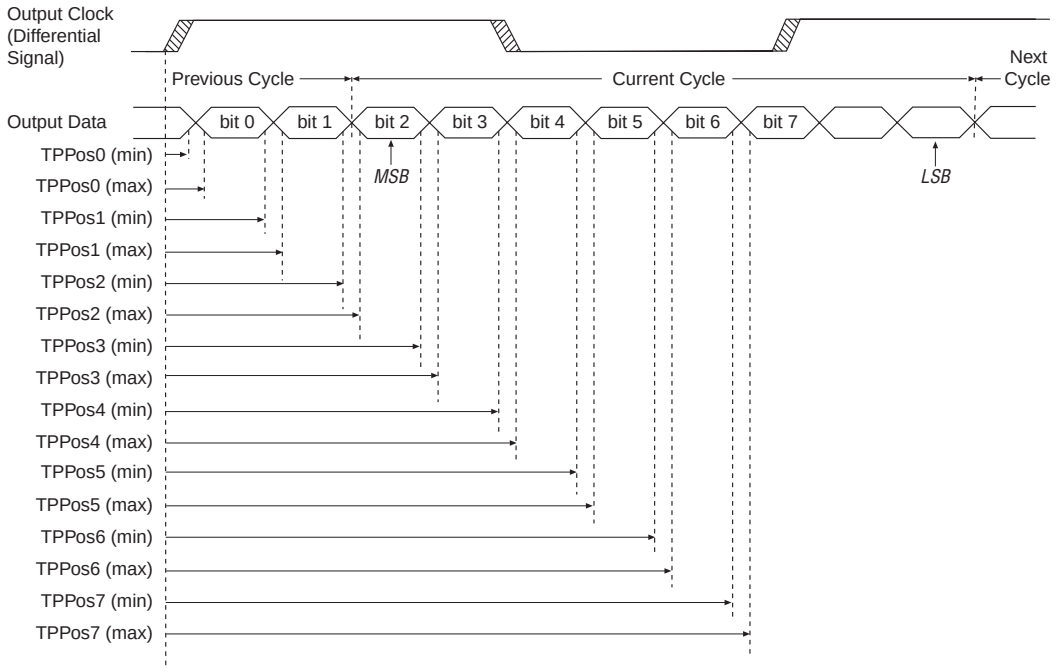


Note to Figure 5–25:
(1) The timing specifications are referenced at a 100-mV differential voltage.

Output Timing

The output timing waveform in Figure 5–26 illustrates the relationship between the output clock and the serial output data stream.

Figure 5–26. Output Timing Waveform *Note (1)*



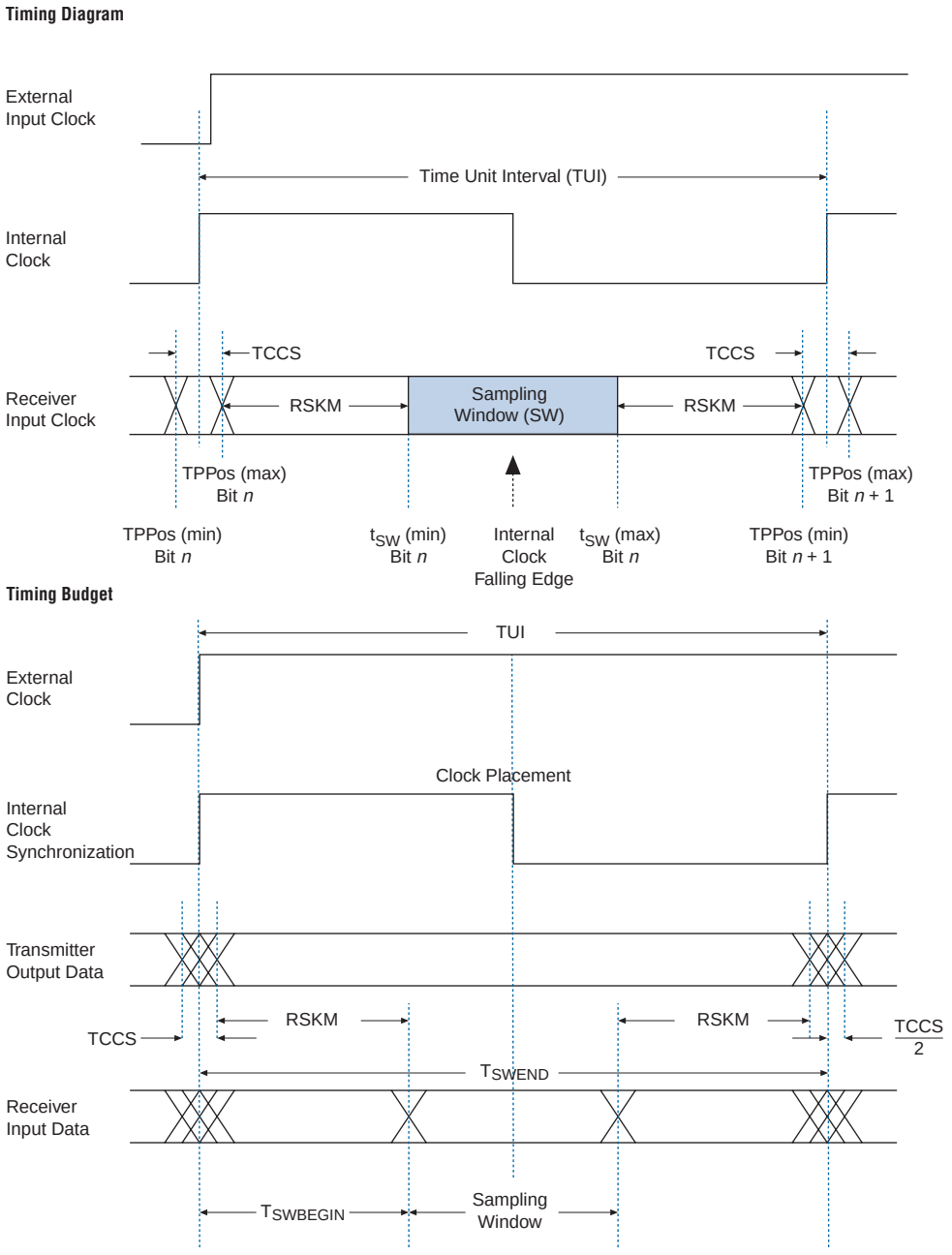
Note to Figure 5–26:

(1) The timing specifications are referenced at a 250-mV differential voltage.

Receiver Skew Margin

Change in system environment, such as temperature, media (cable, connector, or PCB) loading effect, a receiver's inherent setup and hold, and internal skew, reduces the sampling window for the receiver. The timing margin between receiver's clock input and the data input sampling window is known as RSKM. Figure 5–27 illustrates the relationship between the parameter and the receiver's sampling window.

Figure 5–27. Differential High-Speed Timing Diagram & Timing Budget



Switching Characteristics

Timing specifications for Stratix devices are listed in Tables 5–7 and 5–8. You can also find Stratix device timing information in the *Stratix Device Family Data Sheet* section of the *Stratix Device Handbook, Volume 1*.

Timing Analysis

Differential timing analysis is based on skew between data and the clock signals. For static timing analysis, the timing characteristics of the differential I/O standards are guaranteed by design and depend on the frequency at which they are operated. Use the values in the *Stratix Device Family Data Sheet* section of the *Stratix Device Handbook, Volume 1* to calculate system timing margins for various I/O protocols. For detailed descriptions and implementations of these protocols, see the Altera web site at **www.altera.com**.

SERDES Bypass DDR Differential Signaling

Each Stratix device high-speed differential I/O channel can transmit or receive data in by-two (×2) mode at up to 624 Mbps using PLLs. These pins do not require dedicated SERDES circuitry and they implement serialization and deserialization with minimal logic.

SERDES Bypass DDR Differential Interface Review

Stratix devices use dedicated DDR circuitry to implement ×2 differential signaling. Although SDR circuitry samples data only at the positive edge of the clock, DDR captures data on both the rising and falling edges for twice the transfer rate of SDR. Stratix device shift registers, internal global PLLs, and I/O cells can perform serial-to-parallel conversions on incoming data and parallel-to-serial conversion on outgoing data.

SERDES Clock Domains

The SERDES bypass differential signaling can use any of the many clock domains available in Stratix devices. These clock domains fall into four categories: global, regional, fast regional, and internally generated.

General-purpose PLLs generate the global clock domains. The fast PLLs can generate additional global clocks domains. Each PLL features two taps that directly drive two unique global clock networks. A dedicated clock pin drives each general-purpose PLL. These clock lines are utilized when designing for speeds up to 420 Mbps. Tables 5–3 and 5–4 on page 5–19, respectively, show the available clocks in Stratix devices.

SERDES Bypass DDR Differential Signaling Receiver Operation

The SERDES bypass differential signaling receiver uses the Stratix device DDR input circuitry to receive high-speed serial data. The DDR input circuitry consists of a pair of shift registers used to capture the high-speed serial data, and a latch.

One register captures the data on the positive edge of the clock (generated by PLL) and the other register captures the data on the negative edge of the clock. Because the data captured on the negative edge is delayed by one-half of the clock cycle, it is latched before it interfaces with the system logic.

Figure 5–28 shows the DDR timing relationship between the incoming serial data and the clock. In this example, the `inclock` signal is running at half the speed of the incoming data. However, other combinations are also possible. Figure 5–29 shows the DDR input and the other modules used in a Flexible-LVDS receiver design to interface with the system logic.

Figure 5–28. ×2 Timing Relation between Incoming Serial Data & Clock

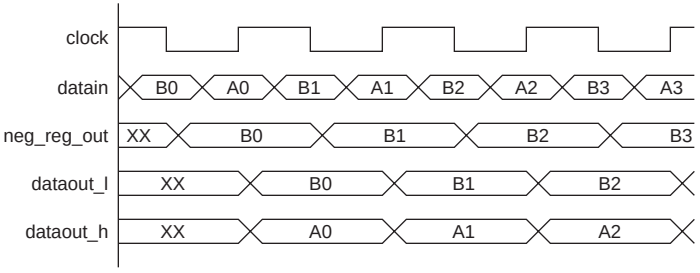
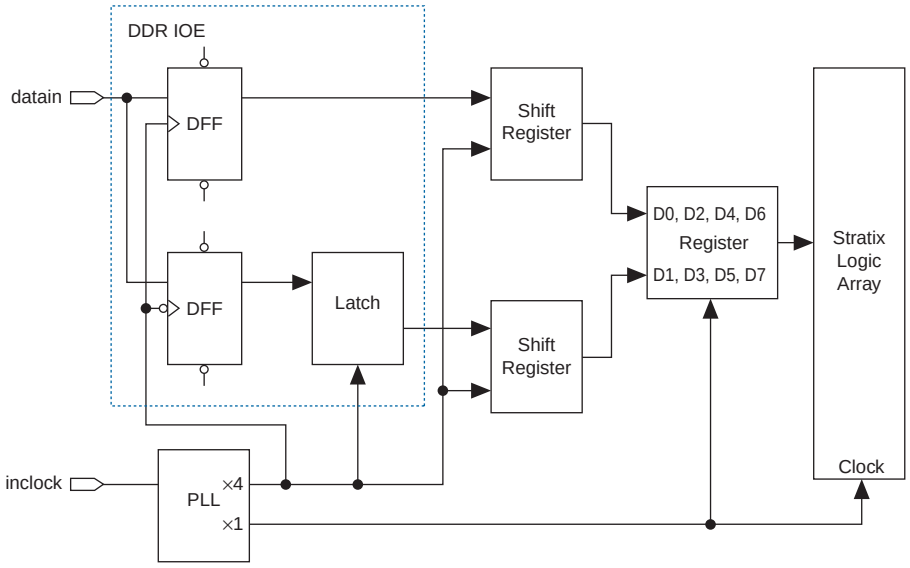


Figure 5–29. ×2 Data Rate Receiver Channel with Deserialization Factor of 8



SERDES Bypass DDR Differential Signaling Transmitter Operation

The ×2 differential signaling transmitter uses the Stratix device DDR output circuitry to transmit high-speed serial data. The DDR output circuitry consists of a pair of shift registers and a multiplexer. The shift registers capture the parallel data on the clock's rising edge (generated by the PLL), and a multiplexer transmits the data in sync with the clock. Figure 5–30 shows the DDR timing relation between the parallel data and the clock. In this example, the `inclock` signal is running at half the speed of the data. However, other combinations are possible. Figure 5–31 shows the DDR output and the other modules used in a ×2 transmitter design to interface with the system logic.

Figure 5–30. ×2 Timing Relation between Parallel Data & Clock

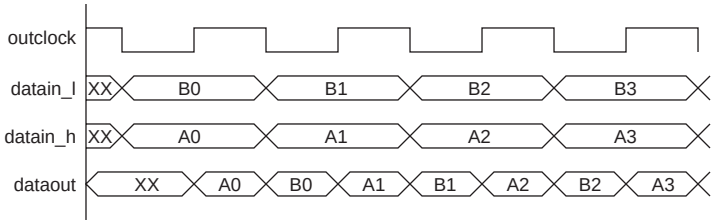
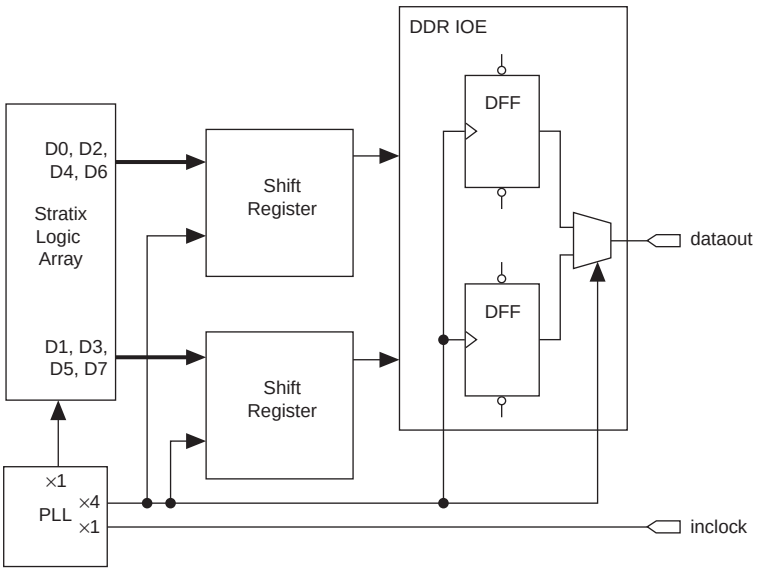


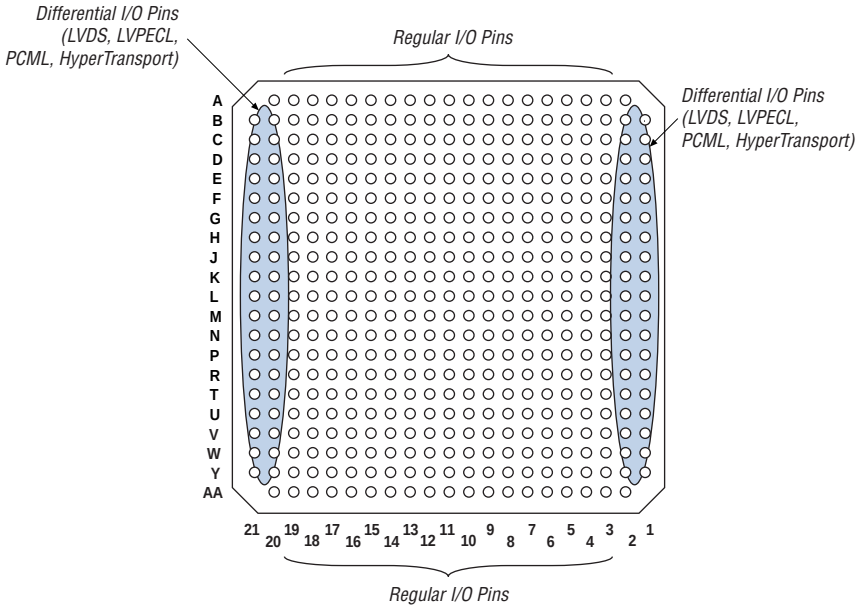
Figure 5–31. ×2 Data Rate Transmitter Channel with Serialization Factor of 8



High-Speed Interface Pin Locations

Stratix high-speed interface pins are located at the edge of the package to limit the possible mismatch between a pair of high-speed signals. Stratix devices have eight programmable I/O banks. Figure 5–32 shows the I/O pins and their location relative to the package.

Figure 5–32. Differential I/O Pin Locations



Differential I/O Termination

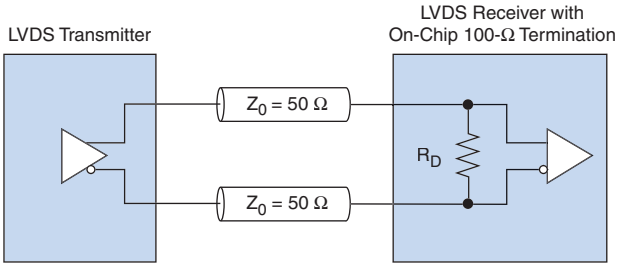
Stratix devices implement differential on-chip termination to reduce reflections and maintain signal integrity. On-chip termination also minimizes the number of external resistors required. This simplifies board design and places the resistors closer to the package, eliminating small stubs that can still lead to reflections.

R_D Differential Termination

Stratix devices support differential on-chip termination for the LVDS I/O standard. External termination is required on output pins for PCML transmitters. HyperTransport, LVPECL, and LVDS receivers require 100 ohm termination at the input pins. Figure 5–33 shows the device with differential termination for the LVDS I/O standard.

For more information on differential on-chip termination technology, see the *Selectable I/O Standards in Stratix & Stratix GX Devices* chapter.

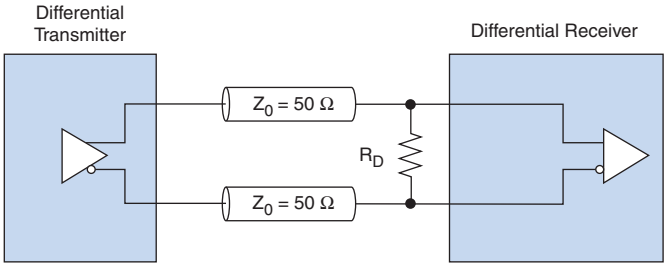
Figure 5–33. LVDS Differential On-Chip Termination



HyperTransport & LVPECL Differential Termination

HyperTransport and LVPECL I/O standards are terminated by an external 100-Ω resistor on the input pin. Figure 5–34 shows the device with differential termination for the HyperTransport or LVPECL I/O standard.

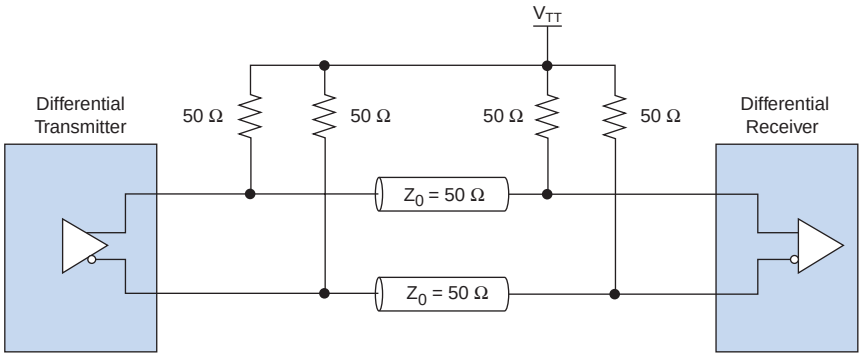
Figure 5–34. HyperTransport & LVPECL Differential Termination



PCML Differential Termination

The PCML I/O technology is an alternative to the LVDS I/O technology, and use an external voltage source (V_{TT}), a pair of 100-Ω resistors on the input side and a pair of 50-Ω resistors on the output side. Figure 5–35 shows the device with differential termination for PCML I/O standard.

Figure 5–35. PCML Differential Termination



Differential HSTL Termination

The HSTL class I and II I/O standards require a 0.75-V V_{REF} and a 0.75-V V_{TT}. Figures 5–36 and 5–37 show the device with differential termination for HSTL class I and II I/O standard.

Figure 5–36. Differential HSTL Class I Termination

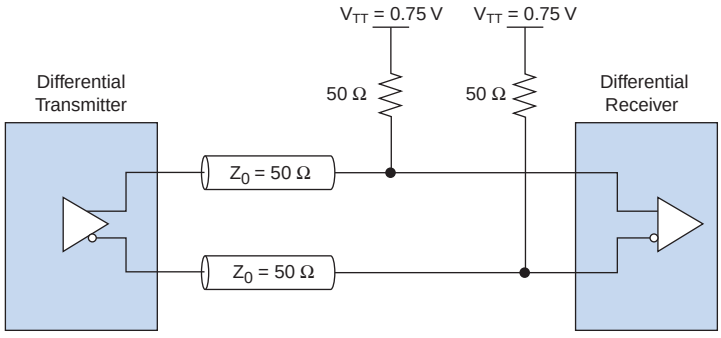
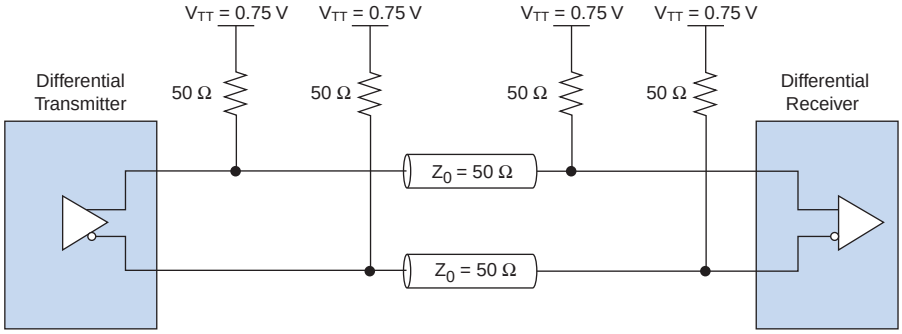


Figure 5–37. Differential HSTL Class II Termination



Differential SSTL-2 Termination

The SSTL-2 class I and II I/O standards require a 1.25-V V_{REF} and a 1.25-V V_{TT}. Figures 5–37 and 5–38 show the device with differential termination for SSTL-2 class I and II I/O standard.

Figure 5–38. Differential SSTL-2 Class I Termination

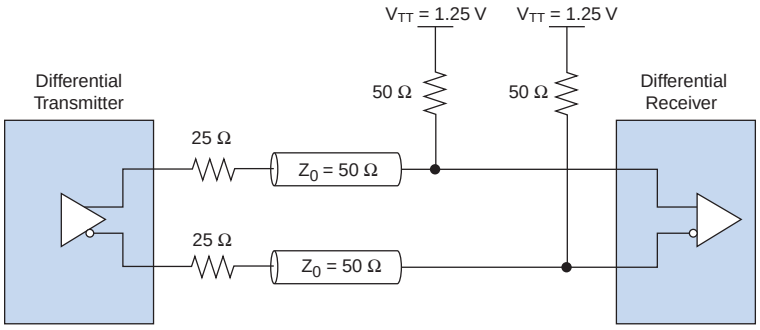
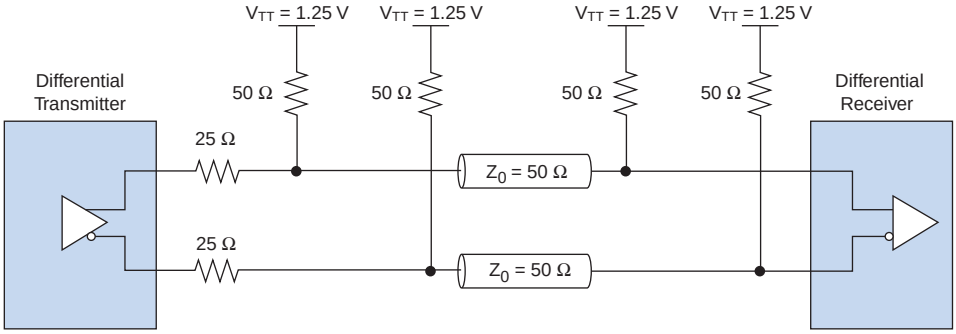


Figure 5–39. Differential SSTL-2 Class II Termination



Board Design Consideration

This section is a brief explanation of how to get the optimal performance from the Stratix high-speed I/O block and ensure first-time success in implementing a functional design with optimal signal quality. For more information on detailed board layout recommendation and I/O pin terminations see *AN 224: High-Speed Board Layout Guidelines*.

You must consider the critical issues of controlled impedance of traces and connectors, differential routing, and termination techniques to get the best performance from the IC. For more information, use this chapter and the *Stratix Device Family Data Sheet* section of the *Stratix Device Handbook, Volume 1*.

The Stratix high-speed module generates signals that travel over the media at frequencies as high as 840 Mbps. Board designers should use the following general guidelines:

- Baseboard designs on controlled differential impedance. Calculate and compare all parameters such as trace width, trace thickness, and the distance between two differential traces.
- Place external reference resistors as close to receiver input pins as possible.
- Use surface mount components.
- Avoid 90° or 45° corners.
- Use high-performance connectors such as HS-3 connectors for backplane designs. High-performance connectors are provided by Teradyne Corp **(www.teradyne.com)** or Tyco International Ltd. (**www.tyco.com**).
- Design backplane and card traces so that trace impedance matches the connector's and/or the termination's impedance.
- Keep equal number of vias for both signal traces.

- Create equal trace lengths to avoid skew between signals. Unequal trace lengths also result in misplaced crossing points and system margins as the TCCS value increases.
- Limit vias because they cause discontinuities.
- Use the common bypass capacitor values such as 0.001 µF, 0.01 µF, and 0.1 µF to decouple the fast PLL power and ground planes.
- Keep switching TTL signals away from differential signals to avoid possible noise coupling.
- Do not route TTL clock signals to areas under or above the differential signals.

Software Support

This section provides information on using the Quartus II software to create Stratix designs with LVDS transmitters or receivers. You can use the `altlvds` megafunction in the Quartus II software to implement the SERDES circuitry. You must bypass the SERDES circuitry in ×1 and ×2 mode designs and use the `altddio` megafunction to implement the deserialization instead. You can use either the logic array or the M512 RAM blocks closest to the differential pins for deserialization in SERDES bypass mode.

Differential Pins in Stratix

Stratix device differential pins are located in I/O banks 1, 2, 5, and 6 (see Figure 5–1 on page 5–2). Each bank has differential transmitter and differential receiver pin pairs. You can use each differential transmitter pin pair as either a differential data pin pair or a differential clock pin pair because Stratix devices do not have dedicated LVDS `tx_outclock` pin pairs. The differential receiver pin pairs can only function as differential data pin pairs. You can use these differential pins as regular user I/O pins when not used as differential pins. When using differential signaling in an I/O bank, you cannot place non-differential output or bidirectional pads within five I/O pads of either side of the differential pins to avoid a decrease in performance on the LVDS signals.

You only need to make assignments to the positive pin of the pin-pair. The Quartus II software automatically reserves and makes the same assignment to the negative pin. If you do not assign any differential I/O standard to the differential pins, the Quartus II software sets them as LVDS differential pins during fitting, if the design uses the SERDES circuitry. Additionally, if you bypass the SERDES circuitry, you can still use the differential pins by assigning a differential I/O standard to the pins in the Quartus II software. However, when you bypass the SERDES circuitry in the ×1 and ×2 mode, you must assign the correct differential I/O standard to the associated pins in the Assignment Organizer. For more information on how to use the Assignment Organizer, see the Quartus II On-Line Help.

Stratix devices can drive the PLL_LOCK signal to both output pins and internal logic. As a result, you do not need a dedicated LOCK pin for your PLLs. In addition, there is only one PLL_ENABLE pin that enables all the PLLs on the device, including the fast PLLs. You must use either the LVTTL or LVCMOS I/O standard with this pin.

Table 5–9 displays the LVDS pins in Stratix devices.

Table 5–9. LVDS Pin Names

Pin Names	Functions
DIFFIO_TX#p	Transmitter positive data or output clock pin
DIFFIO_TX#n	Transmitter negative data or output clock pin
DIFFIO_RX#p	Receiver positive data pin
DIFFIO_RX#n	Receiver negative data pin
FPLLCLK#p	Positive input clock pin to the corner fast PLLs *(1)*, *(2)*
FPLLCLK#n	Negative input clock pin to the corner fast PLLs *(1)*, *(2)*
CLK#p	Positive input clock pin *(2)*
CLK#n	Negative input clock pin *(2)*

Notes to Table 5–9:

(1) The FPLLCLK pin-pair is only available in EP1S30, EP1S40, EP1S60, EP1S80 devices.

(2) Either a FPLLCLK pin or a CLK pin can drive the corner fast PLLs (PLL7, PLL8, PLL9, and PLL10) when used for general purpose. CLK pins cannot drive these fast PLLs in high-speed differential I/O mode.

Fast PLLs

Each fast PLL features a multiplexed input path from a global or regional clock net. A clock pin or an output from another PLL in the device can drive the input path. The input clock for PLLs used to clock receiver the rx_inclock port on the altlvds_rx megafunction must be driven by a dedicated clock pin (CLK[3..0,8..11]) or the corner pins that clock the corner PLLs (FPLL[10..7]CLK). EP1S10, EP1S20, and EP1S25 devices have a total of four fast PLLs located in the center of both sides of the device (see Figure 5–16 on page 5–23). EP1S30 and larger devices have two additional fast PLLs per side at the top and bottom corners of the device. As shown in Figure 5–17 on page 5–24, the corner fast PLL shares an I/O bank with the closest center fast PLL (e.g., PLLs 1 and 7 share an I/O bank). The maximum input clock frequency for enhanced PLLs is 684 MHz and 717 MHz for fast PLLs.

For more information on Stratix PLLs, see the *General-Purpose PLLs in Stratix & Stratix GX Devices* chapter.

One fast PLL can drive the 20 transmitter channels and 20 receiver channels closest to it with data rates of up to 840 Mbps. Wire-bond packages support a data rate of 624 Mbps. The corner fast PLLs in EP1S80 devices support data rates of up to 840 Mbps. See Tables 5–10 through 5–14 for the number of high-speed differential channels in a particular Stratix device density and package.

Since the fast PLL drives the 20 closest differential channels, there are coverage overlaps in the EP1S30 and larger devices that have two fast PLLs per I/O bank. In these devices, either the center fast PLL or the corner fast PLL can drive the differential channels in the middle of the I/O bank.

Fast PLLs can drive more than 20 transmitter and 20 receiver channels (see Tables 5–10 through 5–14 and Figures 5–16, and 5–17 for the number of channels each PLL can drive). In addition, the center fast PLLs can drive either one I/O bank or both I/O banks on the same side (left or right) of the device, while the corner fast PLLs can only drive the differential channels in its I/O bank. Neither fast PLL can drive the differential channels in the opposite side of the device.

The center fast PLLs can only drive two I/O at 840 Mbps. For example, EP1S20 device fast PLL 1 can drive all 33 differential channels on its side (17 channels from I/O bank 2 and 16 channels from I/O bank 1) running at 840 Mbps in 4× mode. When a center fast PLL drives the opposite bank on the same side of the device, the other center fast PLL cannot drive any differential channels on the device.

See Tables 5–10 through 5–14 for the maximum number of channels that one fast PLL can drive. The number of channels is also listed in the Quartus II software. The Quartus II software gives an error message if you try to compile a design exceeding the maximum number of channels.

Additional high-speed DIFFIO pin information for Stratix devices is available in Volume 3 of the *Stratix Device Handbook*.

Table 5–10 shows the number of channels and fast PLLs in EP1S10, EP1S20, and EP1S25 devices. Tables 5–11 through 5–14 show this information for EP1S30, EP1S40, EP1S60, and EP1S80 devices.

Table 5–10. EP1S10, EP1S20 & EP1S25 Device Differential Channels (Part 1 of 2) Note (1)

Device	Package	Transmitter/ Receiver	Total Channels	Maximum Speed (Mbps)	Center Fast PLLs			
					PLL 1	PLL 2	PLL 3	PLL 4
EP1S10	484-pin FineLine BGA	Transmitter (2)	20	840	5	5	5	5
				840 (3)	10	10	10	10
		Receiver	20	840	5	5	5	5
				840 (3)	10	10	10	10
	672-pin FineLine BGA 672-pin BGA	Transmitter (2)	36	624 (4)	9	9	9	9
				624 (3)	18	18	18	18
		Receiver	36	624 (4)	9	9	9	9
				624 (3)	18	18	18	18
	780-pin FineLine BGA	Transmitter (2)	44	840	11	11	11	11
				840 (3)	22	22	22	22
		Receiver	44	840	11	11	11	11
				840 (3)	22	22	22	22
EP1S20	484-pin FineLine BGA	Transmitter (2)	24	840	6	6	6	6
				840 (3)	12	12	12	12
		Receiver	20	840	5	5	5	5
				840 (3)	10	10	10	10
	672-pin FineLine BGA 672-pin BGA	Transmitter (2)	48	624 (4)	12	12	12	12
				624 (3)	24	24	24	24
		Receiver	50	624 (4)	13	12	12	13
				624 (3)	25	25	25	25
	780-pin FineLine BGA	Transmitter (2)	66	840	17	16	16	17
				840 (3)	33	33	33	33
		Receiver	66	840	17	16	16	17
				840 (3)	33	33	33	33

Table 5–10. EP1S10, EP1S20 & EP1S25 Device Differential Channels (Part 2 of 2) Note (1)

Device	Package	Transmitter/ Receiver	Total Channels	Maximum Speed (Mbps)	Center Fast PLLs			
					PLL 1	PLL 2	PLL 3	PLL 4
EP1S25	672-pin FineLine BGA 672-pin BGA	Transmitter (2)	56	624 (4)	14	14	14	14
				624 (3)	28	28	28	28
		Receiver	58	624 (4)	14	15	15	14
				624 (3)	29	29	29	29
	780-pin FineLine BGA	Transmitter (2)	70	840	18	17	17	18
				840 (3)	35	35	35	35
		Receiver	66	840	17	16	16	17
				840 (3)	33	33	33	33
	1,020-pin FineLine BGA	Transmitter (2)	78	840	19	20	20	19
				840 (3)	39	39	39	39
		Receiver	78	840	19	20	20	19
				840 (3)	39	39	39	39

Notes to Table 5–10:

(1) The first row for each transmitter or receiver reports the number of channels driven directly by the PLL. The second row below it shows the maximum channels a PLL can drive if cross bank channels are used from the adjacent center PLL. For example, in the 484-pin FineLine BGA EP1S10 device, PLL 1 can drive a maximum of five channels at 840 Mbps or a maximum of 10 channels at 840 Mbps. The Quartus II software may also merge receiver and transmitter PLLs when a receiver is driving a transmitter. In this case, one fast PLL can drive both the maximum numbers of receiver and transmitter channels.

(2) The number of channels listed includes the transmitter clock output (`tx_outclock`) channel. If the design requires a DDR clock, it can use an extra data channel.

(3) These channels span across two I/O banks per side of the device. When a center PLL clocks channels in the opposite bank on the same side of the device it is called cross-bank PLL support. Both center PLLs can clock cross-bank channels simultaneously if, for example, PLL_1 is clocking all RX channels and PLL_2 is clocking all TX channels. You cannot have two adjacent PLLs simultaneously clocking cross-bank RX channels or two adjacent PLLs simultaneously clocking TX channels. Cross-bank allows for all receiver channels on one side of the device to be clocked on one clock while all transmitter channels on the device are clocked on the other center PLL. Crossbank PLLs are supported at full-speed, 840 Mbps. For wire-bond devices, the full-speed is 624 Mbps.

(4) These values show the channels available for each PLL without crossing another bank.

Table 5–11. EP1S30 Differential Channels Note (1)

Package	Transmitter /Receiver	Total Channels	Maximum Speed (Mbps)	Center Fast PLLs				Corner Fast PLLs (2), (3)			
				PLL1	PLL2	PLL3	PLL4	PLL7	PLL8	PLL9	PLL10
780-pin FineLine BGA	Transmitter (4)	70	840	18	17	17	18	(6)	(6)	(6)	(6)
			840 (5)	35	35	35	35	(6)	(6)	(6)	(6)
	Receiver	66	840	17	16	16	17	(6)	(6)	(6)	(6)
			840 (5)	33	33	33	33	(6)	(6)	(6)	(6)
956-pin FineLine BGA	Transmitter (4)	80 (7)	840	19	20	20	19	20	20	20	20
			840 (5)	39	39	39	39	20	20	20	20
	Receiver	80 (7)	840	20	20	20	20	19	20	20	19
			840 (5)	40	40	40	40	19	20	20	19
1,020-pin FineLine BGA	Transmitter (4)	80 (2) (7)	840	19 (1)	20	20	19 (1)	20	20	20	20
			840 (5),(8)	39 (1)	39 (1)	39 (1)	39 (1)	20	20	20	20
	Receiver	80 (2) (7)	840	20	20	20	20	19 (1)	20	20	19 (1)
			840 (5),(8)	40	40	40	40	19 (1)	20	20	19 (1)

Table 5–12. EP1S40 Differential Channels (Part 1 of 2) Note (1)

Package	Transmitter /Receiver	Total Channels	Maximum Speed (Mbps)	Center Fast PLLs				Corner Fast PLLs (2), (3)			
				PLL1	PLL2	PLL3	PLL4	PLL7	PLL8	PLL9	PLL10
780-pin FineLine BGA	Transmitter (4)	68	840	18	16	16	18	(6)	(6)	(6)	(6)
			840 (5)	34	34	34	34	(6)	(6)	(6)	(6)
	Receiver	66	840	17	16	16	17	(6)	(6)	(6)	(6)
			840 (5)	33	33	33	33	(6)	(6)	(6)	(6)
956-pin FineLine BGA	Transmitter (4)	80	840	18	17	17	18	20	20	20	20
			840 (5)	35	35	35	35	20	20	20	20
	Receiver	80	840	20	20	20	20	18	17	17	18
			840 (5)	40	40	40	40	18	17	17	18

Table 5–12. EP1S40 Differential Channels (Part 2 of 2) Note (1)

Package	Transmitter /Receiver	Total Channels	Maximum Speed (Mbps)	Center Fast PLLs				Corner Fast PLLs (2), (3)			
				PLL1	PLL2	PLL3	PLL4	PLL7	PLL8	PLL9	PLL10
1,020-pin FineLine BGA	Transmitter (4)	80 (10) (7)	840	18 (2)	17 (3)	17 (3)	18 (2)	20	20	20	20
			840 (5), (8)	35 (5)	35 (5)	35 (5)	35 (5)	20	20	20	20
	Receiver	80 (10) (7)	840	20	20	20	20	18 (2)	17 (3)	17 (3)	18 (2)
			840 (5), (8)	40	40	40	40	18 (2)	17 (3)	17 (3)	18 (2)
1,508-pin FineLine BGA	Transmitter (4)	80 (10) (7)	840	18 (2)	17 (3)	17 (3)	18 (2)	20	20	20	20
			840 (5), (8)	35 (5)	35 (5)	35 (5)	35 (5)	20	20	20	20
	Receiver	80 (10) (7)	840	20	20	20	20	18 (2)	17 (3)	17 (3)	18 (2)
			840 (5), (8)	40	40	40	40	18 (2)	17 (3)	17 (3)	18 (2)

Table 5–13. EP1S60 Differential Channels (Part 1 of 2) Note (1)

Package	Transmitter /Receiver	Total Channels	Maximum Speed (Mbps)	Center Fast PLLs				Corner Fast PLLs (2), (3)			
				PLL1	PLL2	PLL3	PLL4	PLL7	PLL8	PLL9	PLL10
956-pin FineLine BGA	Transmitter (4)	80	840	12	10	10	12	20	20	20	20
			840 (5), (8)	22	22	22	22	20	20	20	20
	Receiver	80	840	20	20	20	20	12	10	10	12
			840 (5), (8)	40	40	40	40	12	10	10	12
1,020-pin FineLine BGA	Transmitter (4)	80 (12) (7)	840	12 (2)	10 (4)	10 (4)	12 (2)	20	20	20	20
			840 (5), (8)	22 (6)	22 (6)	22 (6)	22 (6)	20	20	20	20
	Receiver	80 (10) (7)	840	20	20	20	20	12 (8)	10 (10)	10 (10)	12 (8)
			840 (5), (8)	40	40	40	40	12 (8)	10 (10)	10 (10)	12 (8)

Table 5–13. EP1S60 Differential Channels (Part 2 of 2) *Note (1)*

Package	Transmitter /Receiver	Total Channels	Maximum Speed (Mbps)	Center Fast PLLs				Corner Fast PLLs *(2), (3)*			
				PLL1	PLL2	PLL3	PLL4	PLL7	PLL8	PLL9	PLL10
1,508-pin FineLine BGA	Transmitter *(4)*	80 (36) *(7)*	840	12 (8)	10 (10)	10 (10)	12 (8)	20	20	20	20
			840 *(5),(8)*	22 (18)	22 (18)	22 (18)	22 (18)	20	20	20	20
	Receiver	80 (36) *(7)*	840	20	20	20	20	12 (8)	10 (10)	10 (10)	12 (8)
			840 *(5),(8)*	40	40	40	40	12 (8)	10 (10)	10 (10)	12 (8)

Table 5–14. EP1S80 Differential Channels (Part 1 of 2) *Note (1)*

Package	Transmitter /Receiver	Total Channels	Maximum Speed (Mbps)	Center Fast PLLs				Corner Fast PLLs *(2)*			
				PLL1	PLL2	PLL3	PLL4	PLL7	PLL8	PLL9	PLL10
956-pin FineLine BGA	Transmitter *(4)*	80 (40) *(7)*	840	10	10	10	10	20	20	20	20
			840 *(5)*,*(8)*	20	20	20	20	20	20	20	20
	Receiver	80	840	20	20	20	20	10	10	10	10
			840 *(5)*,*(8)*	40	40	40	40	10	10	10	10
1,020-pin FineLine BGA	Transmitter *(4)*	92 (12) *(7)*	840	10 (2)	10 (4)	10 (4)	10 (2)	20	20	20	20
			840 *(5)*,*(8)*	20 (6)	20 (6)	20 (6)	20 (6)	20	20	20	20
	Receiver	90 (10) *(7)*	840	20	20	20	20	10 (2)	10 (3)	10 (3)	10 (2)
			840 *(5)*,*(8)*	40	40	40	40	10 (2)	10 (3)	10 (3)	10 (2)
1,508-pin FineLine BGA	Transmitter *(4)*	80 (72) *(7)*	840	10 (10)	10 (10)	10 (10)	10 (10)	20 (8)	20 (8)	20 (8)	20 (8)
			840 *(5)*,*(8)*	20 (20)	20 (20)	20 (20)	20 (20)	20 (8)	20 (8)	20 (8)	20 (8)
	Receiver	80 (56) *(7)*	840	20	20	20	20	10 (14)	10 (14)	10 (14)	10 (14)
			840 *(5)*,*(8)*	40	40	40	40	10 (14)	10 (14)	10 (14)	10 (14)

Table 5–14. EP1S80 Differential Channels (Part 2 of 2) Note (1)

Package	Transmitter /Receiver	Total Channels	Maximum Speed (Mbps)	Center Fast PLLs				Corner Fast PLLs (2)			
				PLL1	PLL2	PLL3	PLL4	PLL7	PLL8	PLL9	PLL10

Notes to Tables 5–11 through 5–14.

(1) The first row for each transmitter or receiver reports the number of channels driven directly by the PLL. The second row below it shows the maximum channels a PLL can drive if cross bank channels are used from the adjacent center PLL. For example, in the 780-pin FineLine BGA EP1S30 device, PLL 1 can drive a maximum of 18 transmitter channels at 840 Mbps or a maximum of 35 transmitter channels at 840 Mbps. The Quartus II software may also merge transmitter and receiver PLLs when a receiver is driving a transmitter. In this case, one fast PLL can drive both the maximum numbers of receiver and transmitter channels.

(2) Some of the channels accessible by the center fast PLL and the channels accessible by the corner fast PLL overlap. Therefore, the total number of channels is not the addition of the number of channels accessible by PLLs 1, 2, 3, and 4 with the number of channels accessible by PLLs 7, 8, 9, and 10. For more information on which channels overlap, see the *Fast PLL to High-Speed I/O Connections* section in the relevant device pin table available on the web (**www.altera.com**).

(3) The corner fast PLLs in this device support a data rate of 840 Mbps for channels labeled "high" speed in the device pin tables.

(4) The numbers of channels listed include the transmitter clock output (`tx_outclock`) channel. You can use an extra data channel if you need a DDR clock.

(5) These channels span across two I/O banks per side of the device. When a center PLL clocks channels in the opposite bank on the same side of the device it is called cross-bank PLL support. Both center PLLs can clock cross-bank channels simultaneously if, for example, PLL_1 is clocking all receiver channels and PLL_2 is clocking all transmitter channels. You cannot have two adjacent PLLs simultaneously clocking cross-bank receiver channels or two adjacent PLLs simultaneously clocking transmitter channels. Cross-bank allows for all receiver channels on one side of the device to be clocked on one clock while all transmitter channels on the device are clocked on the other center PLL. Crossbank PLLs are supported at full-speed, 840 Mbps. For wire-bond devices, the full-speed is 624 Mbps.

(6) PLLs 7, 8, 9, and 10 are not available in this device.

(7) The number in parentheses is the number of slow-speed channels, guaranteed to operate at up to 462 Mbps. These channels are independent of the high-speed differential channels. For the location of these channels, see the *Fast PLL to High-Speed I/O Connections* section in the relevant device pin table available on the web (**www.altera.com**).

(8) See device pin-outs channels marked "high" speed are 840 Mbps and "low" speed channels are 462 MBps.

The Quartus II software may also merge transmitter and receiver PLLs when a receiver block is driving a transmitter block if the **Use Common PLLs for Rx and Tx** option is set for both modules. The Quartus II software does not merge the PLLs in multiple transmitter-only or multiple receiver-only modules fed by the same clock.

When you span two I/O banks using cross-bank support, you can route only two loadenable signals total between the plls. When you enable `rx_data_align`, you use both `rxloadena` and `txloadena` of a PLL. That leaves no `loadena` for the second PLL.

The only way you can use the rx_data_align is if one of the following is true:

- The RX PLL is only clocking RX channels (no resources for TX)
- If all channels can fit in one I/O bank

LVDS Receiver Block

You only need to enter the input clock frequency, deserialization factor, and the input data rate to implement an LVDS receiver block. The Quartus II software then automatically sets the clock boost (*W*) factor for the receiver. In addition, you can also indicate the clock and data alignment for the receiver or add the pll_enable, rx_data_align, and rx_locked output ports. Table 5–15 explains the function of the available ports in the LVDS receiver block.

Table 5–15. LVDS Receiver Ports

Port Name	Direction	Function	Input Port Source/Output Port Destination
rx_in[number_of_channels - 1..0]	Input	Input data channel	Pin
rx_inclock	Input	Reference input clock	Pin or output from a PLL
rx_pll_enable	Input	Enables fast PLL	Pin *(1)*, *(2)*, *(3)*
rx_data_align	Input	Control for the data realignment circuitry	Pin or logic array *(1)*, *(3)*, *(4)*
rx_locked	Output	Fast PLL locked pin	Pin or logic array *(1)*, *(3)*
rx_out[Deserialization_factor * number_of_channels -1..0]	Output	De-serialized data	Logic array
rx_outclock	Output	Internal reference clock	Logic array

Notes to Table 5–15:

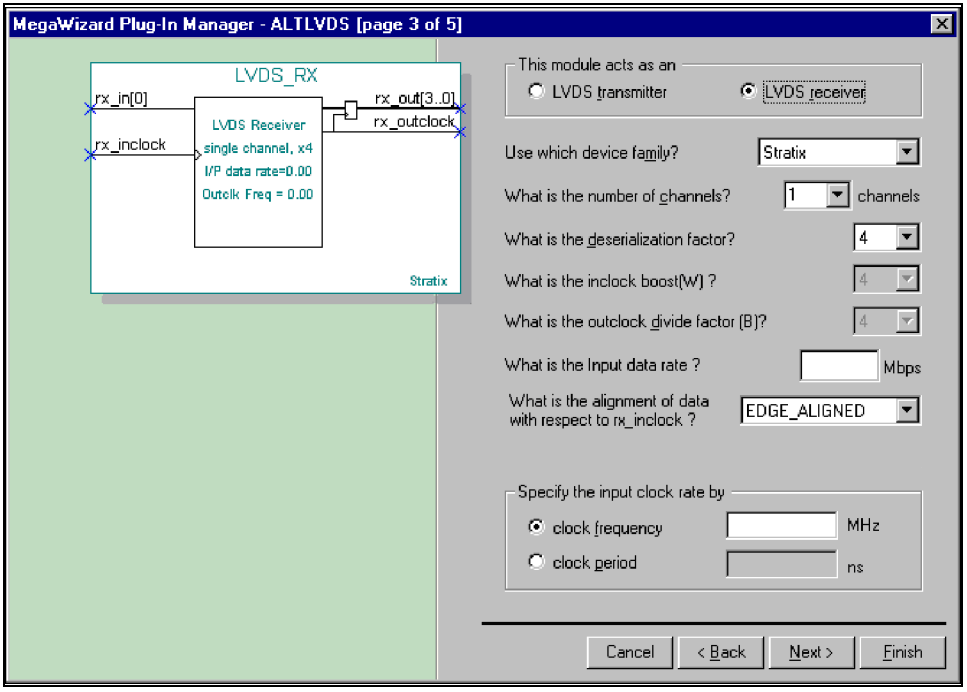
(1) This is an optional port.
(2) Only one rx_pll_enable pin is necessary to enable all the PLLs in the device.
(3) This is a non-differential pin.
(4) See "Realignment Implementation" on page 5–28 for more information. For guaranteed performance and data alignment, you must synchronize rx_data_align with rx_outclock.

Use the altlvds MegaWizard Plug-In Manager to create an LVDS receiver block. The following sections explain the parameters available in the Plug-In Manager when creating an LVDS receiver block.

Page 3 of the altlvds_rx MegaWizard Plug-In Manager

On page 3 of the `altlvds` MegaWizard Plug-In Manager, you can choose to create either an LVDS transmitter or receiver. Depending on what you select, the MegaWizard Plug-In Manager provides you with different options. Figure 5–40 shows page 3 of the `altlvds` MegaWizard Plug-In Manager with options for creating an LVDS receiver.

Figure 5–40. Page 3 of the altlvds_rx MegaWizard Plug-In Manager



Number of Channels

The **What is the number of channels?** parameter specifies the number of receiver channels required and the width of `rx_out` port. To set a fast PLL to drive over 20 channels, type the required number in the Quartus II window instead of choosing a number from the drop-down menu, which only has selections of up to 20 channels.

Deserialization Factor

Use the **What is the deserialization factor?** parameter to specify the number of bits per channel. The Stratix LVDS receiver supports 4, 7, 8, and 10 for deserialization factor (*J*) values. Based on the factor specified, the Quartus II software determines the multiplication and/or division factor for the LVDS PLL to deserialize the data.

See Table 5–5 for the differential bit naming convention. The parallel data for the n^{th} channel spans from the MSB (`rx_out` bit $[(J \times n) - 1]$) to the LSB (`rx_out` bit $[J \times (n - 1)]$), where *J* is the deserialization factor. The total width of the receiver `rx_out` port is equal to the number of channels multiplied by your deserialization factor.

Input Data Rate

The **What is the inclock boost(W)?** parameter sets the data rate coming into the receiver and is usually the deserialization factor (*J*) multiplied by the `inclock` frequency. This parameter's value must be larger than the input clock frequency and has a maximum input data rate of 840 Mbps for Stratix devices. You do not have to provide a value for the inclock boost (*W*) when designing with Stratix devices because the Quartus II software can calculate it automatically from this parameter and the clock frequency or clock period.

The `rx_outclock` frequency is $(W/J) \times$ input frequency. The parallel data coming out of the receiver has the same frequency as the `rx_outclock` port. The clock-to-data alignment of the parallel data output from the receiver depends on the **What is the alignment of data with respect to rx_inclock?** parameter.

Data Alignment with Clock

The **What is the alignment of data with respect to rx_inclock?** parameter adjusts the clock-to-data skew. For most applications, the data is source synchronous to the clock. However, there are applications where you must center-align the data with respect to the clock. You can use the **What is the alignment of data with respect to rx_inclock?** parameter to align the input data with respect to the `rx_inclock` port. The MegaWizard Plug-In automatically calculates the phase for the fast PLL outputs from the **What is the alignment of data with respect to rx_inclock?** parameter. This parameter's default value is `EDGE_ALIGNED`, and other values available from the pull-down menu are `EDGE_ALIGNED`, `CENTER_ALIGNED`, `45_DEGREES`, `135_DEGREES`, `180_DEGREES`, `225_DEGREES`, `270_DEGREES`, and `315_DEGREES`. `CENTER_ALIGNED` is the same as 90 degrees aligned and is useful for applications like HyperTransport technology.

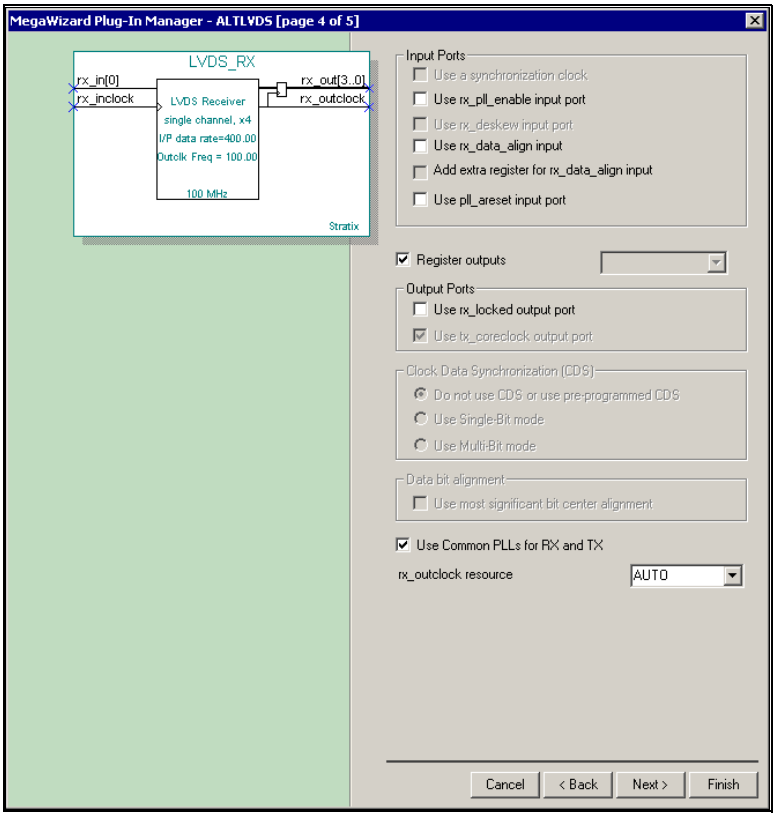
Clock Frequency or Clock Period

The fields in the **Specify the input clock rate by** box specify the input frequency or the period of the input clock going into the fast PLL. When using the same input clock to feed a transmitter and receiver simultaneously, the Quartus II software can use one fast PLL for both the transmitter and receiver.

Page 4 of the altlvds_rx MegaWizard Plug-In Manager

This section describes the parameters found on page 4 of the `altlvds_rx` MegaWizard Plug-In Manager (see Figure 5–41).

Figure 5–41. Page 4 of the altlvds_rx MegaWizard Plug-In Manager



Data Realignment
Check the **Use the "rx_data_align" input port** box within the **Input Ports** box to add the `rx_data_align` output port and enable the data realignment circuitry in Stratix SERDES. See "Receiver Data Realignment" on page 5–25 for more information. If necessary, you can create a state machine to send a pulse to the `rx_data_align` port to realign the data coming in the LVDS receiver. You need to assert the port for at least two clock cycles to enable the data realignment circuitry. Go to the Altera web site at **www.altera.com** for a sample design written in Verilog HDL.

For guaranteed performance when using data realignment, check the **Add Extra registers for rx_data_align input** box when using the `rx_data_align` port. The Quartus II software places one synchronization register in the LE closest to the `rx_data_align` port.

Register Outputs
Check the **Register outputs** box to register the receiver's output data. The register acts as the module's register boundary. If the module fed by the receiver does not have a register boundary for the data, turn this option on. The number of registers used is proportional to the deserialization factor (*J*). The Quartus II software places the synchronization registers in the LEs closest to the SERDES circuitry.

Use Common PLL for Both Transmitter & Receiver
Check the **Use Common PLLs for Rx and Tx** box to place both the LVDS transmitter and the LVDS receiver in the same Stratix device I/O bank. The Quartus II software allows the transmitter and receiver to share the same fast PLL when they use the same input clock. Although you must separate the transmitter and receiver modules in your design, the Quartus II software merges the fast PLLs when appropriate and give you the following message:

Receiver fast PLL <*lvds_rx PLL name*> and transmitter fast PLL <*lvds_tx PLL name*> are merged together

The Quartus II software provides the following message when it cannot merge the fast PLLs for the LVDS transmitter and receiver pair in the design:

Can't merge transmitter-only fast PLL <*lvds_tx PLL name*> and receiver-only fast PLL <*lvds_rx PLL name*>

rx_outclock Resource
You can use either the global or regional clock for the `rx_outclock` signal. If you select **Auto** in the Quartus II software, the tool uses any available lines.

LVDS Transmitter Module

The Quartus II software calculates the inclock boost (*W*) factor for the LVDS transmitter based on input data rate, input clock frequency, and the deserialization factor. In addition to setting the data and clock alignment, you can also set the outclock divide factor (*B*) for the transmitter output clock and add the `pll_enable`, `tx_locked`, and `tx_coreclock` ports. Table 5–16 explains the function of the available ports in the LVDS transmitter block.

Table 5–16. LVDS Transmitter Ports

Port Name	Direction	Function	Input port Source/Output port Destination
`tx_in[Deserialization_factor * number_of_channels - 1..0]`	Input	Input data	Logic array
`tx_inclock`	Input	Reference input clock	Pin or output clock from a PLL
`tx_pll_enable`	Input	Fast PLL enable	Pin *(1)*, *(2)*, *(3)*
`tx_out[number_of_channels - 1..0]`	Output	Serialized LVDS data signal	Pin
`tx_outclock`	Output	External reference clock	Pin
`tx_coreclock`	Output	Internal reference clock	Pin, logic array, or input clock to a fast PLL *(1)*
`tx_locked`	Output	Fast PLL locked pin	Pin or logic array *(1)*, *(2)*, *(3)*

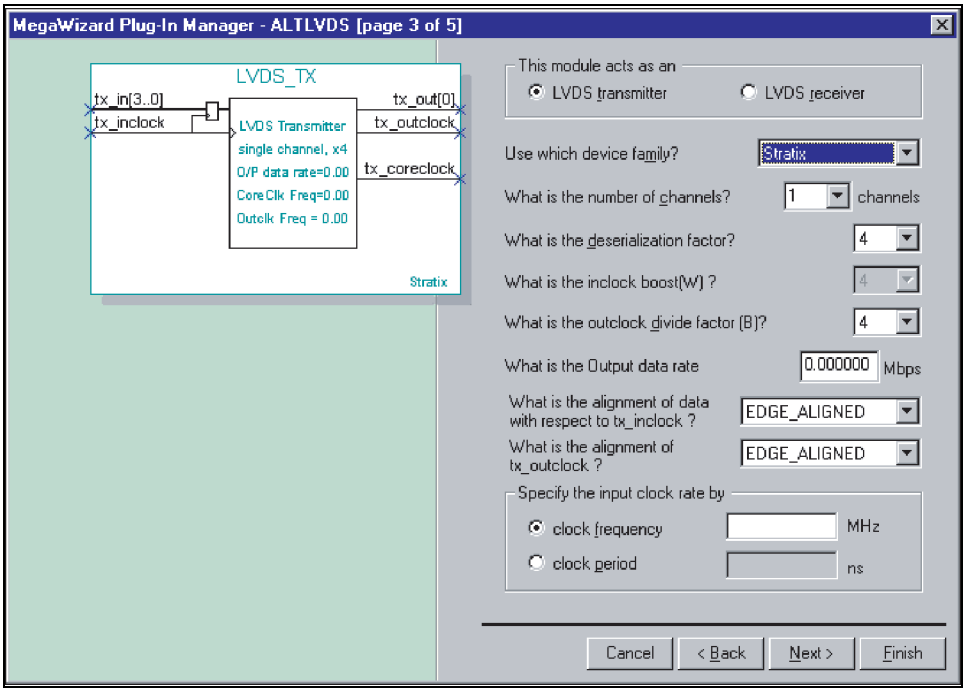
Notes to Table 5–16:
(1) This is an optional port.
(2) Only one `tx_pll_enable` pin is necessary to enable all the PLLs in the device.
(3) This is a non-differential pin.

You can also use the `altlvds` MegaWizard Plug-In Manager to create an LVDS transmitter block. The following sections explain the parameters available in the Plug-In Manager when creating an LVDS transmitter block.

Page 3 of the altlvds_tx MegaWizard Plug-In Manager

This section describes the parameters found on page 3 of the `altlvds_tx` MegaWizard Plug-In Manager (see Figure 5–42).

Figure 5–42. Page 3 of the Transmitter altlvds MegaWizard Plug-In Manager



Number of Channels

The **What is the number of channels?** parameter specifies the number of transmitter channels required and the width of the `tx_in` port. You can have more than 20 channels in a transmitter or receiver module by typing in the required number instead of choosing a number from the drop down menu, which only has selections of up to 20 channels.

Deserialization Factor

The **What is the deserialization factor?** parameter specifies the number of bits per channel. The transmitter block supports deserialization factors of 4, 7, 8, and 10. Based on the factor specified, the Quartus II software determines the multiplication and/or division factor for the LVDS PLL in order to serialize the data.

Table 5–5 on page 5–32 lists the differential bit naming convention. The parallel data for the n^{th} channel spans from the MSB (`rx_out` bit $[(J \times n) - 1]$) to the LSB (`rx_out` bit $[J \times (n - 1)]$), where J is the

deserialization factor. The total width of the tx_in port of the transmitter is equal to the number of channels multiplied by the deserialization factor.

Outclock Divide Factor

The **What is the Output data rate?** parameter specifies the ratio of the tx_outclock frequency compared to the data rate. The default value for this parameter is the value of the deserialization factor parameter. The tx_outclock frequency is equal to [*W*/*B*] x input clock frequency. There is also an optional tx_coreclock port which has the same frequency as the [*W*/*J*] × input frequency.

The outclock divide factor is useful for applications that do not require the data rate to be the same as the clock frequency. For example, HyperTransport technology uses a half-clock data rate scheme where the clock frequency is half the data rate. Table 5–17 shows the supported outclock divide factor for a given deserialization factor.

Table 5–17. Deserialization Factor (J) vs. Outclock Divide Factor (B)

Deserialization Factor (*J*)	Outclock Divide Factor (*B*)
4	1, 2, 4
7	1, 7*(1)*
8	1, 2, 4, 8
10	1, 2, 10

Note to Table 5–17:
(1) The clock does not have a 50% duty cycle when b=7 in x7 mode.

Output Data Rate

The **What is the Output data rate** parameter specifies the data rate out of the fast PLL and determines the input clock boost/multiplication factor needed for the transmitter. This parameter must be larger than the input clock frequency and has a maximum rate of 840 Mbps for Stratix devices. The input clock boost factor (*W*) is the output data rate divided by the input clock frequency. The Stratix SERDES circuitry supports input clock boost factors of 4, 7, 8, or 10. The maximum output data rate is 840 Mbps, while the clock has a maximum output of 500 MHz.

Data Alignment with Clock

Use the **What is the alignment of data with respect to tx_inclock?** parameter and the **What is the alignment of tx_outclock?** to align the input and output data, respectively, with the clock. For most applications, the data is edge-aligned with the clock. However, there are applications where the data must be center-aligned with respect to the clock. With

Stratix devices, you can align the input data with respect to the tx_inclock port and align the output data with respect to the tx_outclock port. The MegaWizard Plug-In Manager uses the alignment of input and output data to automatically calculate the phase for the fast PLL outputs. Both of these parameters default to EDGE_ALIGNED, and other values are CENTER_ALIGNED, 45_DEGREES, 135_DEGREES, 180_DEGREES, 225_DEGREES, 270_DEGREES, and 315_DEGREES. CENTER_ALIGNED is the same as 180 degrees aligned and is required for the HyperTransport technology I/O standard.

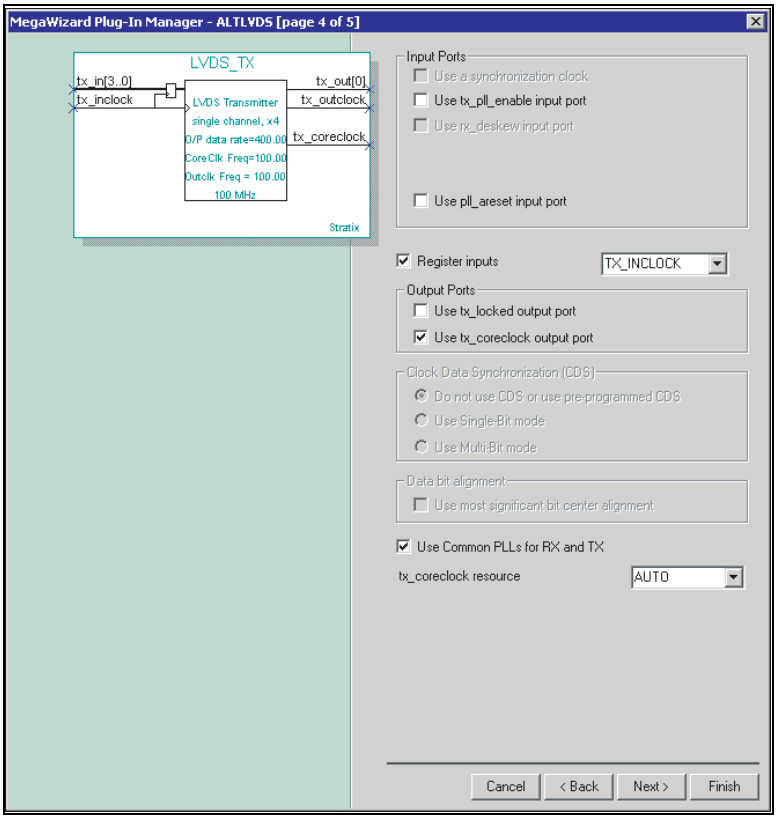
Clock Frequency & Clock Period

The fields in the **Specify the input clock rate by** box specify either the frequency or the period of the input clock going into the fast PLL. However, you cannot specify both. If your design uses the same input clock to feed a transmitter and a receiver module simultaneously, the Quartus II software can merge the fast PLLs for both the transmitter and receiver when the **Use common PLLs for Tx & Rx** option is turned on.

Page 4 of the altlvds_tx MegaWizard Plug-In Manager

This section describes the parameters found on page 4 of the altlvds_tx MegaWizard Plug-In Manager (see Figure 5–43).

Figure 5–43. Page 4 of the Transmitter altlvds MegaWizard Plug-In Manager



Registered Inputs

Check the **Register inputs** box if the input data to the transmitter is not registered just before it feeds the transmitter module. You can choose either `tx_clkin` or `tx_coreclk` to clock the transmitter data (`tx_in[]`) signal. This serves as the register boundary. The number of registers used is proportional to the deserialization factor (*J*). The Quartus II software places the synchronization registers with the LEs in the same row and closest to the SERDES circuitry.

Use Common PLL for Transmitter & Receiver

Check the **Use Common PLLs for Rx and Tx** box to place both the LVDS transmitter and receiver in the same I/O bank in Stratix devices. The Quartus II software also allows the transmitter and receiver to share the PLL when the same input clock is used for both. Although you must

separate the transmitter and receiver in your design, the Quartus II software merges the fast PLLs when appropriate and gives you the following message:

Receiver fast PLL <lvds_rx pll name> and transmitter fast PLL <lvds_tx pll name> are merged together

The Quartus II software gives the following message when it cannot merge the fast PLLs for the LVDS transmitter and receiver pair in the design:

```
Can't merge transmitter-only fast PLL
<lvds_tx pll name> and receiver-only fast PLL
<lvds_rx pll name>
```

tx_outclock Resource
You can use either the global or regional clock for the `tx_outclock` signal. If you select **Auto** in the Quartus II software, the tool uses any available lines.

SERDES Bypass Mode

You can bypass the SERDES block if your data rate is less than 624 Mbps, and you must bypass the SERDES block for the ×1 and ×2 LVDS modules.

Since you cannot route the fast PLL output to an output pin, you must create additional DDR I/O circuitry for the transmitter clock output. To create an ×*J* transmitter output clock, instantiate an `alt_ddio` megafunction clocked by the ×*J* clock with `datain_h` connected to V_{CC} and `datain_l` connected to `GND`.

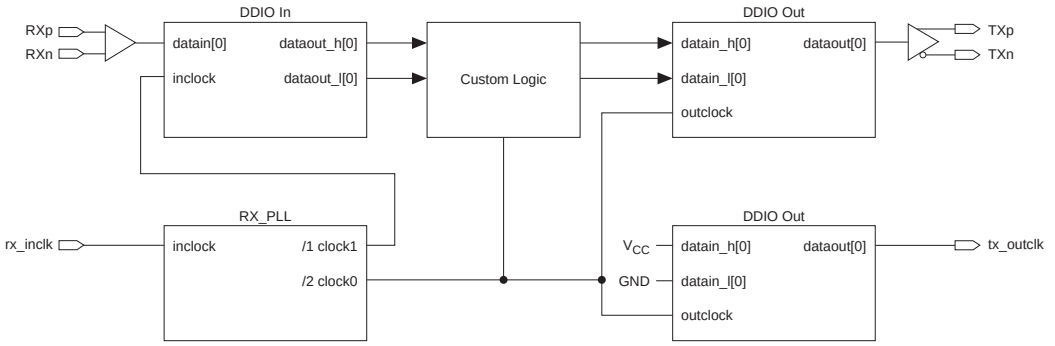
×1 Mode

For ×1 mode, you only need to specify the I/O standard of the pins to tell the Quartus II software that you are using differential signaling. However, Altera recommends using the DDRIO circuitry when the input or output data rate is higher than 231 Mbps. The maximum output clock frequency for ×1 mode is 420 MHz.

×2 Mode

You must use the DDRIO circuitry for ×2 mode. The Quartus II software provides the `altddio_in` and `altddio_out` megafunctions to use for ×2 receiver and ×2 transmitter, respectively. The maximum data rate in ×2 mode is 624 Mbps. Figure 5–44 shows the schematic for using DDR circuitry in ×2 mode.

Figure 5–44. LVDS x2 Mode Schematic Using DDR I/O Circuitry



The transmitter output clock requires extra DDR output circuitry that has the input high and input low connected to V_{CC} and GND respectively. The output clock frequency is the same as the input frequency of the DDR output circuitry.

Other Modes

For other modes, you can still to use the DDR circuitry for better frequency performance. You can use either the LEs or the M512 RAM block for the deserialization.

M512 RAM Block as Serializer/Deserializer Interface

In addition to using the DDR circuitry and the M512 RAM block, you need two extra counters per memory block to provide the address for the memory: a fast counter powering up at 0 and a slow counter powering up at 2. The M512 RAM block is configured as a simple dual-port memory block, where the read enable and the write enable signals are always tied high. Figures 5–45 and 5–46 show the block diagram for the SERDES bypass receiver and SERDES bypass transmitter, respectively.

Figure 5–45. SERDES Bypass LVDS Receiver Using M512 RAM Block as the Deserializer

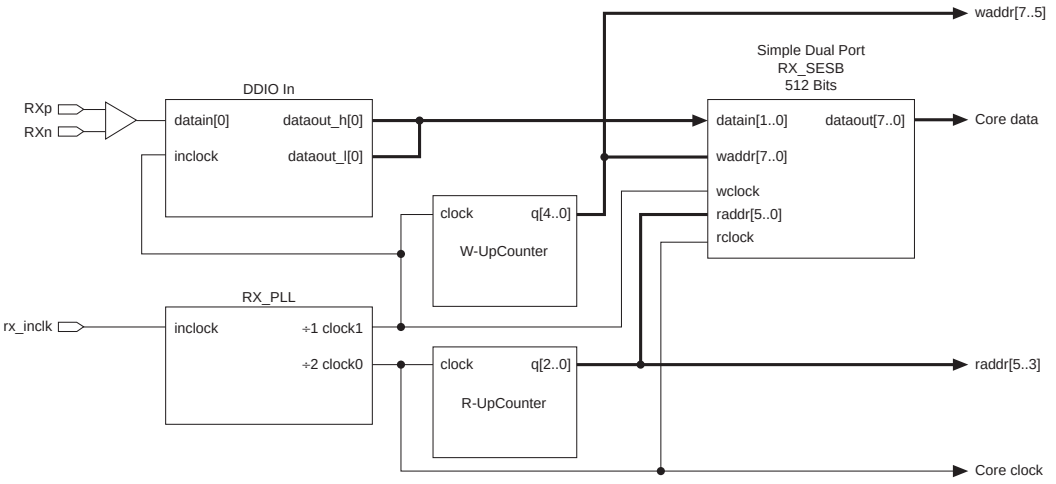
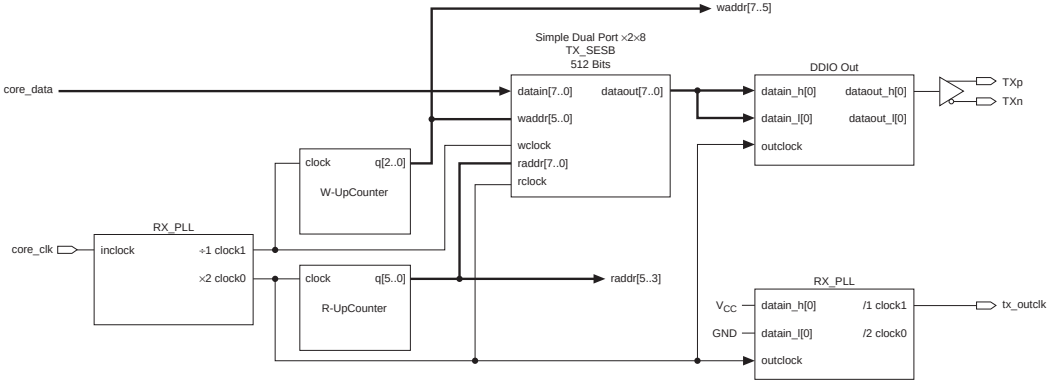


Figure 5–46. SERDES Bypass LVDS Transmitter Using M512 RAM Block as Deserializer



For the transmitter, the read counter is the fast counter and the write counter is the slow counter. For the receiver, the write counter is the fast counter and the read counter is the slow counter. Tables 5–18 and 5–19 provide the address counter configurations for the transmitter and the receiver, respectively.

Table 5–18. Address Counters for SERDES Bypass LVDS Receiver

M512 Mode	Deserialization Factor	Write Up-Counter (Fast Counter)		Read Up-Counter (Slow Counter)		Invalid Initial Cycles	
		Width	Starts at	Width	Starts at	Write	Read
×2×4	4	4	0	3	2	12	6
×2×8	8	5	0	3	2	24	6
×4×16	8	5	0	3	2	24	6
×2×16	16	6	0	3	2	48	6

Table 5–19. Address Counters for SERDES Bypass LVDS Transmitter

M512 Mode	Deserialization Factor	Write Up-Counter (Fast Counter)		Read Up-Counter (Slow Counter)		Invalid Initial Cycles	
		Width	Starts at	Width	Starts at	Write	Read
×2×4	4	4	0	3	2	2	4
×2×8	8	5	0	3	2	2	8
×4×16	8	5	0	3	2	2	8
×2×16	16	6	0	3	2	2	16

In different M512 memory configurations, the counter width is smaller than the address width, so you must ground some of the most significant address bits. Table 5–20 summarizes the address width, the counter width, and the number of bits to be grounded.

Table 5–20. Address & Counter Width

M512 Mode	Write Counter Width	Read Counter Width	Write Address Width	Read Address Width	Number of Grounded Bits	
					Write Address	Read Address
×2×4	4	3	8	7	4	4
×2×8	5	3	8	6	3	3
×4×16	6	3	7	5	1	2
×2×16	5	3	8	5	3	2

Logic Array as Serializer/Deserializer Interface

The design can use the `lpm_shift_reg` megafunction instead of a simple dual port memory block to serialize/deserialize data. The receiver requires an extra flip-flop clocked by the slow clock to latch on to the deserialized data. The transmitter requires a counter to generate the enable signal for the shift register to indicate the times to load and serialize the data. Figures 5–47 and 5–48 show the schematic of the ×8 LVDS receiver and ×8 LVDS transmitter, respectively, with the logic array performing the deserialization.

This scheme can also be used for APEX II and Mercury device flexible LVDS solutions.

Figure 5–47. SERDES Bypass LVDS Receiver with Logic Array as Deserializer

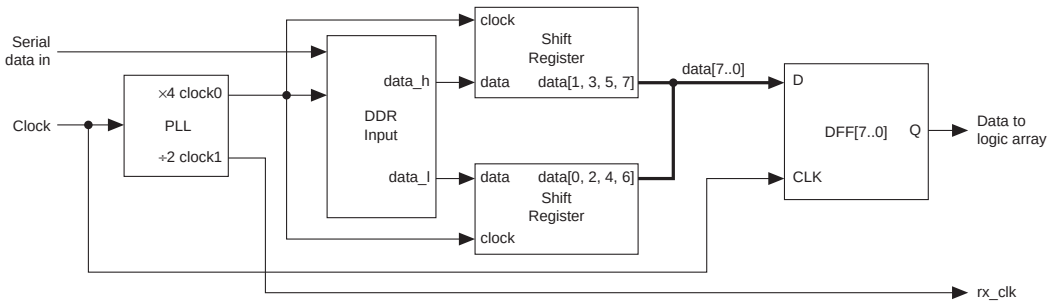
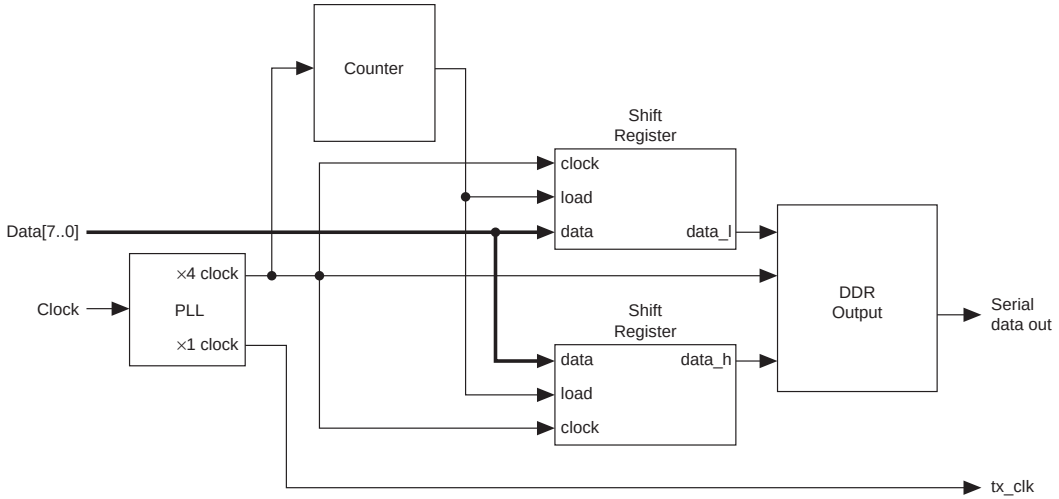


Figure 5–48. SERDES Bypass LVDS Transmitter with Logic Array as Deserializer



Summary

The Stratix device family of flexible, high-performance, high-density PLDs delivers the performance and bandwidth necessary for complex system-on-a-programmable-chip (SOPC) solutions. Stratix devices support multiple I/O protocols to interface with other devices within the system. Stratix devices can easily implement processing-intensive data-path functions that are received and transmitted at high speeds. The Stratix family of devices combines a high-performance enhanced PLD architecture with dedicated I/O circuitry in order to provide I/O standard performances of up to 840 Mbps.