3. Source-Synchronous Signaling With DPA

SGX51003-1.1

Introduction

Expansion in the telecommunications market and growth in Internet use requires systems to move more data faster than ever. To meet this demand, rely on solutions such as differential signaling and emerging high-speed interface standards including RapidIO, POS-PHY 4, SFI-4, or XSBI.

These new protocols support differential data rates up to 1 Gbps and higher. At these high data rates, it becomes more challenging to manage the skew between the clock and data signals. One solution to this challenge is to use CDR to eliminate skew between data channels and clock signals. Another potential solution, DPA, is beginning to be incorporated into some of these protocols.

The source-synchronous high-speed interface in Stratix GX devices is a dedicated circuit embedded into the PLD allowing for high-speed communications. The *High-Speed Source-Synchronous Differential I/O Interfaces in Stratix GX Devices* chapter of the *Stratix GX Device Handbook, Volume 2* provides information on the high-speed I/O standard features and functions of the Stratix GX device.

Stratix GX I/O Banks

Stratix GX devices contain 17 I/O banks. I/O banks one and two support high-speed LVDS, LVPECL, and 3.3-V PCML inputs and outputs. These two banks also incorporate an embedded dynamic phase aligner within the source-synchronous interface (see Figure 3–8 on page 3–10). The dynamic phase aligner corrects for the phase difference between the clock and data lines caused by skew. The dynamic phase aligner operates automatically and continuously without requiring a fixed training pattern, and allows the source-synchronous circuitry to capture data correctly regardless of the channel-to-clock skew.

Principles of SERDES Operation

Stratix GX devices support source-synchronous differential signaling up to 1 Gbps in DPA mode, and up to 840 Mbps in non-DPA mode. 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 8 or 10 for the DPA mode, or 4, 7, 8, or 10 for all other modes. The SERDES factor does not have to equal the clock multiplication value. The ×1 and ×2 operation is also possible by bypassing the SERDES. The SERDES DPA cannot support ×1, ×2, or ×4 natively.

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 two dedicated fast PLLs each in EP1SGX10 to EP1SGX25 devices, and four in EP1SGX40 devices. These PLLs are used for the SERDES operations as well as general-purpose use.

Stratix GX Differential I/O Receiver Operation (Non-DPA Mode)

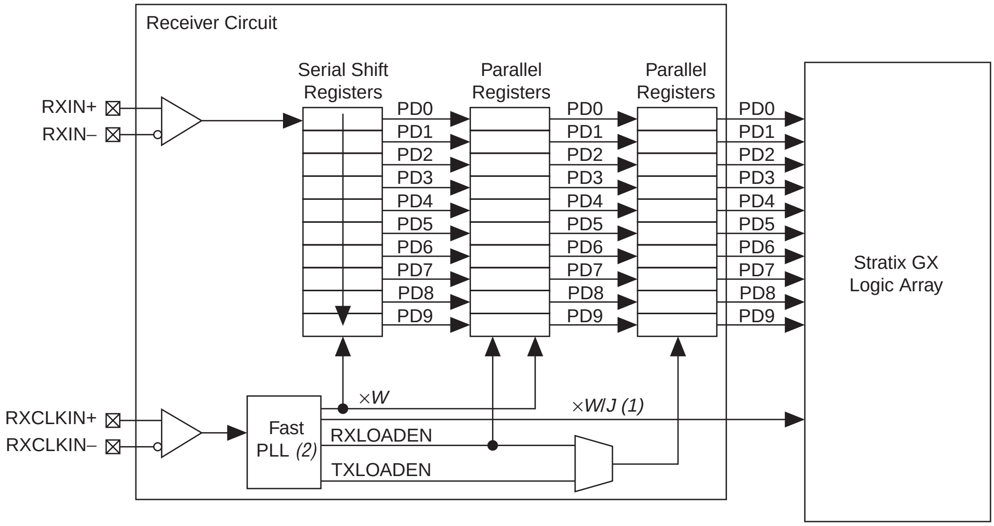
You can configure any of the Stratix GX source synchronous differential input channels as a receiver channel (see Figure 3–1). The differential receiver deserializes the incoming high-speed data. The input shift register continuously clocks the incoming data on the negative transition of the high-frequency 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.

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 3–1 shows the block diagram of a single SERDES receiver channel. Figure 3–2 shows the timing relationship between the data and clocks in Stratix GX devices in ×10 mode. *W* is the low-frequency multiplier and *J* is the data parallelization division factor.

Figure 3–1. Stratix GX High-Speed Interface Deserialized in ×10 Mode

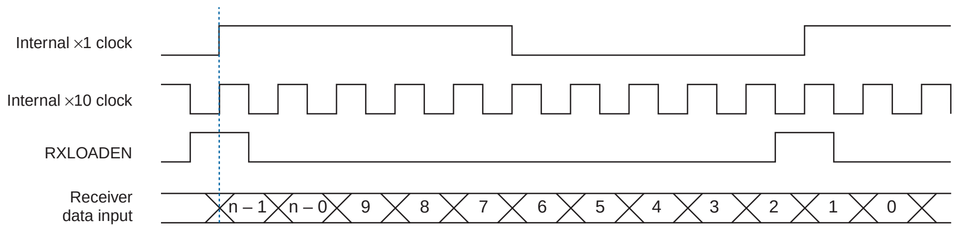


Notes to Figure 3–1:

(1) *W* = 1, 2, 4, 7, 8, or 10.
J = 4, 7, 8, or 10 for non-DPA (*J* = 8 or 10 for DPA).
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 3–2. Receiver Timing Diagram



Stratix GX Differential I/O Transmitter Operation

You can configure any of the Stratix GX differential output channels as a transmitter channel. The differential transmitter serializes 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 3–3 shows the block diagram of a single SERDES transmitter channel and Figure 3–4 shows the timing relationship between the data and clocks in Stratix GX devices in ×10 mode. *W* is the low-frequency multiplier and *J* is the data parallelization division factor.

Figure 3–3. Stratix GX High-Speed Interface Serialized in ×10 Mode

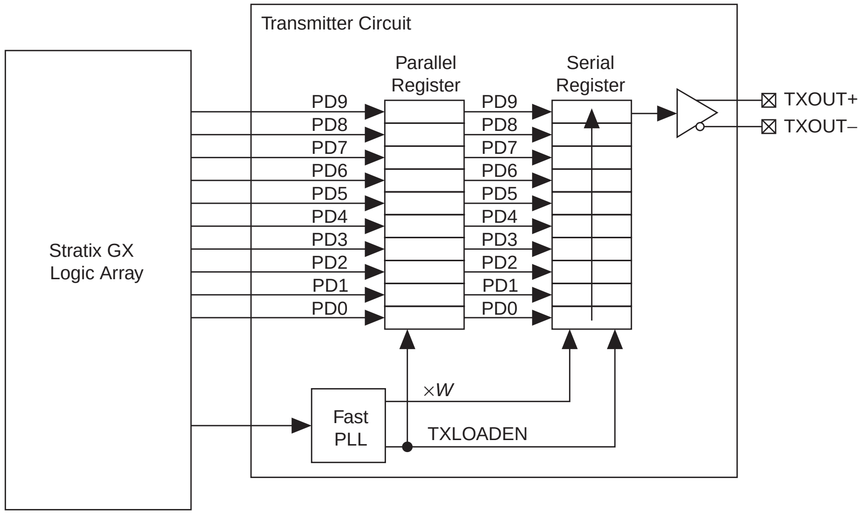
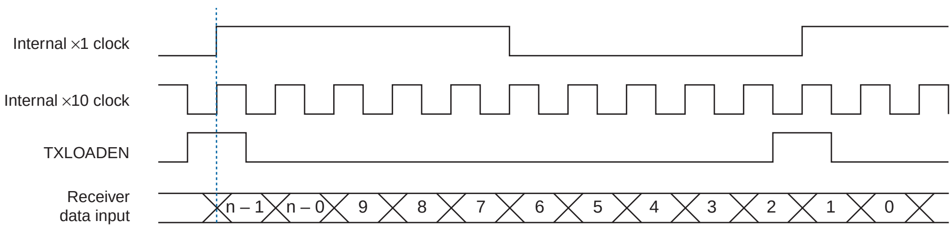


Figure 3–4. Transmitter Timing Diagram

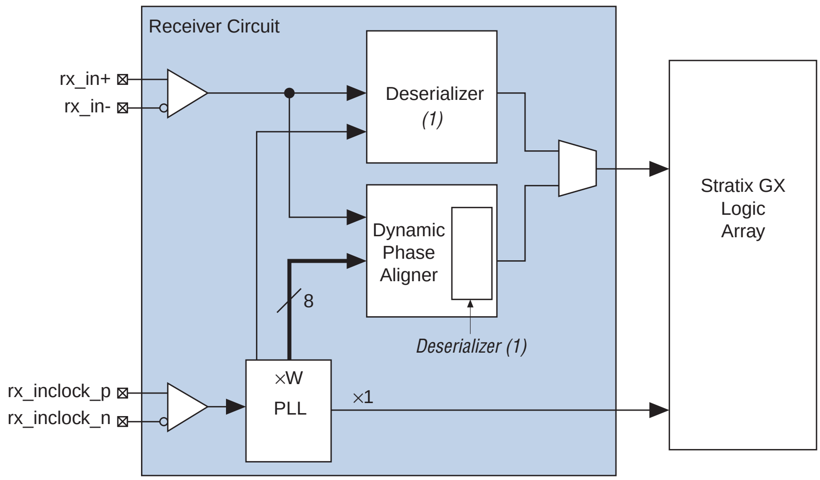


DPA Block Overview

Each Stratix GX receiver channel features a DPA block. The block contains a dynamic phase selector for phase detection and selection, a SERDES, a synchronizer, and a data realigner circuit. You can bypass the dynamic phase aligner without affecting the basic source-synchronous operation of the channel by using a separate deserializer shown in Figure 3–5.

The dynamic phase aligner uses both the source clock and the serial data. The dynamic phase aligner automatically and continuously tracks fluctuations caused by system variations and self-adjusts to eliminate the phase skew between the multiplied clock and the serial data. Figure 3–5 shows the relationship between Stratix GX source-synchronous circuitry and the Stratix GX source-synchronous circuitry with DPA.

Figure 3–5. Source-Synchronous DPA Circuitry



Note to Figure 3–5:

(1) Both deserializers are identical. The deserializer operation is described in the "Principles of SERDES Operation" section.

Unlike the de-skew function in APEX™ 20KE and APEX 20KC devices, you do not have to use a fixed training pattern with DPA in Stratix GX devices. Table 3–1 shows the differences between source-synchronous circuitry with DPA and source-synchronous circuitry without DPA circuitry in Stratix GX devices.

Table 3–1. Source-Synchronous Circuitry With & Without DPA

Feature	Source-Synchronous Circuitry	
	Without DPA	With DPA
Data rate	300 to 840 Megabits per second (Mbps)	300 Mbps to 1 Gbps
Deserialization factors	1, 2, 4, 8, 10	8, 10
Clock frequency	10 to 717 MHz	74 to 717 MHz
Interface pins	I/O banks 1 and 2	I/O banks 1 and 2
Receiver pins	Dedicated inputs	Dedicated inputs

DPA Input Support

Stratix GX device I/O banks 1 and 2 contain dedicated circuitry to support differential I/O standards at speeds up to 1 Gbps with DPA (or up to 840 Mbps without DPA). Stratix GX device source-synchronous circuitry supports LVDS, LVPECL, and 3.3-V PCML I/O standards, each with a supply voltage of 3.3 V. Refer to the *High-Speed Source-Synchronous Differential I/O Interfaces in Stratix GX Devices* chapter of the *Stratix GX Device Handbook, Volume 2* for more information on these I/O standards. Transmitter pins can be either input or output pins for single-ended I/O standards. Refer to Table 3–2.

Table 3–2. Bank 1 & 2 Input Pins

Input Pin Type	I/O Standard	Receiver Pin	Transmitter Pin
Differential	Differential	Input only	Output only
Single ended	Single ended	Input only	Input or output

Interface & Fast PLL

This section describes the number of channels that support DPA and their relationship with the PLL in Stratix GX devices. EP1SGX10 and EP1SGX25 devices have two dedicated fast PLLs and EP1SGX40 devices have four dedicated fast PLLs for clock multiplication. Table 3–3 shows the maximum number of channels in each Stratix GX device that support DPA.

Table 3–3. Stratix GX Source-Synchronous Differential I/O Resources

Device	Fast PLLs	Pin Count	Receiver Channels *(1)*	Transmitter Channels *(1)*	Receiver & Transmitter Channel Speed (Gbps) *(2)*	LEs
EP1SGX10C	2 *(3)*	672	22	22	1	10,570
EP1SGX10D	2 *(3)*	672	22	22	1	10,570
EP1SGX25C	2	672	39	39	1	25,660
EP1SGX25D	2	672	39	39	1	25,660
		1,020	39	39	1	25,660
EP1SGX25F	2	1,020	39	39	1	25,660
EP1SGX40D	4 *(4)*	1,020	45	45	1	41,250
EP1SGX40G	4 *(4)*	1,020	45	45	1	41,250

Notes to Table 3–3:

(1) This is the number of receiver or transmitter channels in the source-synchronous (I/O bank 1 and 2) interface of the device.

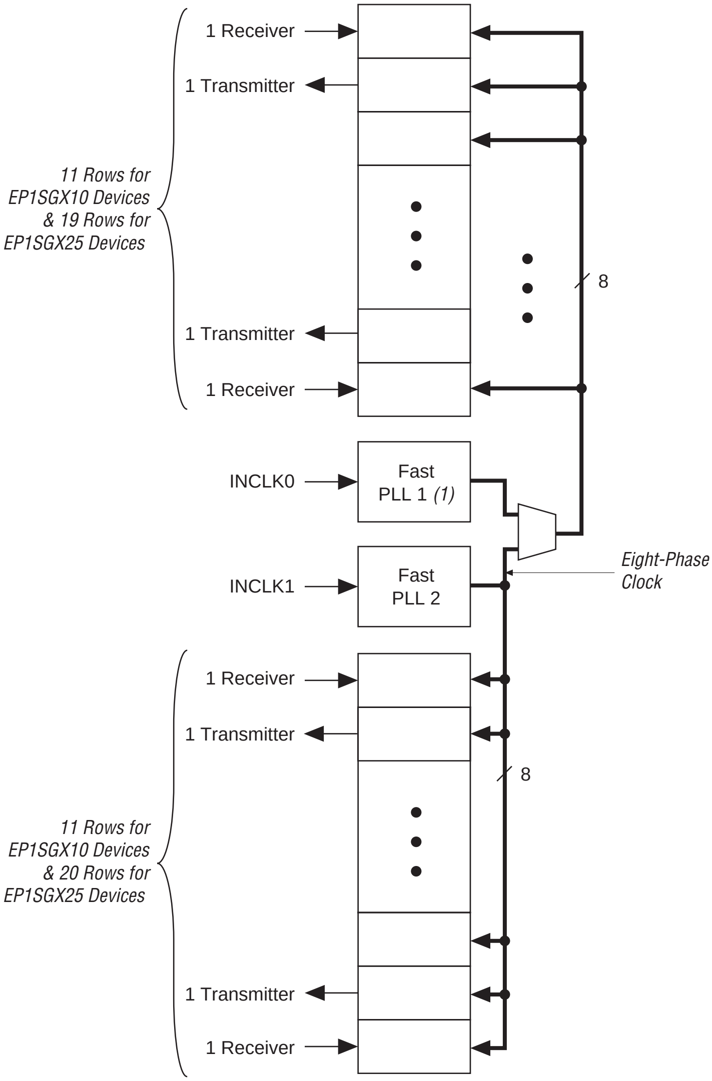
(2) Receiver channels operate at 1,000 Mbps with DPA. Without DPA, the receiver channels operate at 840 Mbps.

(3) One of the two fast PLLs in EP1SGX10C and EP1SGX10D devices supports DPA.

(4) Two of the four fast PLLs in EP1SGX40D and EP1SGX40G devices support DPA

The receiver and transmitter channels are interleaved so that each I/O row in I/O banks 1 and 2 of the device has one receiver channel and one transmitter channel per row. Figures 3–6 and 3–7 show the fast PLL and channels with DPA layout in EP1SGX10, EP1SGX25, and EP1SGX40 devices. In EP1SGX10 devices, only fast PLL 2 supports DPA operations.

Figure 3–6. PLL & Channel Layout in EP1SGX10 & EP1SGX25 Devices *Notes (1), (2)*

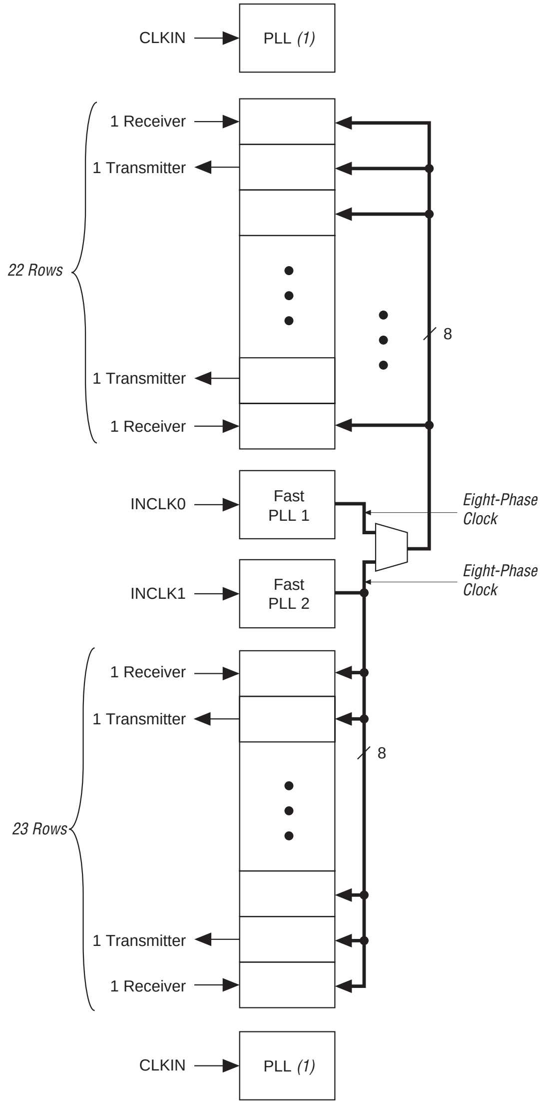


Notes to Figure 3–6:

(1) Fast PLL 1 in EP1SGX10 devices does not support DPA.

(2) Not all eight phases are used by the receiver channel or transmitter channel in non-DPA mode.

Figure 3–7. PLL & Channel Layout in EP1SGX40 Devices *Notes (1), (2), (3)*



Notes to Figure 3–7:

(1) Corner PLLs do not support DPA.

(2) Not all eight phases are used by the receiver channel or transmitter channel in non-DPA mode.

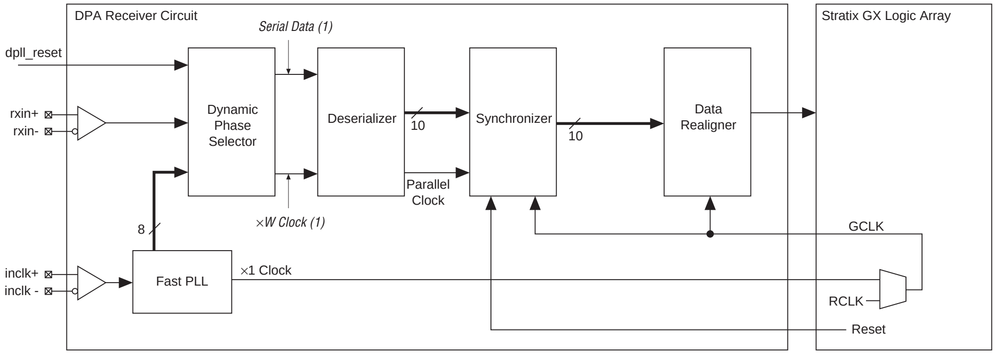
(3) The center PLLs can only clock 20 transceivers in either direction. Using Fast PLL2, you can clock a total of 40 transceivers, 20 in each direction.

DPA Operation

The DPA receiver circuitry contains the dynamic phase selector, the deserializer, the synchronizer, and the data realigner (see Figure 3–8). This section describes the DPA operation, synchronization and data realignment. In the SERDES with DPA mode, the source clock is fed to the fast PLL through the dedicated clock input pins. This clock is multiplied by the multiplication value *W* to match the serial data rate.

For information on the deserializer, see "Principles of SERDES Operation" on page 3–1.

Figure 3–8. DPA Receiver Circuit



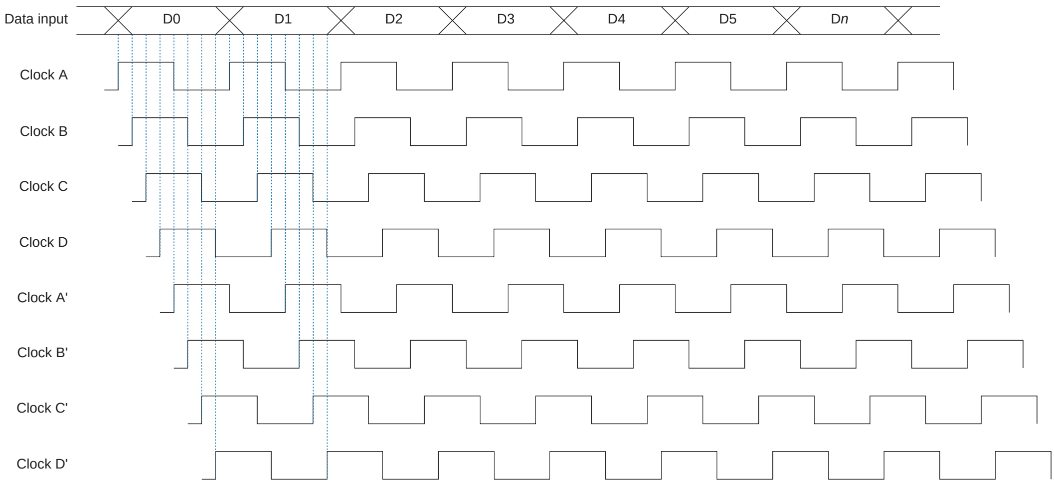
Note to Figure 3–8:

(1) These are phase-matched and retimed high-speed clocks and data.

The dynamic phase selector matches the phase of the high-speed clock and data before sending them to the deserializer.

The fast PLL supplies eight phases of the same clock (each a separate tap from a four-stage differential VCO) to all the differential channels associated with the selected fast PLL. The DPA circuitry inside each channel locks to a phase closest to the serial data's phase and sends the retimed data and the selected clock to the deserializer. The DPA circuitry automatically performs this operation and is not something you select. Each channel's DPA circuit can independently choose a different clock phase. The data phase detection and the clock phase selection process is automatic and continuous. The eight phases of the clock give the DPA circuit a granularity of one eighth of the unit interval (UI) or 125 ps at 1 Gbps. Figure 3–9 illustrates the clocks generated by the fast PLL circuitry and their relationship to a data stream.

Figure 3–9. Fast PLL Clocks & Data Input



Protocols, Training Pattern & DPA Lock Time

The dynamic phase aligner uses a fast PLL for clock multiplication, and the dynamic phase selector for the phase detection and alignment. The dynamic phase aligner uses the high-speed clock out of the dynamic phase selector to deserialize high-speed data and the receiver's source synchronous operations.

At each rising edge of the clock, the dynamic phase selector determines the phase difference between the clock and the data and automatically compensates for the phase difference between the data and clock.

The actual lock time for different data patterns varies depending on the data's transition density (how often the data switches between 1 and 0) and jitter characteristic. The DPA circuitry is designed to lock onto any data pattern with sufficient transition density, so the circuitry works with current and future protocols. Experiments and simulations show that the DPA circuitry locks when the data patterns listed in Table 3–4 are repeated for the specified number of times. There are other suitable patterns not shown in Table 3–4 and/or pattern lengths, but the lock time may vary. The circuit can adjust for any phase variation that may occur during operation.

Table 3–4. Training Patterns for Different Protocols

Protocols	Training Pattern	Number of Repetitions
SPI-4, NPSI	Ten 0's, ten 1's (`00000000001111111111`)	256
RapidIO	Four 0's, four 1's (`00001111`) or one 1, two 0's, one 1, four 0's (`10010000`)	
Other designs	Eight alternating 1's and 0's (`10101010` or `01010101`)	
SFI-4, XSBI	Not specified	

Phase Synchronizer

Each receiver has its own phase synchronizer. The receiver phase synchronizer aligns the phase of the parallel data from all the receivers to one global clock. The synchronizers in each channel consist of a 4-bit deep and *J*-bit wide FIFO buffer. The parallel clock writes to the FIFO buffer and the global clock (`GCLK`) reads from the FIFO buffer. The global and parallel clock inputs into the synchronizers must have identical frequencies and differ only in phase. The FIFO buffer never becomes full or empty (because the source and receive signals are frequency locked) when operating within the DPA specifications, and the operation does not require an empty/full flag or read/write enable signals.

Receiver Data Realignment In DPA Mode

While DPA operation aligns the incoming clock phase to the incoming data phase, it does not guarantee the parallelization boundary or byte boundary. When the dynamic phase aligner realigns the data bits, the bits may be shifted out of byte alignment, as shown in Figure 3–10.

Figure 3–10. Misaligned Captured Bits

Correct Alignment

0	1	2	3	4	5	6	7

Incorrect Alignment

3	4	5	6	7	0	1	2

The dynamic phase selector and synchronizer align the clock and data based on the power-up of both communicating devices, and the channel to channel skew. However, the dynamic phase selector and synchronizer cannot determine the byte boundary, and the data may need to be byte-aligned. The dynamic phase aligner's data realignment circuitry shifts data bits to correct bit misalignments.

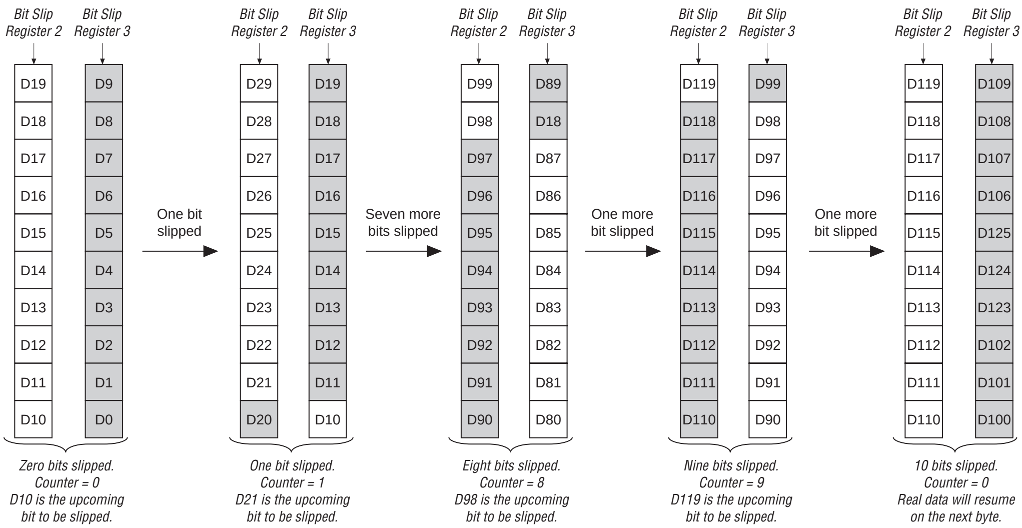
The Stratix GX circuitry contains a data-realignment feature controlled by the logic array. Stratix GX devices perform data realignment on the parallel data after the deserialization block. The data realignment can be performed per channel for more flexibility. The data alignment operation requires a state machine to recognize a specific pattern. The procedure requires the bits to be slipped on the data stream to correctly align the incoming data to the start of the byte boundary.

The DPA uses its realignment circuitry and the global clock for data realignment. Either a device pin or the logic array asserts the internal `rx_channel_data_align` node to activate the DPA data-realignment circuitry. Switching this node from low to high activates the realignment circuitry and the data being transferred to the logic array is shifted by one bit. The data realignment block cannot be bypassed. However, if the `rx_channel_data_align` is not turned on (through the `altvlds` MegaWizard Plug-In Manager), or when it is not toggled, it only acts as a register latency.

A state machine and additional logic can monitor the incoming parallel data and compare it against a known pattern. If the incoming data pattern does not match the known pattern, you can activate the `rx_channel_data_align` node again. Repeat this process until the realigner detects the desired match between the known data pattern and incoming parallel data pattern.

The DPA data-realignment circuitry allows further realignment beyond what the *J* multiplication factor allows. You can set the *J* multiplication factor to be 8 or 10. However, because data must be continuously clocked in on each low-speed clock cycle, the upcoming bit to be realigned and previous $n - 1$ bits of data are selected each time the data realignment logic's counter passes $n - 1$. At this point the data is selected entirely from bit-slip register 3 (see Figure 3–11) as the counter is reset to 0. The logic array receives a new valid byte of data on the next divided low speed clock cycle. Figure 3–11 shows the data realignment logic output selection from data in the data realignment register 2 and data realignment register 3 based on its current counter value upon continuous request of data slipping from the logic array.

Figure 3–11. DPA Data Realigner



Use the `rx_channel_data_align` signal within the device to activate the data realigner. You can use internal logic or an external pin to control the `rx_channel_data_align` signal. To ensure the rising edge of the `rx_channel_data_align` signal is latched into the control logic, the `rx_channel_data_align` signal should stay high for at least two low-frequency clock cycles.

To manage the alignment procedure, a state machine should be built in the FPGA logic array to generate the realignment signal. The following guidelines outline the requirements for this state machine.

- The design must include an input synchronizing register to ensure that data is synchronized to the ×*W*/*J* clock.
- After the state machine, use another synchronizing register to capture the generated `rx_channel_data_align` signal and synchronize it to the ×*W*/*J* clock.
- Because 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 *W*/*J* clock periods.
- To guarantee the state machine does not incorrectly generate multiple `rx_channel_data_align` pulses to shift a single bit, the state machine must hold the `rx_channel_data_align` signal low for at least three ×1 clock periods between pulses.