



XAPP871 (v1.0) February 28, 2008

## SERDES Framer Interface Level 5

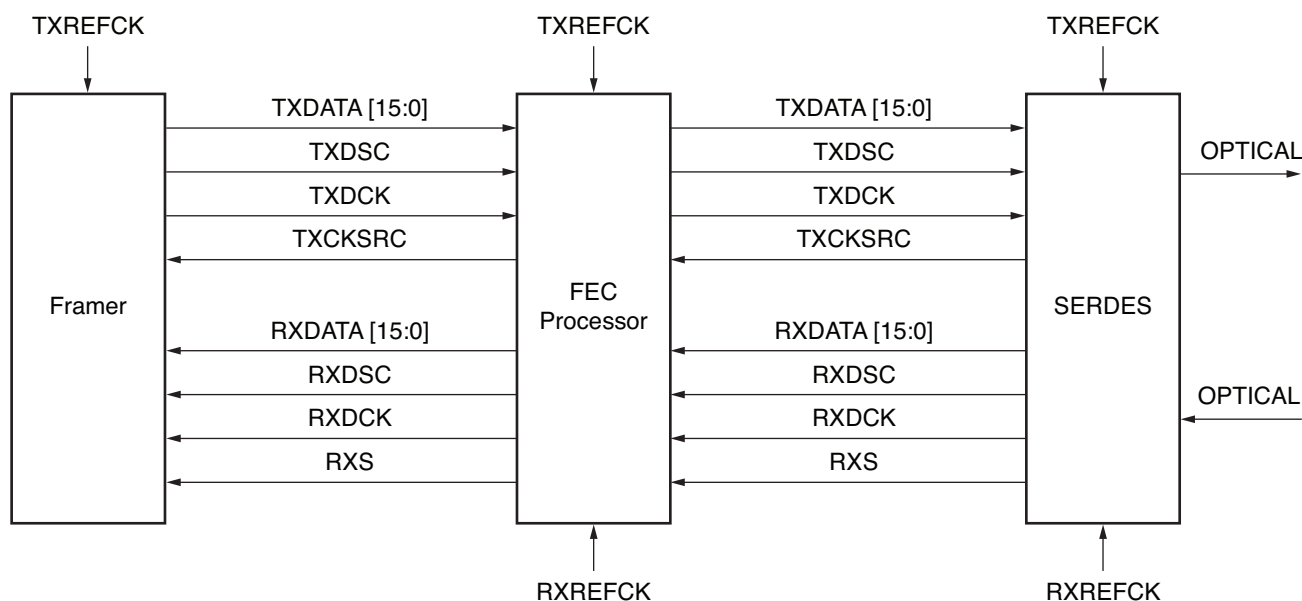
Author: Ralf Krueger

### Summary

This application note describes the implementation of SERDES Framer Interface Level 5 (SFI-5) in a Virtex-5 XC5VLX330T FPGA. SFI-5 is a standard defined by the Optical Internetworking Forum (OIF). The interface must operate bidirectionally at a payload data rate of 40 Gb/s with 0–25% forward error correction (FEC) overhead, up to a maximum of 50 Gb/s. The interface consists of 17 bidirectional GTP transceivers and logic to compensate skew differences between the transmission paths of the data channels.

### Introduction

The SFI-5 system reference model (as defined by the OIF) is shown in Figure 1. SFI-5 is intended to interface between a SERDES component and an FEC processor, between an FEC processor and a framer, or directly between a SERDES component and an FEC processor. The reference model consists of 16 data channels and a 17th channel called the deskew channel (TXDSC/RXDSC), which transmits out-of-band data samples to enable an algorithm in the receiver to deskew the 16 data channels.



X871\_01\_121307

Figure 1: SFI-5 System Reference Model



SFI-5 is a fully synchronous system, meaning that there is only a single reference clock. For example, on the link from the FEC processor to the SERDES in [Figure 1](#), the source reference clock is the same as the sink reference clock. This synchronization can be accomplished using any of these three methods:

Method 1: TXREFCK is physically connected to both the FEC processor and the SERDES.

Method 2: If the FEC processor does not have access to TXREFCK, the SERDES must send the reference clock to the FEC processor via the TXCKSRC signal.

Method 3: If the SERDES does not have access to TXREFCK, the FEC processor must send the reference clock to the SERDES via the TXDCK signal.

In the reference design, TXREFCK is connected directly to both the source and sink devices of the SFI-5 interface (method 1). However, the reference design can easily be modified to support the clocking schemes described in methods 2 and 3 with no change to the logic design.

The same synchronous principles apply to the receive direction (e.g., SERDES to FEC processor) in [Figure 1](#), except that the reference clock is RXREFCK. The OIF specification states that TXREFCK and RXREFCK can be separate clocks, or they can be tied together. Because the transmit and receive circuitry of each GTP transceiver shares a common reference clock, Virtex-5 FPGAs can only support the configuration in which TXREFCK and RXREFCK are tied together.

The signal RXS in [Figure 1](#) is not used in this reference design. There is a port for RXS, but it is tied to 0. If RXS is desired, this functionality must be added by the user. When set to 1, the signal indicates that RXDCK and RXDATA are not derived from the optical receive signal.

The reference design is not fully compliant to the SFI-5 requirement for output skew between the data lanes. The SFI-5 specification budgets 2 unit intervals (UI) of skew at the source device outputs. It budgets a further 3 UI of skew for the transmission lines to the sink device. The sink device is then required to deskew a minimum of 5 UI. The GTP transmitter lane-to-lane skew specification (T\_LLSKEW) for the Virtex-5 FPGA is defined in [DS202](#), *Virtex-5 Data Sheet: DC and Switching Characteristics* as 2 UI + 500 ps, which is greater than the 2 UI requirement. To guarantee full compliance, some of the 3 UI of skew dedicated to the transmission lines can be borrowed such that the sink device still sees no more than 5 UI of skew.

The reference design is not fully compliant to the SFI-5 performance requirements, which state that the interface must operate between 40–50 Gb/s. The maximum specified frequency of the recovered clock (F\_RXREC) in the GTP transceiver limits performance to the values shown in [Table 1](#).

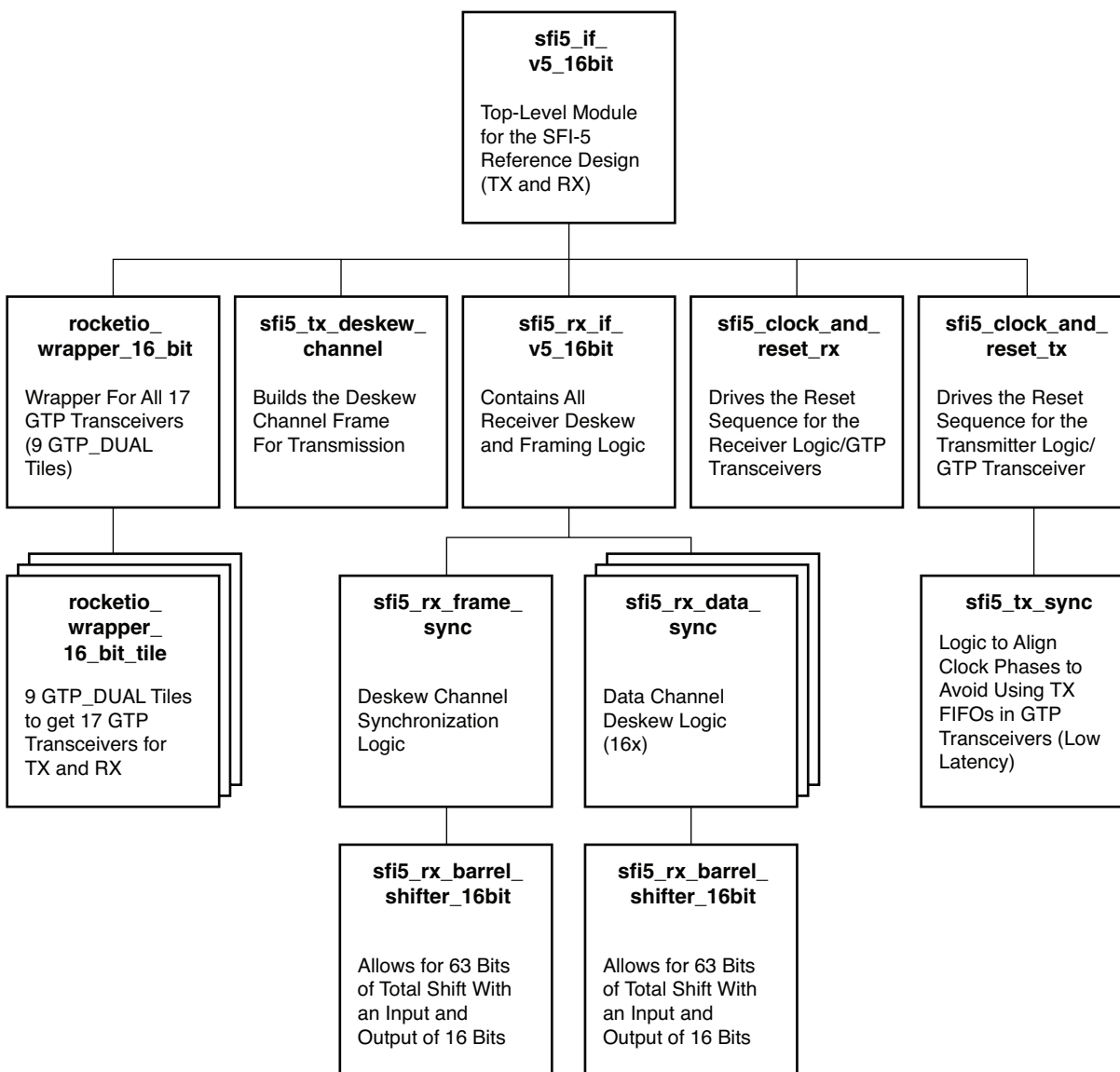
**Table 1: Maximum Performance Targets of SFI-5 Reference Design in XC5VLX330T**

Speed Grade	Maximum F_RXREC (MHz)	Maximum Interface Performance (Gb/s)
–1	320	40.96
–2	375	48
–3	375	48

In LXT devices, the recovered clock is one-eighth the line rate. This application note focuses specifically on implementing SFI-5 in an XC5VLX330T device.

The hierarchy of the HDL modules composing the SFI-5 reference design is shown in [Figure 2](#). Each of these modules is described in detail in this application note. The 17 GTP transceivers are instantiated in pairs (tiles) and then wrapped in a single module that serves as the interface to the rest of the reference design. The transmitter (TX) is composed of deskew channel frame generation logic, and the receiver (RX) is composed of barrel shifters and logic to adjust the delay of each channel to compensate skew between data channels. The TX and RX both have initialization sequences in which all clocks are generated and all circuitry is reset.





X871\_02\_121307

Figure 2: Hierarchy of HDL Modules Composing SFI-5 Interface



## Port List and Descriptions

Table 2 lists the ports used in the SFI-5 interface. All signals are active-High unless stated otherwise. Optional settings should not be left unconnected. They must be set either to user-specific values or to the default values given in this table.

Table 2: Port List and Description of SFI-5 Interface

Port	Type (I/O)	Width	Clock Domain	Description
<b>SFI-5 TX Interface Signals</b>				
TXDATA_P	O	16	Line Rate	SFI-5 TX data channels (P-side).
TXDATA_N	O	16	Line Rate	SFI-5 TX data channels (N-side).
TXDSC_P	O	1	Line Rate	SFI-5 TX deskew channel (P-side).
TXDSC_N	O	1	Line Rate	SFI-5 TX deskew channel (N-side).
TXREFCK	I	2	txrefclk	Reference clock input to GTP transceivers for TX and RX. Frequency is 1/16th line rate (156.25–187.50 MHz, or 2.5–3.0 Gb/s).
TXREFCK_2	I	2	txrefclk	Reference clock input to GTP transceivers for TX and RX. Must be the same frequency as TXREFCK and synchronous to it.
TXDCK	O	2	txusrclk	Differential clock reference at one-eighth line rate forwarded to RX (optional to use in RX).
RXS	O	1	N/A	Receive status (not used, tied to 0).
<b>SFI-5 RX Interface Signals</b>				
RXDATA_P	I	16	Line Rate	SFI-5 RX data channels (P-side).
RXDATA_N	I	16	Line Rate	SFI-5 RX data channels (N-side).
RXDSC_P	I	1	Line Rate	SFI-5 RX deskew channel (P-side).
RXDSC_N	I	1	Line Rate	SFI-5 RX deskew channel (N-side).
<b>Global Signals</b>				
i_RST	I	1	txrefclk	Global reset to initiate reset sequence of TX/RX.
o_TXREFCLK	O	1	txrefclk	User-accessible connection to txrefclk.
o_RESETDONE	O	1	txrefclk	Indicates that all GTP transceivers have completed their reset sequences. Clear by asserting i_CLEAR_MISMATCHES. <sup>(1)</sup>
o_GTPPLL_LOCK	O	1	txrefclk	Indicates that all shared PMA PLLs are locked. Clear by asserting i_CLEAR_MISMATCHES. <sup>(1)</sup>
<b>System-side TX Data/Clock Signals</b>				
iv_TXDATA00_IN [15:0] iv_TXDATA01_IN [15:0] ... iv_TXDATA15_IN [15:0]	I	256	txusrclk2	System-side data input to SFI-5 TX. Serialized by GTP transceivers and transmitted on TXDATA_P and TXDATA_N.
o_TXUSRCLK2	O	1	txusrclk2	User-accessible connection to txusrclk2.



Table 2: Port List and Description of SFI-5 Interface (Cont'd)

Port	Type (I/O)	Width	Clock Domain	Description
<b>System-side RX Data/Clock Signals</b>				
ov_RXDATA00_OUT [15:0] ov_RXDATA01_OUT [15:0] ... ov_RXDATA15_OUT [15:0]	O	256	rxusrclk2	System-side data output from SFI-5 RX. Received on RXDATA_P and RXDATA_N and deserialized by GTP transceivers.
o_RXRECCLK	O	1	rxrecclk	User-accessible connection to rxrecclk, the recovered clock from the SFI-5 RX deskew channel.
o_RXUSRCLK2	O	1	rxusrclk2	User-accessible connection to rxusrclk2.
<b>System-side TX Diagnostics</b>				
o_TX_INIT_DONE	O	1	txrefclk	Indicates that TX reset sequence is complete. Clear by asserting i_CLEAR_MISMATCHES. <sup>(1)</sup>
o_TXDCM_LOCKED	O	1	txrefclk	Digital clock manager (DCM) that produces txusrclk and txusrclk2 is locked. Clear by asserting i_CLEAR_MISMATCHES. <sup>(1)</sup>
i_INSERT_FRAME_ERROR	I	1	txusrclk2	Insert one bit error in frame header of deskew channel.
i_INSERT_DATA_ERROR	I	1	txusrclk2	Insert error on data channel 15.
i_LOOPBACK	I	3	async	GTP transceiver loopback setting for troubleshooting: 000: Normal (no loopback). 001: Near-end physical coding sublayer (PCS) loopback. 010: Near-end physical medium attachment (PMA) loopback. 100: Far-end PMA loopback. 110: Far-end PCS loopback.
<b>System-side RX Diagnostics</b>				
o_RXOOA	O	1	rxusrclk2	RX out of alignment. One or more data channels is misaligned.
o_RXOOA_HISTORY	O	1	rxusrclk2	RX out of alignment history. One or more data channels is misaligned. Clear by asserting i_CLEAR_MISMATCHES. <sup>(1)</sup>
o_RXLOF	O	1	rxusrclk2	RX loss of frame. Framer is not locked to deskew channel frame.
o_RXLOF_HISTORY	O	1	rxusrclk2	RX loss of frame. Framer is not locked to deskew channel frame. Clear by asserting i_CLEAR_FRAME_ERRORS. <sup>(1)</sup>
i_CLEAR_FRAME_ERRORS	I	1	rxusrclk2	Clears frame error count and o_RXLOF_HISTORY.
i_CLEAR_MISMATCHES	I	1	rxusrclk2	Clears mismatch counts and all diagnostic history bits.
ov_FRAME_ERRORS	O	32	rxusrclk2	Wrap-around running count of frame errors.
ov_FRAMES_RECEIVED	O	32	rxusrclk2	Wrap-around running count of frames received.
ov_DATA_MISMATCHES_CH00 ov_DATA_MISMATCHES_CH01 ... ov_DATA_MISMATCHES_CH15	O	32	rxusrclk2	Running count of data mismatches when compared to deskew channel.



Table 2: Port List and Description of SFI-5 Interface (Cont'd)

Port	Type (I/O)	Width	Clock Domain	Description
ov_RXFRAME_SHIFT	O	6	rxusrclk2	Barrel shifter setting of deskew channel.
ov_RXDATA_SHIFT_CH00 ov_RXDATA_SHIFT_CH01 ... ov_RXDATA_SHIFT_CH15	O	6	rxusrclk2	Barrel shifter setting of each data channel.
o_RX_INIT_DONE	O	1	txrefclk	Indicates that RX reset sequence is complete. Clear by asserting i_CLEAR_MISMATCHES. <sup>(1)</sup>
o_TXDCM_LOCKED	O	1	txrefclk	DCM that produces rxusrclk and rxusrclk2 is locked. Clear by asserting i_CLEAR_MISMATCHES. <sup>(1)</sup>
o_RX_BUFFER_UNDERFLOW	O	1	txrefclk	GTP RX elastic buffer underflow. Clear by asserting i_CLEAR_MISMATCHES. <sup>(1)</sup>
o_RX_BUFFER_OVERFLOW	O	1	txrefclk	GTP RX elastic buffer overflow. Clear by asserting i_CLEAR_MISMATCHES. <sup>(1)</sup>
<b>Optional Settings</b>				
i_TX_DRIVER_INHIBIT_STARTUP	I	1	txusrclk2	When set to 1, TX drivers transmit a constant differential 0 value until TX initialization is complete (o_TX_INIT_DONE asserted). Default = 0.
i_TX_DRIVER_INHIBIT_MANUAL	I	1	txusrclk2	When set to 1, TX drivers transmit a constant differential 0 value. Default = 0.
i_TX_PREEMPHASIS	I	3	async	Driver pre-emphasis setting of all GTP transceivers in SFI-5 interface. For details on pre-emphasis settings, see <a href="#">UG196</a> , <i>Virtex-5 FPGA RocketIO GTP Transceiver User Guide</i> . Default = 000 (minimum).
i_TX_DIFF_CTRL	I	3	async	Driver output swing of all GTP transceivers in SFI-5 interface. For details on driver swing settings, see <i>Virtex-5 FPGA RocketIO GTP Transceiver User Guide</i> . Default = 010 (1000 mV differential swing).
i_RX_EQUALIZATION_ENABLE	I	1	async	Enables RX Equalization (active-Low): 0: Enabled. 1: Disabled. Default = 0.
i_RX_EQUALIZATION_MIX	I	2	async	Mixing of high and low frequency signal components. For details on equalization mix settings, see <i>Virtex-5 FPGA RocketIO GTP Transceiver User Guide</i> . Default = 00 (50% wideband, 50% high pass).
i_RX_EQUALIZATION_POLE	I	4	async	Location of pole of RX high-pass filter. For details on equalization pole settings, see <i>Virtex-5 FPGA RocketIO GTP Transceiver User Guide</i> . Default = 1000 (nominal).
i_FRAMES2LOCK	I	7	async	User-defined threshold for the number of consecutive matching frames that need to be seen before deasserting o_RXLOF. Valid settings = 0–127 (decimal). Default = 3F (63 decimal).



Table 2: Port List and Description of SFI-5 Interface (Cont'd)

Port	Type (I/O)	Width	Clock Domain	Description
i_FRAMES2UNLOCK	I	7	async	User-defined threshold for the number of frame mismatches that need to be seen before asserting o_RXLOF. Valid settings = 0–127 (decimal). Default: 3F (63 decimal).
iv_MISMATCHES_2_UNLOCK	I	7	async	User-defined threshold for the number of consecutive data mismatches that need to be seen before asserting o_RXOOA. Valid settings = 0–127 (decimal). Default = 3F (63 decimal).

**Notes:**

1. This is a “history” bit. It is intended to capture an alarm condition that possibly lasts for only a few microseconds. For this reason, when the bit enters the alarm state, it preserves the history until it is manually cleared, regardless of whether or not the alarm state is still present.

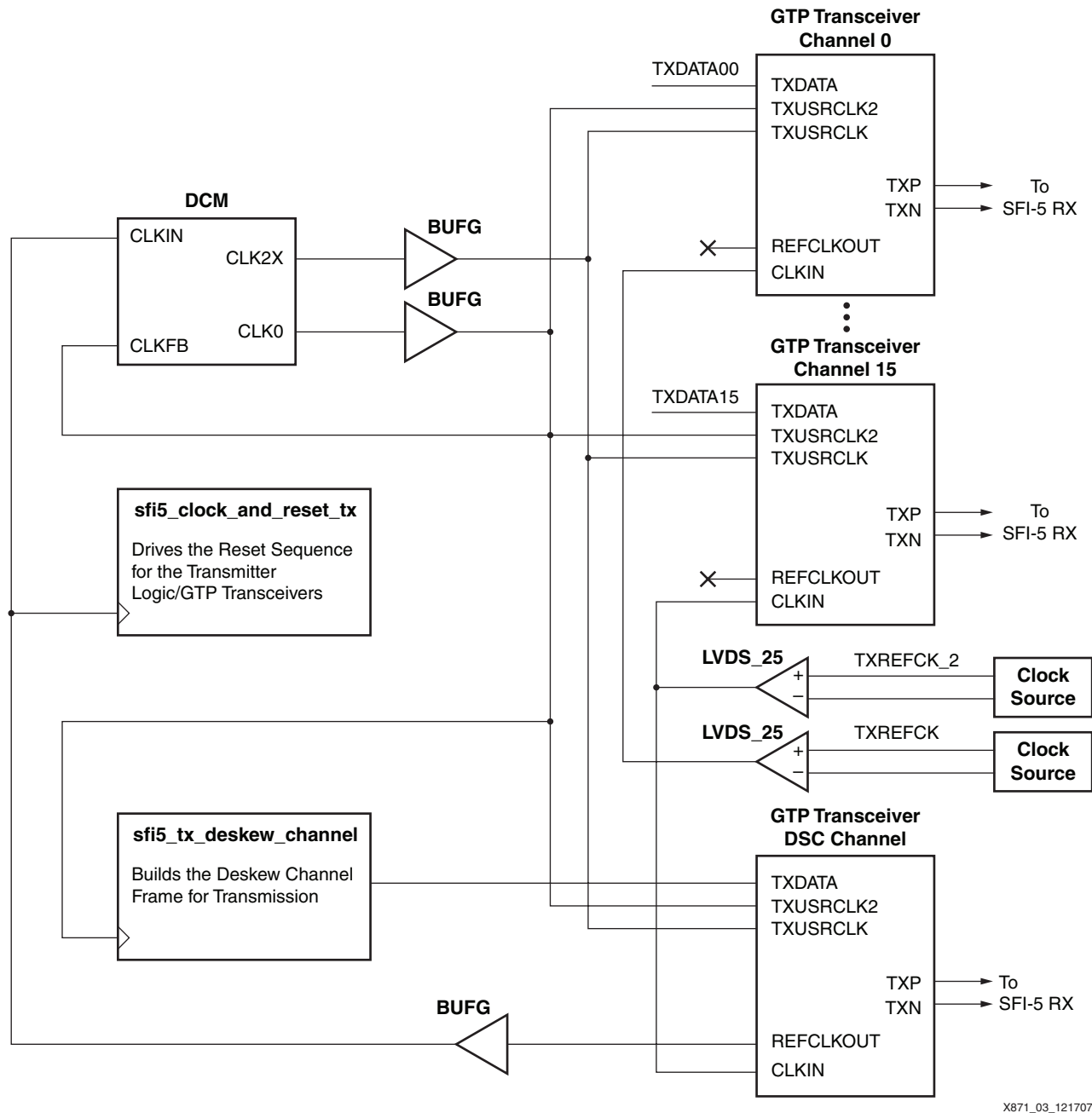
## SFI-5 Clocking

This section describes the various clock domains in the SFI-5 interface. TXREFCK is the clock defined in the OIF specification as the frequency reference for the Framer–FEC–SERDES links in Figure 1. RXREFCK is defined as the frequency reference for the SERDES–FEC–Framer links. As discussed in “Introduction,” page 1, TXREFCK and RXREFCK must always be the same clock in a Virtex-5 FPGA. For this reason, the label TXREFCK is used interchangeably to refer to TXREFCK and RXREFCK. TXREFCK must be provided by an oscillator directly to the dedicated reference clock input of a GTP transceiver. Dedicated routing from the reference clock input to the GTP transceivers ensures that the GTP transceivers receive a clock with minimal jitter. TXREFCK can also be provided to the GTP transceivers via a global clock network in the FPGA, but the path is not ideal for maintaining the highest quality signal. This reference design uses clock sources provided directly to the reference clock input pins of the GTP transceiver.

Figure 3 shows the clocking architecture of the SFI-5 transmitter. TXREFCK is provided by an external clock source and distributed to transceiver channels 8–15, as well as the deskew channel (DSC). TXREFCK\_2 is also provided by an external clock source and distributed to transceiver channels 0–7. Two TXREFCKs are required because the *Virtex-5 FPGA RocketIO GTP Transceiver User Guide* states that a reference clock input on a GTP transceiver can only span a maximum of three tiles above and below that GTP transceiver. Therefore, a reference clock can only span 7 GTP\_DUAL tiles, or 14 GTP transceivers, which is insufficient to clock the 17 channels of the SFI-5 interface. TXREFCK\_2 must be synchronous to TXREFCK and have the same frequency, although their phase relationship is not important.

The REFCLKOUT port of the DSC channel is used to gain access to TXREFCK for the system logic. REFCLKOUT is a forwarded version of CLKIN that is not affected by resetting the GTP transceiver. This is important because the initialization logic for the TX and RX interfaces must run on a clock that is independent of the reset sequence. All logic in the `sfi5_clock_and_reset_tx` module is driven exclusively by TXREFCK via the REFCLKOUT port of the DSC channel. TXREFCK is used as the input to the DCM that generates TXUSRCLK and TXUSRCLK2. TXUSRCLK2 is the same frequency as TXREFCK, but it is important to use TXUSRCLK2 to drive the GTP transceivers because TXUSRCLK and TXUSRCLK2 are nominally phase-aligned at the GTP transceiver inputs. The clocks are aligned as long as both clocks come from DCM outputs. The only logic in the transmitter, besides the initialization logic, is the deskew channel frame generation. This logic is driven by TXUSRCLK2.



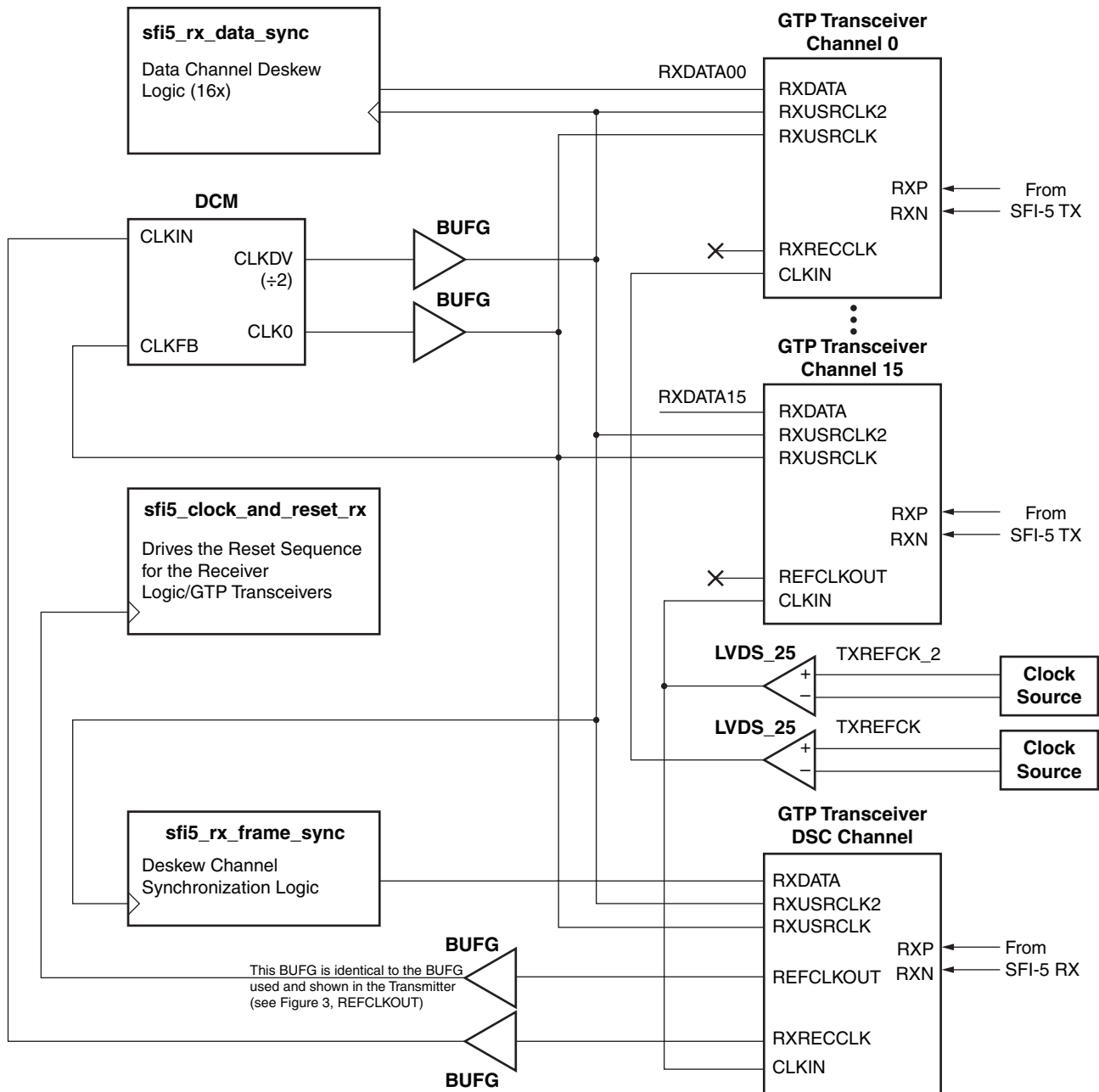


X871\_03\_121707

**Figure 3: SFI-5 Transmitter Clocking**

Figure 4 shows the clocking architecture of the SFI-5 receiver. The reference clock inputs TXREFCK and TXREFCK\_2 are the same inputs shown in Figure 3 because the TX and RX GTP transceivers share the same reference clock. However, RXUSRCLK and RXUSRCLK2 are not generated from TXREFCK, but rather from RXRECCLK, the recovered clock from the incoming data stream on the deskew channel. RXRECCLK is one-eighth the line rate, which makes it the frequency required for RXUSRCLK. RXUSRCLK2 is generated by dividing RXRECCLK by two in a DCM. Framing and data deskewing logic is driven by RXUSRCLK2. Although it is not shown in Figure 4, RXDATA15 also feeds into its own `sfi5_rx_data_sync` module. Only the logic in the `sfi5_clock_and_reset_rx` module is driven by TXREFCK because of the requirement that the clock be independent of the reset sequence.





X871\_04\_020708

Figure 4: SFI-5 Receiver Clocking

## SFI-5 Transmitter

The SFI-5 transmitter takes 16 inputs of 16 bits each (256 bits total) and stripes each 16-bit word across all 16 GTP transceivers. Therefore, the 16-bit word that each GTP transceiver transmits has one bit from each of the 16 inputs. The SFI-5 receiver in this reference design “unstripes” the data to return it to the normal order. It is important that the transmitter and receiver are complements of each other. If striping is not desired in either the TX or RX of this reference design, the assignment statements that perform the striping in the `sfi5_if_v5_16bit` module can be replaced by simple one-to-one assignments.

After the data inputs have been striped, they form 16 new words, each of which is transmitted by one of the 16 GTP transceivers. The 16 words arrive at the inputs of the GTP transceivers at the same time, but over the course of the transmission process, some words can arrive at the receiver sooner or later than others. There is no framing information in the data channels



themselves. Thus, there must be a way to record the proper order of the data words, such that the receiver can later use that recorded order to realign the data words. This is done by recording fragments of the data words onto the deskew channel before transmission when the order is still known to be correct. The deskew channel content is strictly defined by the OIF specification. The deskew frame is shown in Figure 5.

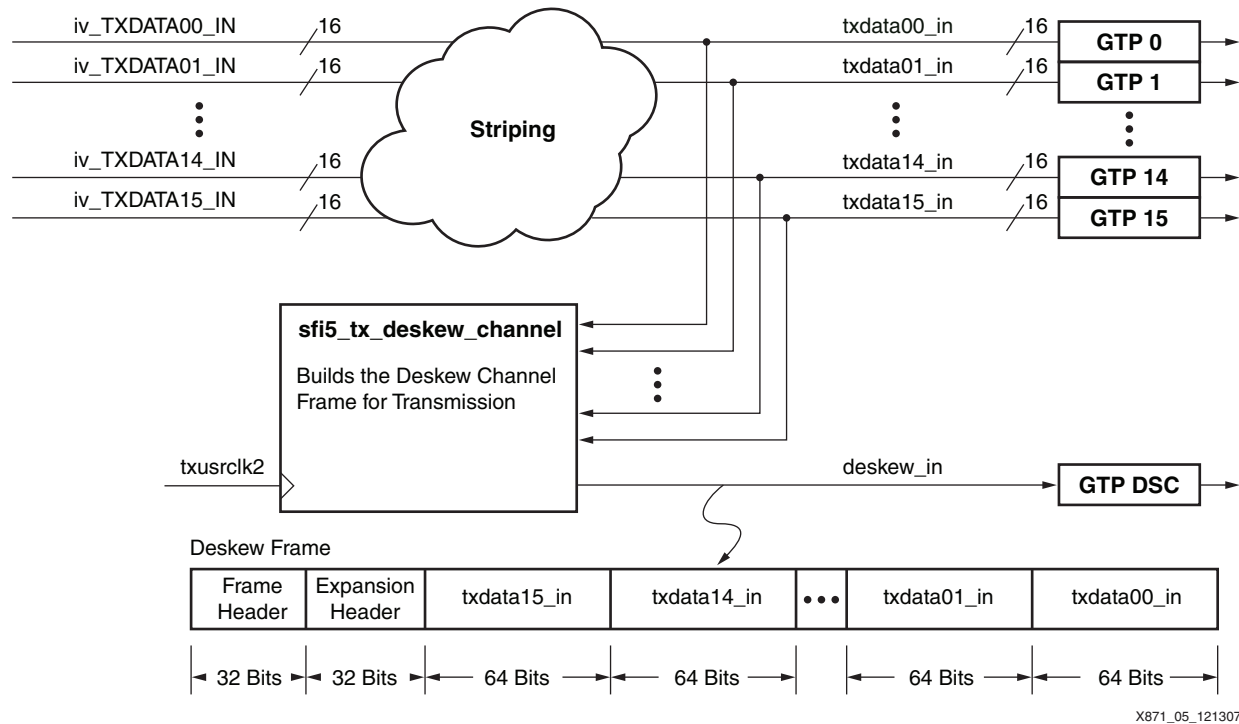


Figure 5: SFI-5 Transmitter Datapath and Deskew Channel Generation

The state machine that generates the deskew frame is shown in Figure 6. In normal operation, the state machine has no input dependencies. Each state represents a different setting in a large multiplexer in the datapath that selects from one of the 16 data channels (or static frame bits). The state machine is a loop that has no exit condition except for reset, and each iteration of the loop generates one frame. Two input conditions are provided for diagnostic purposes: frame error insertion and data error insertion. When `i_INSERT_FRAME_ERROR` is asserted, the state machine replaces the correct frame (F6F6 2828) with an incorrect frame (F6F6 2928). When `i_INSERT_DATA_ERROR` is asserted, the state machine inverts the first 16 bits of the 64-bit fragment of data channel 15. The names of the states in Figure 6 are taken directly from the Verilog module. The states Data00, Data01, etc., correspond to the transmission of txdata15\_in, txdata14\_in, etc. The names of the states are in the reverse order of the data channels that are copied to the DSC channel in those states.



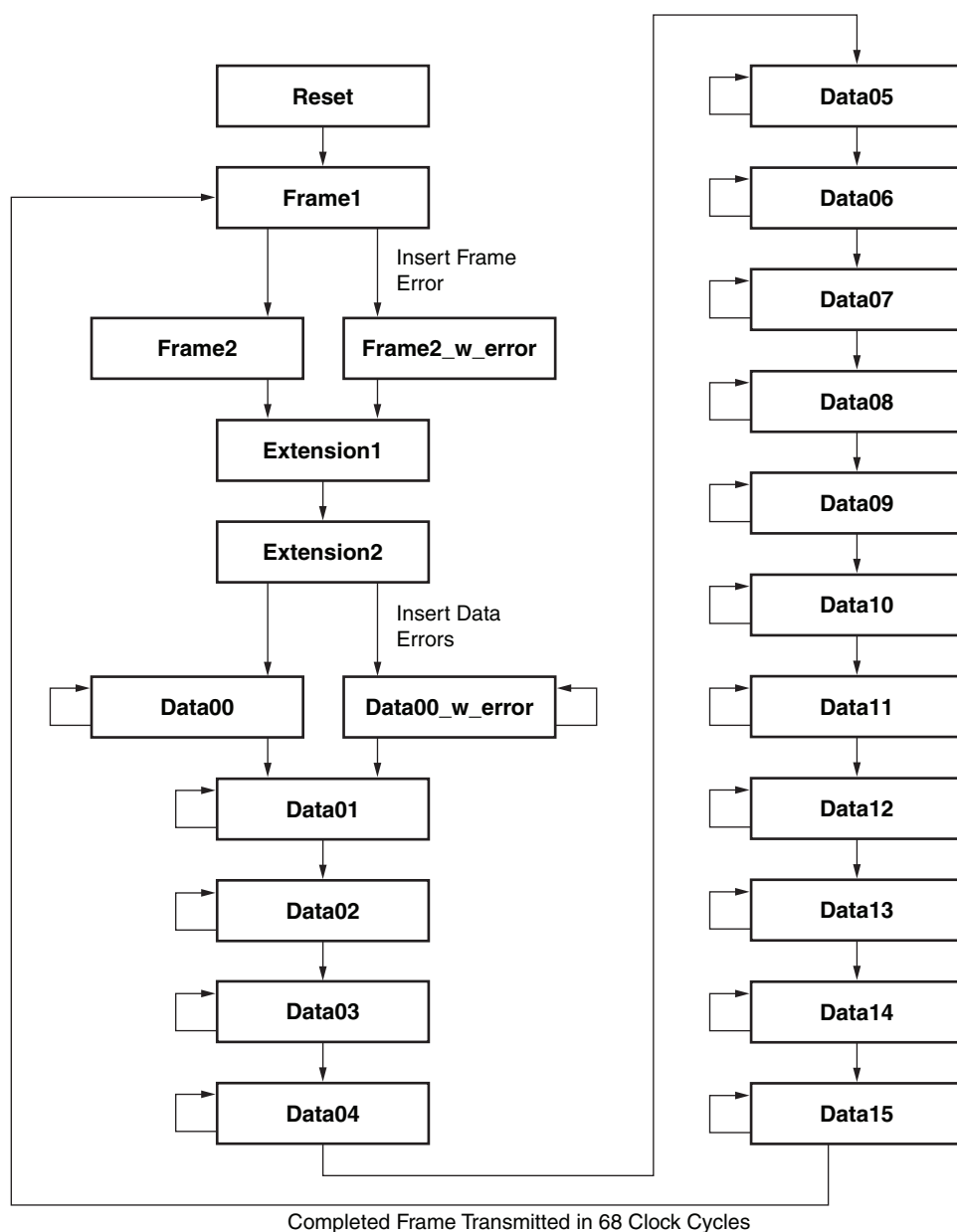


Figure 6: Deskew Frame Generation State Machine

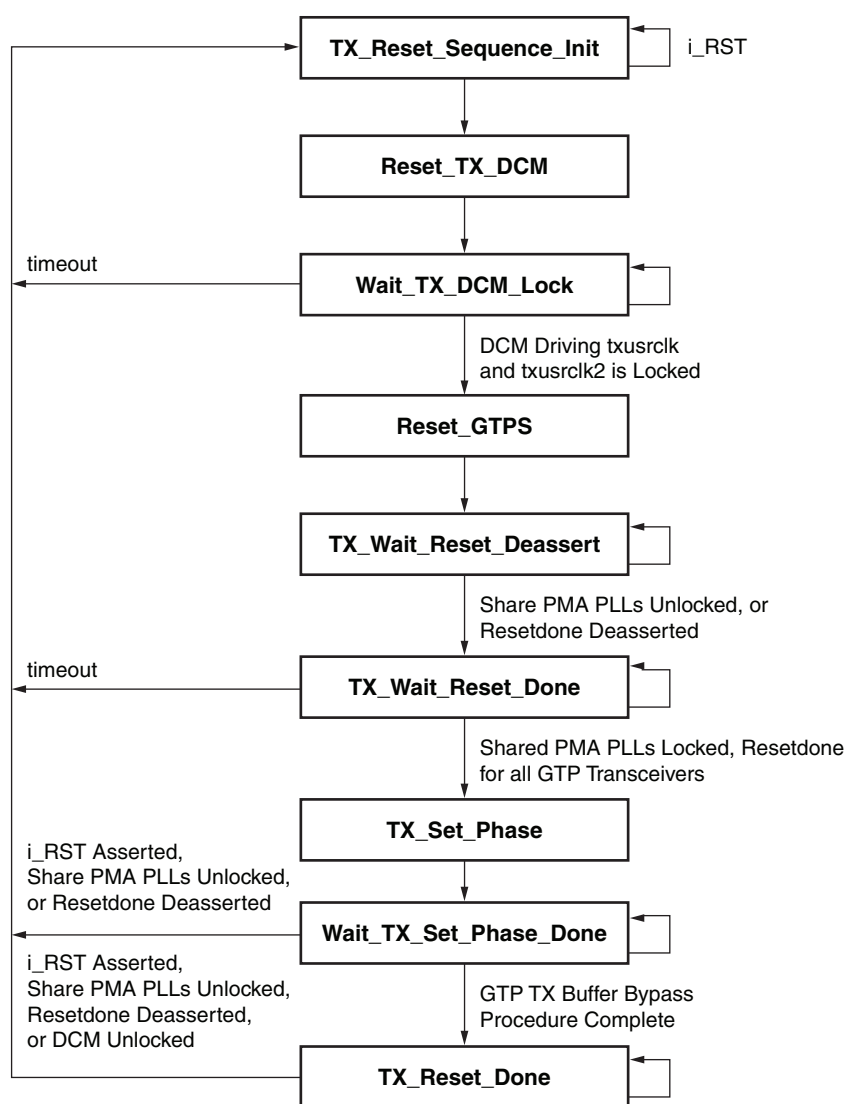
The initialization sequence of the transmitter is controlled by the state machine shown in Figure 7. The TX Initialization sequence consists of three main steps:

1. Reset the DCM that generates TXUSRCLK and TXUSRCLK2 and wait for the DCM to lock before proceeding to the next step.
2. Reset all 17 GTP transceivers by asserting the GTPRESET input of all GTP\_DUAL tiles. This initiates the full reset sequence of all circuitry in the GTP\_DUAL tiles. The TX\_Wait\_Reset\_Deassert state is included in the state machine because the RESETDONE ports do not deassert immediately when GTPRESET is asserted. Without the TX\_Wait\_Reset\_Deassert state, the state machine prematurely continues to the next step without waiting for the reset sequence to complete.
3. Initialize the TX phase-alignment circuit, which bypasses the TX buffer and guarantees low output skew between GTP transceivers. This procedure follows the instructions for bypassing the buffer given in the *Virtex-5 FPGA RocketIO GTP Transceiver User Guide*.



After these three steps are complete, the state machine rests in the TX\_Reset\_Done state unless another reset sequence is initiated, the shared PMA PLLs unlock, the RESETDONE output of any GTP transceiver is deasserted, or the DCM unlocks. If the i\_RST input is manually asserted to the SFI-5 interface, the reset sequence starts over regardless of the current state of the state machine.

The Wait\_TX\_DCM\_Lock and TX\_Wait\_Reset\_Done states have an additional exit condition of TIMEOUT. If the DCM is not locked after one million TXREFCK cycles (~6 ms), the reset sequence starts over. The reset sequence also starts over if the RESETDONE outputs of the GTP transceivers are not asserted after one million TXREFCK cycles.



X871\_06\_121407

Figure 7: Initialization Sequence of SFI-5 Transmitter



## SFI-5 Receiver

The SFI-5 receiver recovers the data from the 16 GTP transceiver inputs and presents 256 bits of data at the RXUSRCLK2 rate to the system side of the interface. The receiver must adjust the delay of each of the incoming datapaths to guarantee alignment between channels. The SFI-5 receiver datapath is shown in Figure 8.

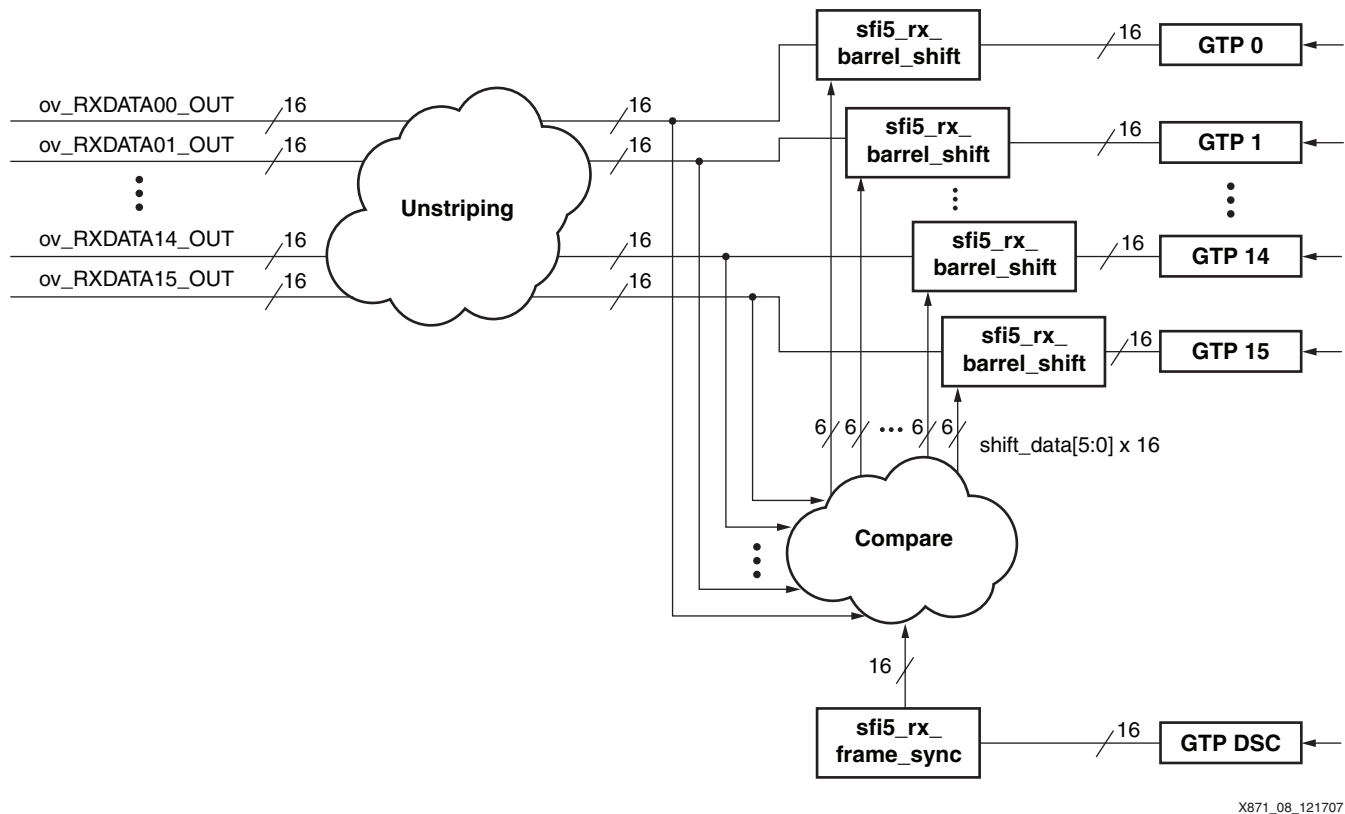


Figure 8: SFI-5 Receiver Datapath

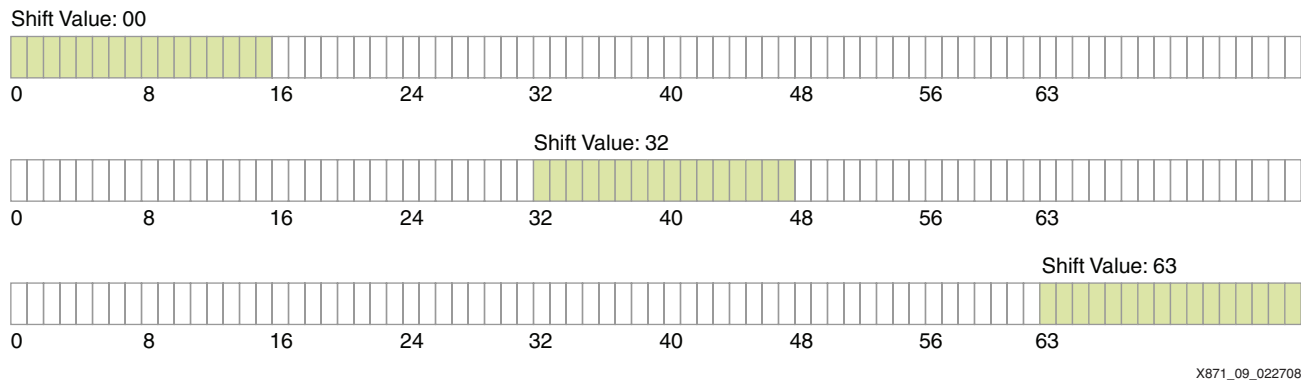
Each data channel has a 79-bit barrel shifter in its path that allows a state machine to select between 63 different delayed versions of the 16-bit output ( $63 + 16 = 79$ ) of the GTP transceiver. A barrel shifter in the deskew channel datapath facilitates the framing process. The 256 data bits are divided into 16 words of 16 bits each.

The receiver assumes that the data has been striped across the 16 GTP transceivers, so the words are unstriped before being presented to the system. It is important that the transmitter and receiver are complements of each other. If striping is not desired in either the TX or RX of this reference design, the assignment statements that perform the striping in the `sfi5_if_v5_16bit` module can be replaced by simple one-to-one assignments.

### Barrel Shifter

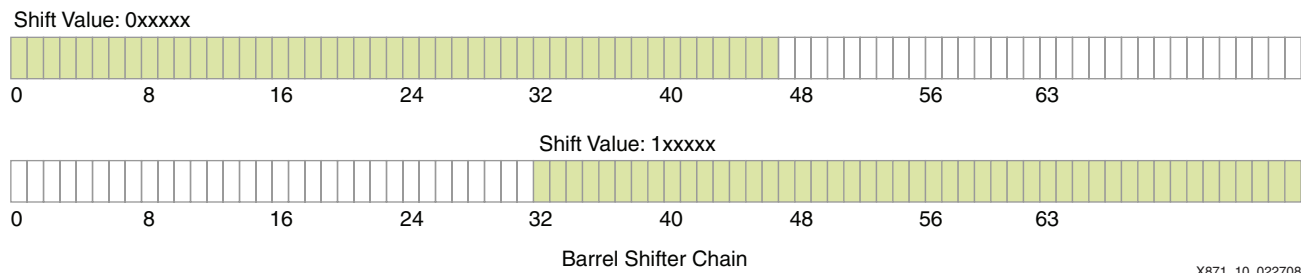
The barrel shifter is the fundamental building block of the receiver. All datapaths and the deskew channel path pass through a 79-bit barrel shifter like the one shown in Figure 9. With 79 bits, the barrel shifter can add 0 to 63 bit times of delay to a 16-bit word. Sixteen new bits are added to the left of the chain on each RXUSRCLK2 cycle, and all other bits in the chain are rotated by 16 to the right. The 16 bits farthest to the right are lost when new data is added. A shift value setting of 00 selects the most recent data bits for the output of the barrel shifter.





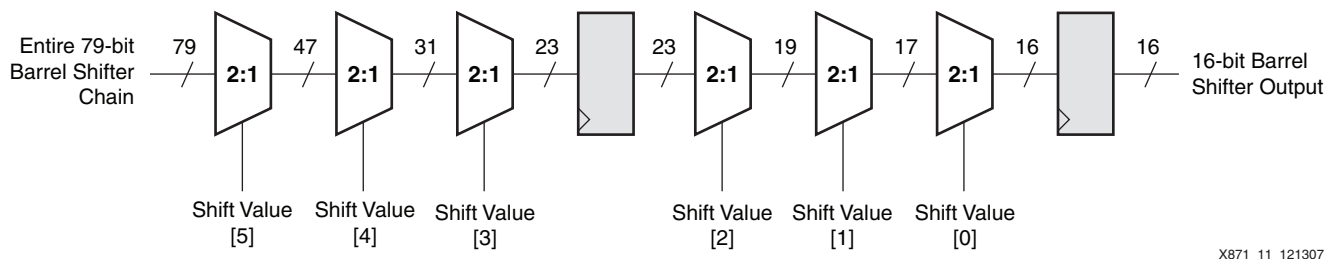
**Figure 9: 79-bit Barrel Shifter Chain**

The shift value is a 6-bit control signal ( $2^6 = 64$ ). There are six levels of 2-to-1 MUXs connected in series that reduce the data selection from 79 bits (the full barrel shifter) to 16 bits (the desired output width). Each bit of the shift value is a select line to one level of 2-to-1 MUXs. The most significant shift value bit must reduce the entire chain from 79 bits ( $2^6 - 1 + 16$ ) to 47 bits ( $2^5 - 1 + 16$ ). This first reduction corresponds to MUX level 1, shown in Figure 10, and is accomplished using 47 2-to-1 MUXs. If the most significant shift value bit is 0, the 47 bits selected are [46:0]. If the most significant shift value bit is 1, the 47 bits selected are [78:32].



**Figure 10: Barrel Shifter MUX Level 1 Reduction**

All six levels of MUXs and their corresponding reductions are shown in Figure 11. To ease timing closure, registers are added at the outputs of MUX levels 3 and 6.



**Figure 11: Barrel Shifter Output Selection Logic**

## Deskew Frame Synchronization

Before the receiver can begin deskewing the data channels, it must first synchronize to the incoming deskew channel. To do this, a state machine in the receiver must scan the incoming deskew channel data in search of the frame header (0xF6F62828). The state machine must monitor the incoming deskew channel data for at least 68 cycles, which is the size of the deskew frame (actual scanning time is 128 cycles). If the frame is not found, it is possible that the deskew channel is not aligned to the frame boundary. After scanning for 128 cycles without



finding the frame, the state machine adds one bit time of delay by incrementing the shift value of the barrel shifter in the deskew channel path. This process never repeats more than 16 times because the size of the barrel shifter output is 16 bits, and there must be one correct alignment of the frame header (0xF6F6) in any 16-bit window. The deskew channel frame synchronization state machine is shown in Figure 12.

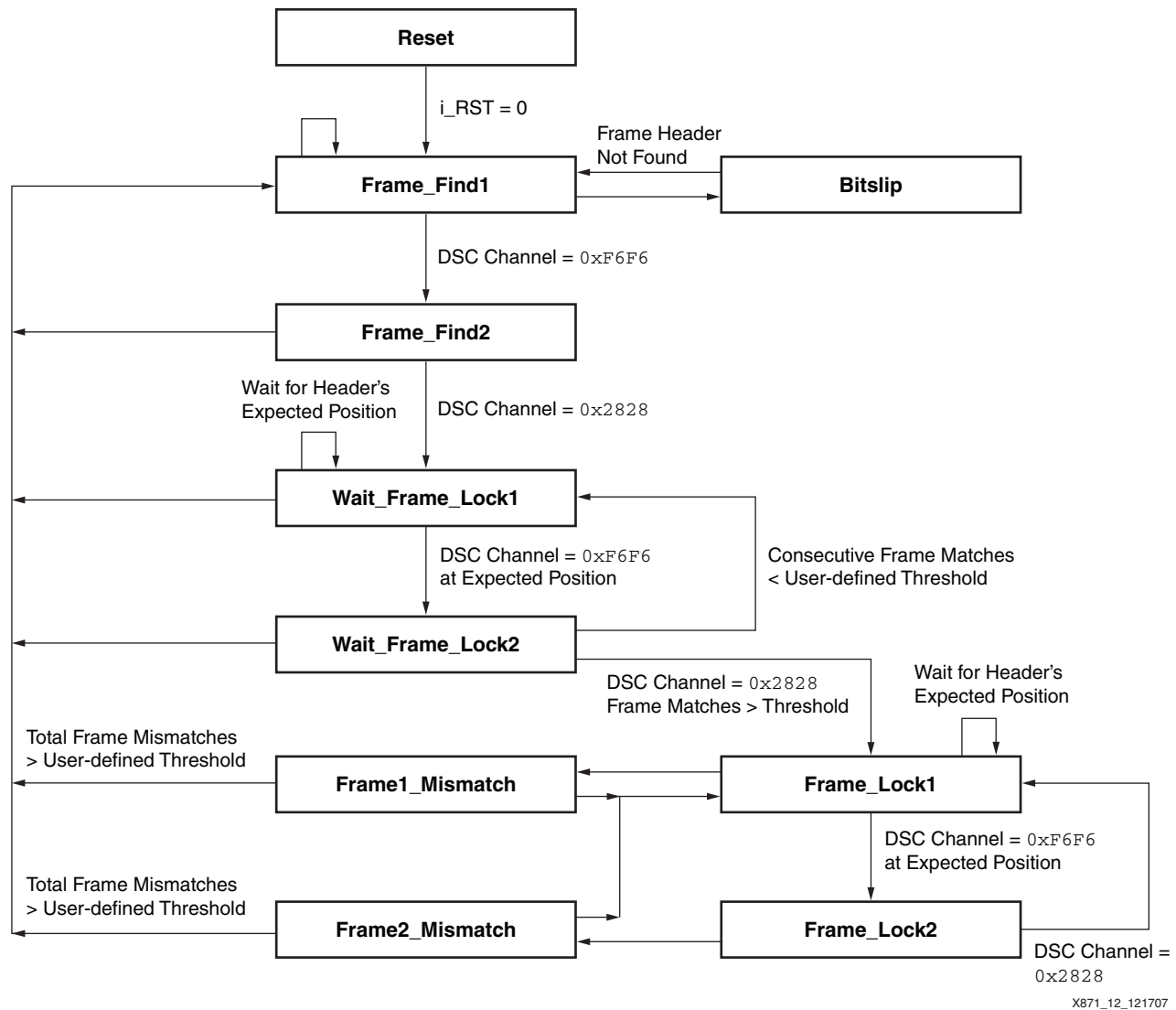


Figure 12: Deskew Channel Frame Synchronization State Machine

After the frame synchronization algorithm is complete, the barrel shifter setting of the deskew channel becomes an anchor for all the data channel deskew logic. Each data channel's skew is defined relative to the deskew channel. If the deskew channel barrel shifter is set to 0, the data deskew algorithm can deskew data that is +63 UI later than the deskew channel, but only 0 UI before the deskew channel. This asymmetric deskew capability is not desirable because data channels can be slower or faster than the deskew channel, and both contingencies must be addressed.

Ideally, the barrel shifter setting of the deskew channel is 32 because that setting results in perfectly symmetric deskew capability ( $\pm 32$  UI). Unfortunately, the deskew channel setting cannot be simply set to a specific value. It is determined by the frame synchronization algorithm. The frame synchronization algorithm can find the correct alignment setting in a search field of any 16 barrel shifter settings. For this reason, the deskew channel barrel shifter is initialized to 24 and increments only as high as 39 (search field = 16). The frame



synchronization algorithm finds the setting with the correct alignment somewhere between 24 and 39. Allowing for the worst cases of 24 and 39, there is still a minimum of  $\pm 24$  UI of skew compensation capability (both  $63 - 39$  and  $24 - 0$  equal 24). The synchronization process consists of these steps:

1. Find the frame header in the incoming deskew channel, adjusting the barrel shifter setting as necessary.
2. After the first frame is detected, keep a running count of frames detected. As soon as the user-defined threshold (`i_FRAMES2LOCK`) for declaring frame lock is reached, `RXLOF` is driven Low to indicate that the deskew channel is synchronized.
3. When the framer is in the locked state, it continuously checks every frame. If an error is detected in any part of the frame, a mismatch is recorded. As long as the total number of frame mismatches does not exceed the user-defined threshold (`i_FRAMES2UNLOCK`), the state machine returns to the locked state. When the threshold is exceeded, `RXLOF` is asserted and the algorithm starts over from the beginning.

## Data Channel Deskew

After the deskew channel barrel shifter setting has been established by the frame synchronization algorithm, each data channel can be compared to the deskew channel. The barrel shifter setting of each data channel can be adjusted until the data content of that channel matches the 64-bit data fragment contained in the deskew frame. Each data channel is initialized with a barrel shifter setting of 0, and every setting from 0 to 63 is searched to find the data that matches the deskew channel. After all 16 channels complete this process, each channel finishes with a unique barrel shifter setting that is both a measure of skew relative to the deskew channel and a measure of skew relative to the other data channels. The smallest barrel shifter setting of the 17 channels subtracted from the largest barrel shifter setting yields the peak skew of the entire interface from TX to RX. [Figure 13](#) shows actual measurements of skew compensation collected on a 48 Gb/s interface (3.0 Gb/s x 16 channels).



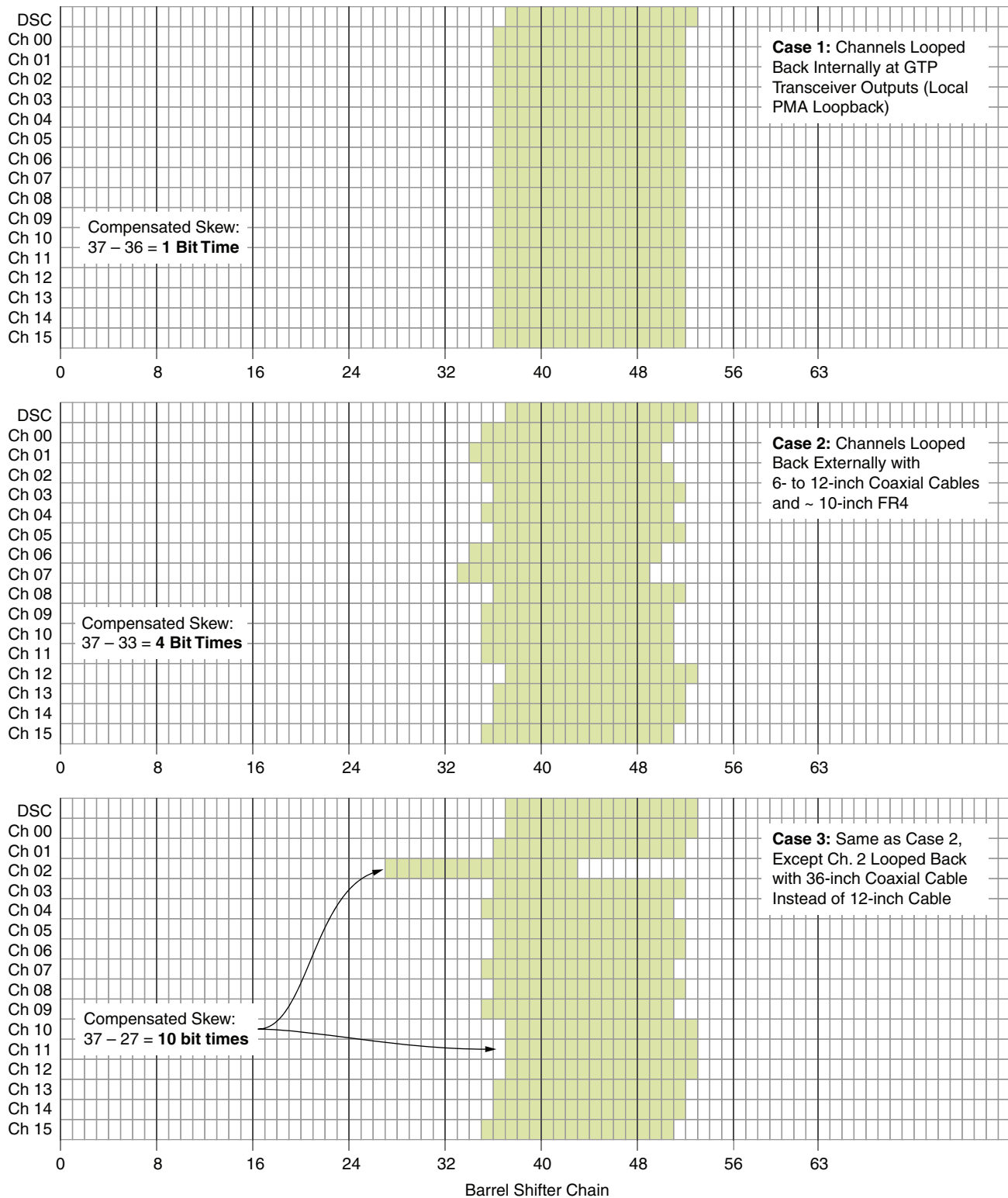


Figure 13: Barrel Shifter Selections (3.0 Gb/s x 16 Channels, Serial Number 029)

Every data channel and the deskew channel have uniquely determined barrel shifter settings, each creating a picture of the skew between the channels. To illustrate this, each of the three cases introduces a different amount of skew on the link. In case 1, the transmitter is looped back to the receiver internally (minimal skew). The results show that every data channel independently chooses the same barrel shifter setting of 36, and the total compensated skew is



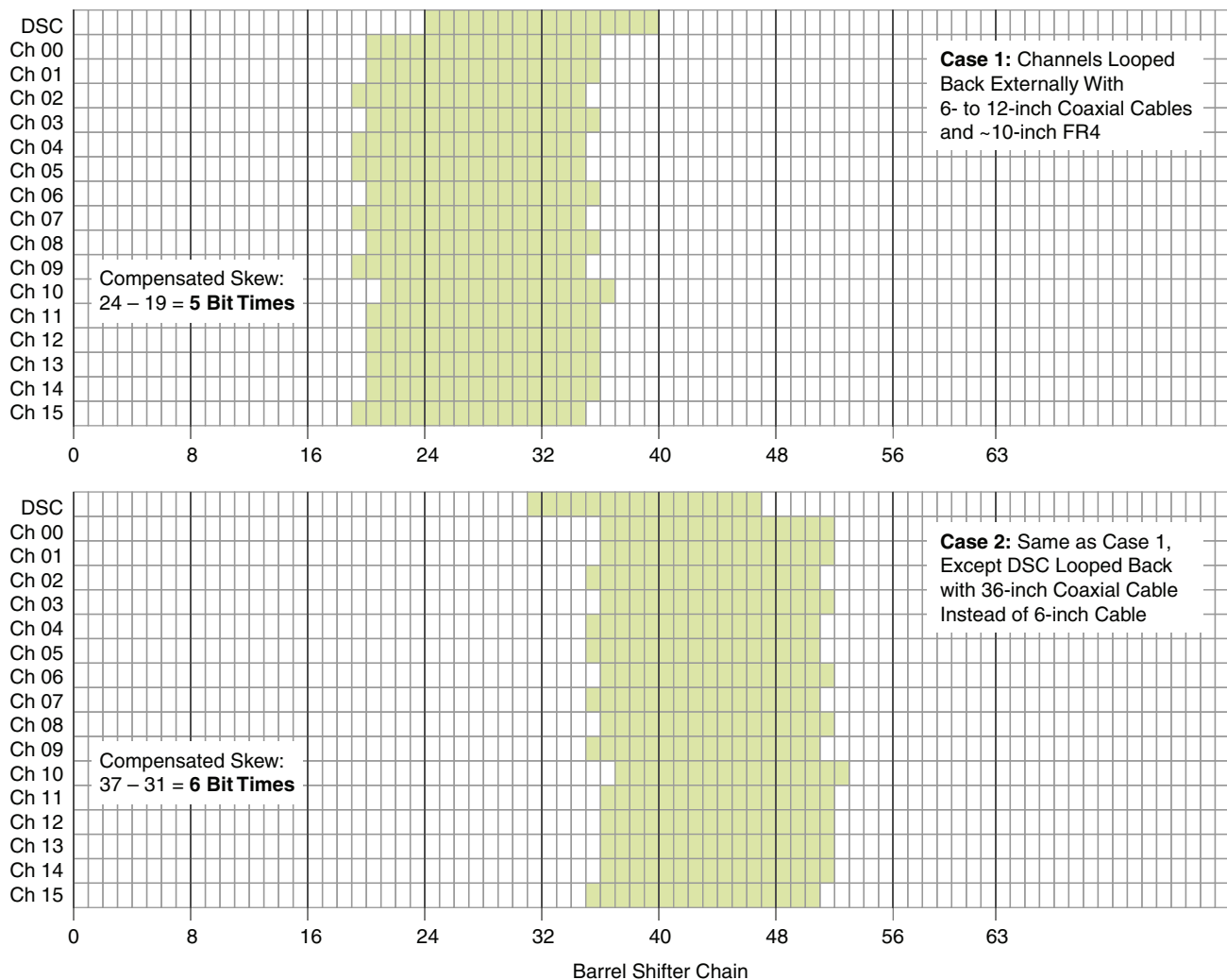
only 1 UI. In case 2, the channels are looped back externally, some with 6-inch cables and some with 12-inch cables. The data channels now have different barrel shifter settings because the datapaths have differences in latency due to package routing, board routing, and cable lengths. The compensated skew is 4 UI in case 2. To accentuate the skew compensation even further, the cable on channel 2 is replaced by a cable that is 24 inches longer (36 inches total) in case 3. This produces a very visible effect on channel 2, which now has a significantly different barrel shifter setting. From case 2 to case 3, the channel 2 setting changes from 35 to 27. The setting decreases (move towards the left) because the most recent data is added to the barrel shifter on the left. Because channel 2 takes longer to get to the receiver in case 3, the state machine has to choose a more recent barrel shifter position from which to read the data (27 is more recent than 35). As for the actual magnitude of the change from case 2 to case 3, the theoretical calculation matches the measurement well:

Measured skew difference:  $35 - 27 = 8$  UI.

Added delay: (24 inches of extra cable)  $\times$  (113 ps/inch signal velocity in coaxial cable) = 2.712 ns.

Theoretical skew difference: (2.712 ns added delay)  $\div$  (0.333 ns period of 3.0 Gb/s) = 8.14 UI.

Figure 14 shows another set of actual measurements of skew compensation. It is for the same device as in Figure 13, but the data rate is now 40 Gb/s (2.5 Gb/s  $\times$  16 channels).



X871\_14\_022708

Figure 14: Barrel Shifter Selections (2.5 Gb/s x 16 Channels, Serial Number 029)

In case 1, all channels are looped back externally with 6- to 12-inch cables. In case 2, the deskew channel cable is replaced by a 36-inch cable. The result illustrates how the deskew



channel barrel shifter works. In case 1, the deskew channel is set to 24. After adding 30 inches of cable, the expectation is that the setting would decrease (move to the left) according to this calculation:

Added delay: (30 inches of extra cable)  $\times$  (113 ps/inch signal velocity in coaxial cable) = 3.390 ns.

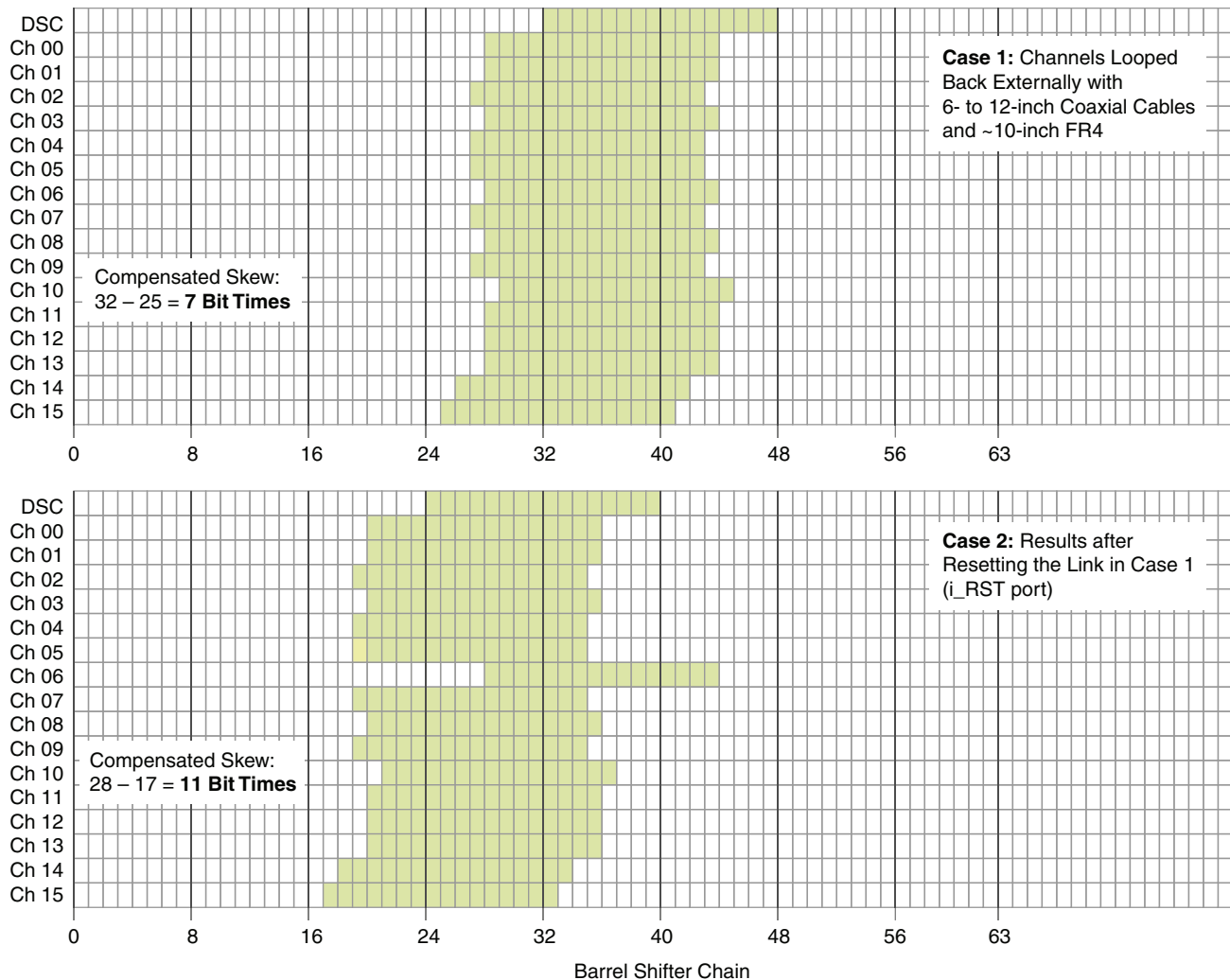
Theoretical skew difference: (3.390 ns added delay)  $\div$  (0.400 ns period of 2.5 Gb/s) = 8.5 UI.

However, in case 2, the deskew channel setting actually increases (moves to the right). The explanation for this is discussed in [“Deskew Frame Synchronization,” page 14](#). The deskew channel is confined to settings from 24 to 39, so instead of varying the setting in case 1 from 24 to 15, the frame synchronization algorithm chooses a higher alignment setting. In case 1, there is, by definition, another alignment setting at  $24 + 16 = 40$ . When 30 inches of cable is added in case 2, the deskew channel setting is 31, which is 9 UI to the left of 40, matching the theoretical calculation exactly. The deskew frame setting in case 2 is a full 16 UI behind the original frame setting in case 1. Thus, the data channels in case 2 are all exactly 16 bits behind the data channels in case 1.

There is another source of skew that is not seen in the examples of [Figure 13](#) and [Figure 14](#). Unlike the transmitter, the receiver does not bypass the elastic FIFO in the GTP RX path. The phase-alignment feature that allows the FIFO to be bypassed is not allowed when an 8-bit datapath is used in the GTP transceivers of an LXT device. With the FIFO in the path, no phase relationship exists between RXRECCLK (FIFO write clock) and RXUSRCLK (FIFO read clock). This means that there is no phase relationship between the RXRECCLKs of the different data channels.

Depending on the relationship of RXRECCLK and RXUSRCLK for each GTP transceiver,  $\pm 8$  UI of skew (one RXUSRCLK cycle) can be introduced between the different channels of the GTP transceiver simply by resetting the link without changing any of the physical connections. This behavior is shown in [Figure 15](#).





**Figure 15: Barrel Shifter Selections (2.5 Gb/s x 16 Channels, Serial Number 027)**

In cases 1 and 2, the physical connections are identical. The link reset sequence was executed once and the data thus collected is shown in case 1. The reset sequence was executed again and the data thus collected is shown in case 2.

The results show that all channels in case 2 are moved to the left by exactly 8 bit times, except for channel 6, which is the same value that it was in case 1. The FIFOs of all GTP transceivers in case 2 (except channel 6) present data to the fabric 8 bit times (1 RXUSRCLK cycle) later than case 1, forcing the frame synchronization algorithm to frame the deskew channel 8 bits earlier in the barrel shifter chain. This advances all the data channel settings in the barrel shifter chain by 8 bit times. The phase relationship of RXRECCLK and RXUSRCLK in channel 6 happens to be such that the data is read to the outputs one RXUSRCLK cycle earlier than all the other GTP transceivers. This means that the channel 6 setting of the barrel shifter is larger than the other channels because it arrives earlier and has more time to propagate deeper into the chain. The FIFO skew can be introduced between any GTP transceivers and can change repeatedly after several resets. However, because SFI-5 is a synchronous system (same FIFO read and write clock frequency), the upper bound of the skew that the FIFOs can introduce is  $\pm 8$  UI.

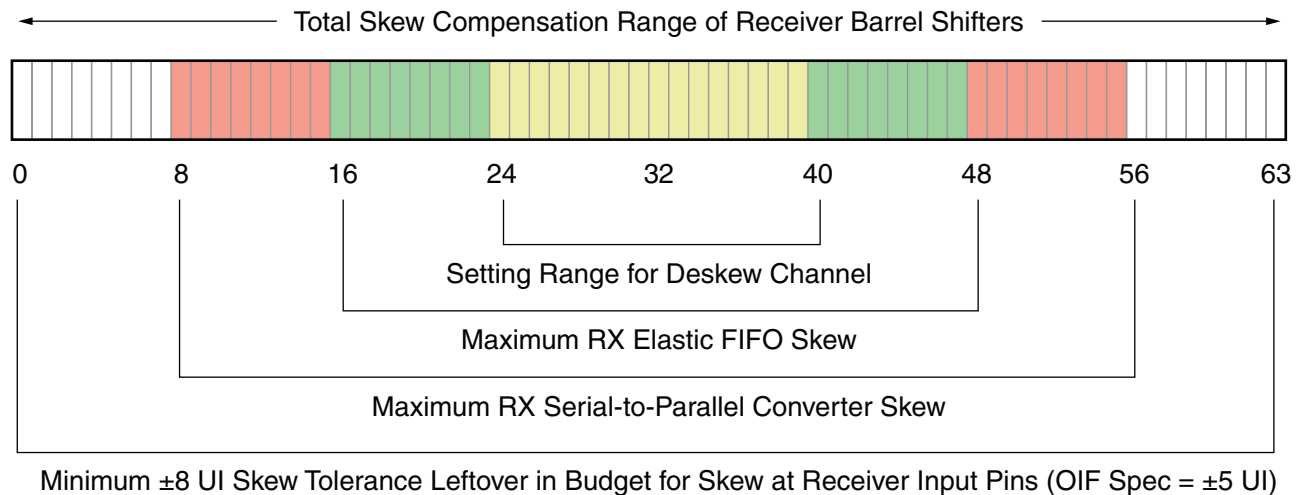
Another source of skew in the receiver is the serial-to-parallel converter. Again, due to the fact that the RXRECCLKs of the different GTP transceiver channels have no fixed phase



relationship to one another, the serial-to-parallel conversion stage can introduce anywhere between 1 to 8 bit times of skew (for an 8-bit datapath).

The amount of skew that can be compensated by the receiver must be quantified. How much skew is allowed at the input pins of the receiver? With a 63-bit barrel shifter, the absolute maximum skew compensation of the receiver is  $\pm 32$  UI. However, some of that margin is lost to the uncertainty of the deskew channel setting, which is not always set to the ideal setting of 32. It can be set as low as 24 or as high as 39. That reduces the ideal margin from  $\pm 32$  UI to  $\pm 24$  UI.

The skew budget is further reduced by the receive path, which contributes a maximum of  $\pm 8$  UI for the serial-to-parallel conversion and a maximum of  $\pm 8$  UI for the RX elastic FIFO. That leaves an absolute minimum of  $\pm 8$  UI leftover for the skew budget at the input pins of the receiver, as shown in Figure 16. The OIF SFI-5 specification for the minimum jitter that the receiver must tolerate is  $\pm 5$  UI of skew.

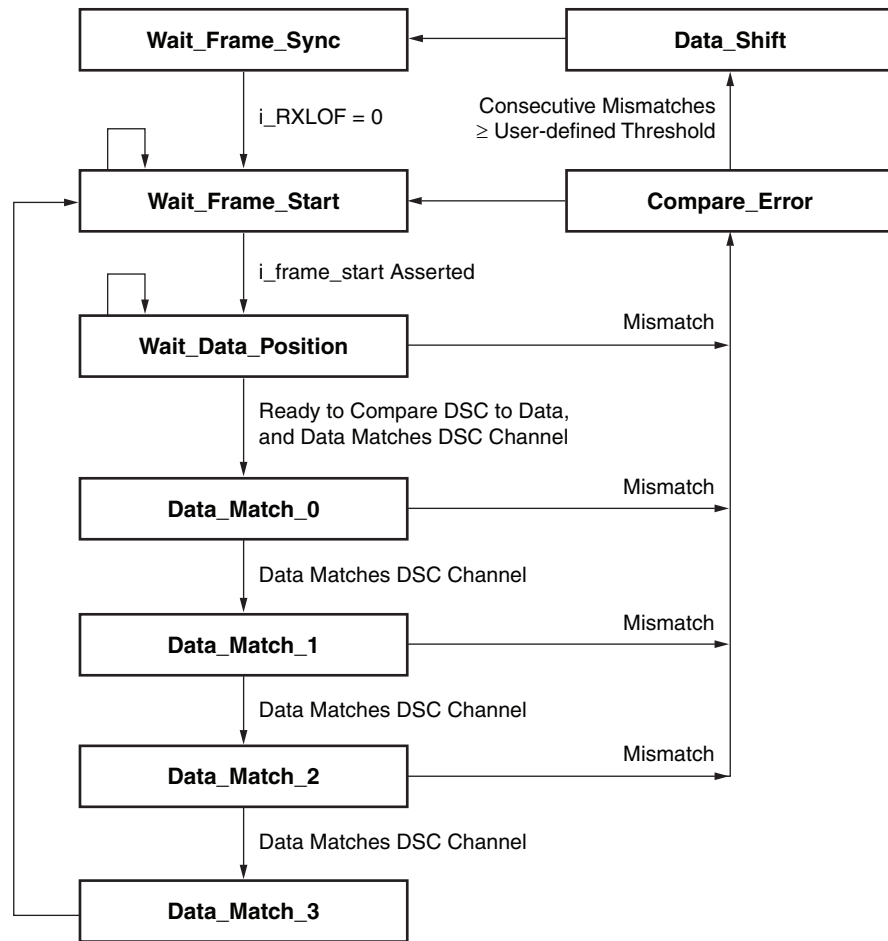


X871\_16\_121707

Figure 16: Receiver Skew Budget

The algorithm that deskews each data channel by controlling the barrel shifter setting and comparing the data to the deskew channel is shown in Figure 17.





X871\_17\_121307

**Figure 17: Data Channel Deskew State Machine**

The data channel deskew state machine is contained in the `sfi5_rx_data_sync` module. The state machine does not attempt alignment until the frame synchronization algorithm reports that it has synchronized to the deskew channel (RXLOF deasserted). After the deskew channel is framed, the state machine waits for the beginning of the frame. The state machine for each channel then waits an additional number of cycles to arrive at the specific data fragment in the deskew frame for that channel. At this point, the data channel and deskew channel content are compared for four consecutive cycles (64-bit data fragment in deskew channel). All four of the comparisons match if the data channel is correctly aligned. If there is a mismatch in one of the four comparisons, a counter records the mismatch. If the number of consecutive mismatches reaches the user-defined threshold `iv_MISMATCHES_2_UNLOCK`, the barrel shifter setting of that data channel is incremented and the algorithm repeats the comparison process. The state machine starts at barrel shifter setting 0 and increments all the way to 63. If the receiver input skew specification is satisfied, the algorithm finds a match. If the algorithm does not find a match, it searches from 0 to 63 indefinitely. However, this indefinite search is interrupted when the receiver times out and reinitializes the whole link. This time-out functionality is part of the `sfi5_clock_and_reset_rx` module.



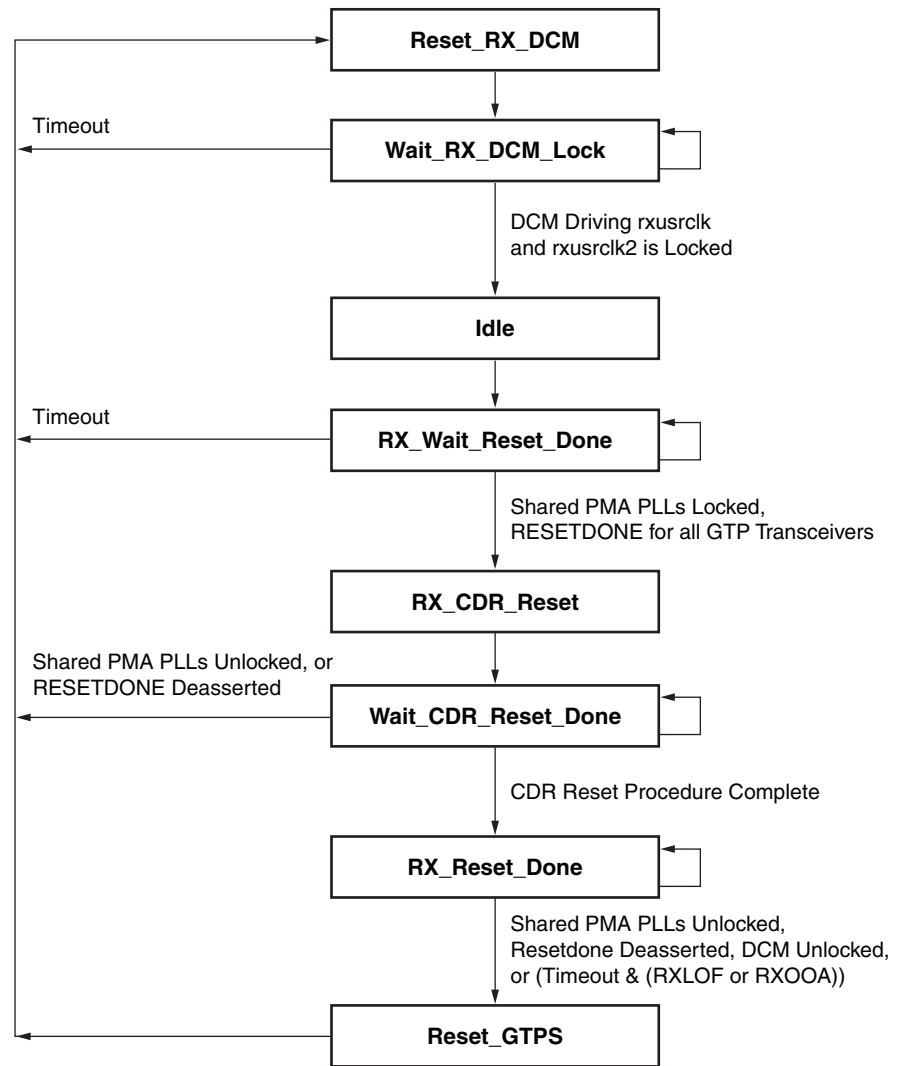
## Receiver Initialization

The receiver initialization process consists of three steps:

1. Reset the DCM that produces RXUSRCLK and RXUSRCLK2 from RXRECCLK, and wait for it to lock.
2. Wait for the GTP transceivers to finish their reset sequence and for the shared PMA PLLs to lock. The state machine itself does not initiate the reset of the GTP transceiver. Each GTP\_DUAL tile is shared by the TX and RX interface. Thus, the TX interface initialization is responsible for initiating the GTP reset. The RX state machine waits for the GTP reset sequence to complete.
3. Initiate an electric idle reset of the GTP transceivers. This is done at the end of the RX initialization sequence in case transients during the startup sequence temporarily put the clock data recovery (CDR) in the electric idle state.

Steps 1 and 2 each have time-out counters in case the DCM never locks, the GTP transceivers never come out of reset, or the shared PMA PLLs do not lock. After timing out, the algorithm restarts the RX initialization. The time-out allows the link to recover automatically when the physical link is broken for some period of time due to disconnection, one side of the link in a power-down state, and so on. After the link is restored, the receiver restarts the initialization process, and the link recovers without user intervention or manual resets. The receiver initialization process is shown in [Figure 18](#).





X871\_18\_121407

**Figure 18: Initialization Sequence of SFI-5 Receiver**

After the three steps are complete, the state machine remains in the **RX\_Reset\_Done** state. However, if the shared PMA PLLs unlock, if the RESETDONE output of any GTP transceiver is deasserted, or if the DCM unlocks, the state machine resets all GTP transceivers and the initialization process starts over. The **RX\_Reset\_Done** state also has a time-out condition. If the frame synchronization and data deskew state machines cannot achieve frame alignment (RXLOF) or data alignment (RXOOA) after one million RXUSRCLK2 cycles (6 ms), the state machine declares an exceptional condition and restarts the initialization process after resetting the GTP transceivers. If it is not desired that the TX or RX interfaces reset themselves automatically (for troubleshooting purposes or otherwise), the TX\_Reset\_Done and RX\_Reset\_Done next-state logic must be modified to remain in those states unconditionally.



## SFI-5 Resource Utilization

The FPGA resources used by the entire stand-alone SFI-5 interface are shown in [Table 3](#).

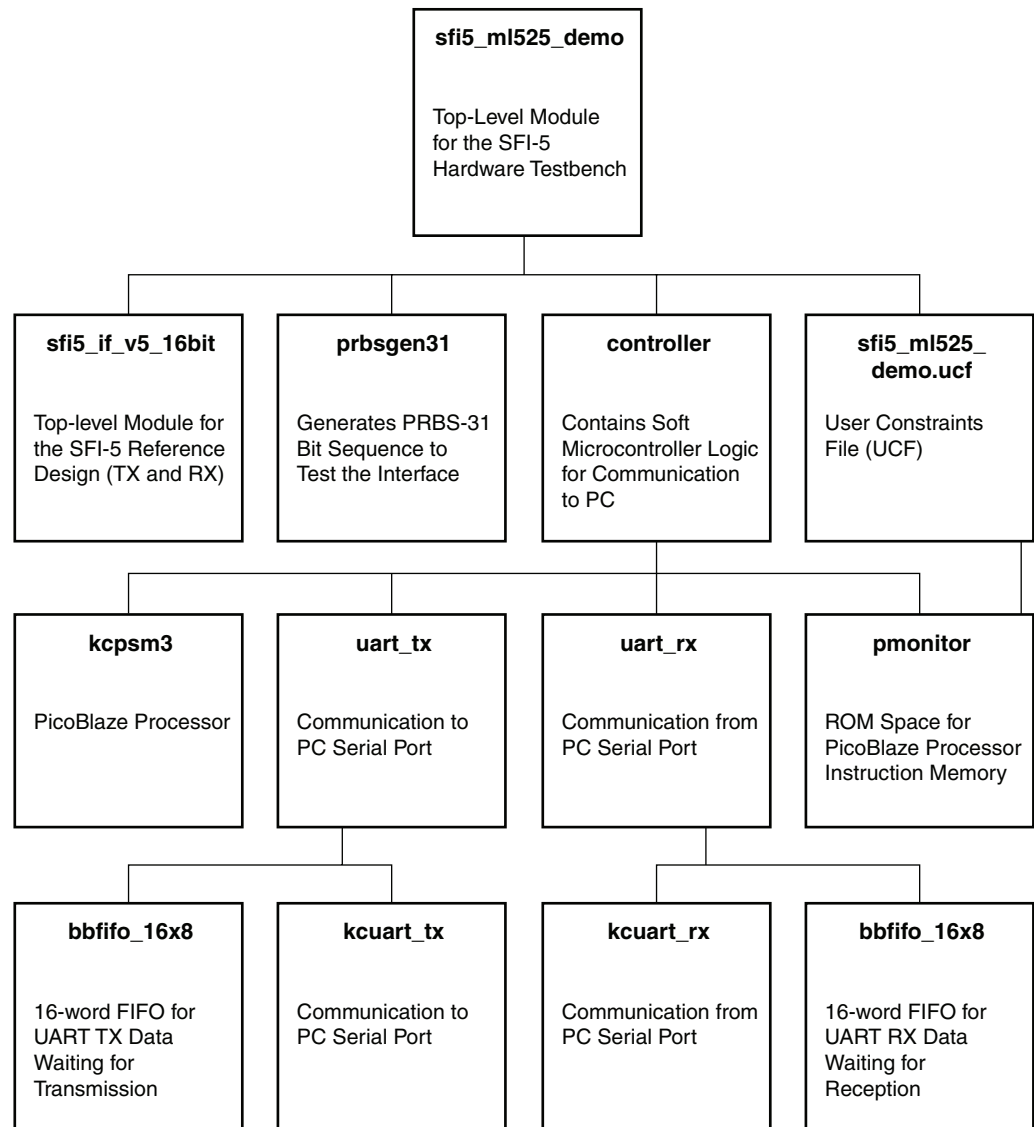
*Table 3: Resource Utilization*

Resource	Used	Percentage of Total	Comment
Slice	1,814	3	
Slice Register	3,233	1.6	
Slice LUT	3,869	1.9	
BUFG	6	19	txrefclk, txusrclk, txusrclk2, rxusrclk, rxusrclk2, rxrecclk
DCM	2	17	
GTP Transceiver	17	71	16 data channels + DSC channel
IOB	7	<1	TXREFCK (differential) TXREFCK_2 (differential) TXDCK (differential) RXS

## SFI-5 Hardware Testbench

The reference design is hardware tested on the ML525 development board ([UG225](#), *ML52x User Guide*). The device is an XC5VLX330T in an FF1738 package. To test the interface, a hardware testbench consisting of a PicoBlaze™ processor is wrapped around the interface. The processor reads link statistics from the SFI-5 interface, and allows control of the interface's optional settings from a HyperTerminal interface. The hierarchy of the hardware testbench is shown in [Figure 19](#).





X871\_19\_022108

**Figure 19: Hierarchy of HDL Modules Comprising SFI-5 Interface**

A pseudo-random bit sequence 31 (PRBS31) is generated by the testbench to simulate the complexity of user data. The PRBS31 pattern is sent across the SFI-5 link. The error checking mechanism on the receive side is a comparison of the data channels with the deskew channel, which is already done by the SFI-5 receiver. The HyperTerminal interface shows the mismatch counts of all 16 data channels as counted by the SFI-5 receiver.

The **uart\_tx** and **uart\_rx** modules communicate information to a PC from the PicoBlaze processor. The serial link (RS-232) operates at a baud rate of 38,400. The PicoBlaze processor can receive commands from the user via the HyperTerminal interface. The commands can have arguments of up to 8 bits expressed as hexadecimal values. When commands are written with arguments, they are interpreted as write commands. When commands are written without arguments, they are interpreted as read commands. Arguments are separated from the command by a single space. The supported user commands are shown in [Table 4](#).

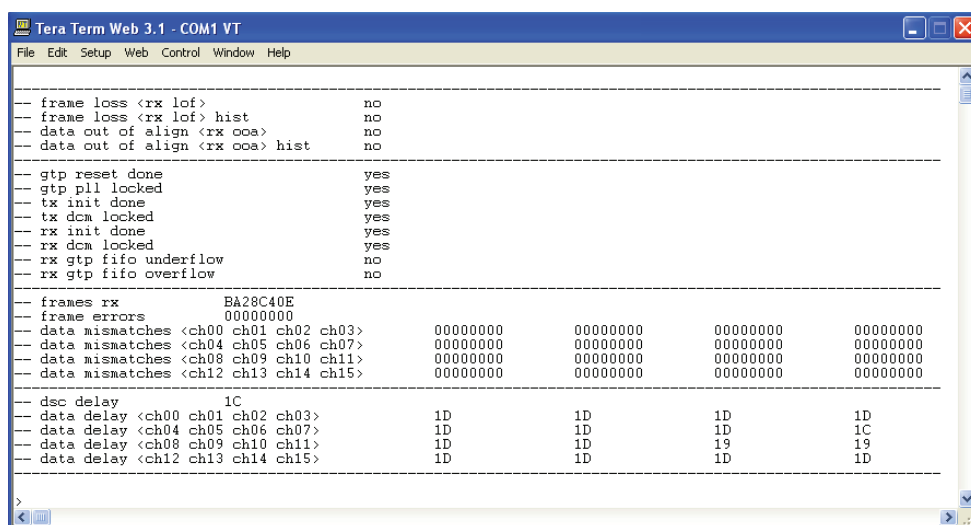


Table 4: User Commands for SFI-5 Interface

Command	Argument	Description
<enter>	None	Updates statistics on HyperTerminal display.
r	None	Asserts and deasserts i_RST input of SFI-5 interface, causing full interface reset.
l	<xx>	Loopback control: [2:0]: i_LOOPBACK. [7:3]: Not used.
l	None	Read back control setting for loopback.
c	None	Clear all error counters and alarms. Does not change any interface settings or interrupt the flow of data.
d	<xx>	Transmitter settings: [0]: i_TX_DRIVER_INHIBIT_STARTUP. [1]: i_TX_DRIVER_INHIBIT_MANUAL. [4:2]: i_TX_PREAMPHASIS. [7:5]: i_TX_DIFF_CTRL.
d	None	Read back transmitter control settings.
e	<xx>	Receiver settings: [0]: i_RX_EQUALIZATION_ENABLE. [2:1]: i_RX_EQUALIZATION_MIX. [3]: Not used. [7:4]: i_RX_EQUALIZATION_POLE.
e	None	Read back receiver control settings.
fl	<xx>	Frame lock threshold: [7:0]: i_FRAMES2LOCK
fl	None	Read back frame lock threshold.
fu	<xx>	Frame unlock threshold: [7:0]: i_FRAMES2UNLOCK
fu	None	Read back frame unlock threshold.
m	<xx>	Data synchronization loss threshold: [7:0]: iv_MISMATCHES_2_UNLOCK.
m	None	Read back data synchronization loss threshold.

The HyperTerminal interface is shown in [Figure 20](#). In the first section of the reported statistics, the two most important flags of the SFI-5 interface are shown: RXLOF and RXOOA. RXLOF indicates that the deskew channel is not frame locked. RXOOA indicates that the data channels are not correctly aligned to the deskew channel. Both RXLOF and RXOOA are reported as current values and historical values. The historical values capture an alarm condition and are not cleared until the user issues a *c* command.





```

Tera Term Web 3.1 - COM1 VT
File Edit Setup Web Control Window Help
-----
-- frame loss <rx lof>                no
-- frame loss <rx lof> hist            no
-- data out of align <rx ooa>          no
-- data out of align <rx ooa> hist     no
-----
-- gtp reset done                     yes
-- gtp pll locked                     yes
-- tx init done                       yes
-- tx dcm locked                      yes
-- rx init done                       yes
-- rx dcm locked                      yes
-- rx gtp fifo underflow              no
-- rx gtp fifo overflow               no
-----
-- frames rx                          BA28C40E
-- frame errors                       00000000
-- data mismatches <ch00 ch01 ch02 ch03> 00000000 00000000 00000000 00000000
-- data mismatches <ch04 ch05 ch06 ch07> 00000000 00000000 00000000 00000000
-- data mismatches <ch08 ch09 ch10 ch11> 00000000 00000000 00000000 00000000
-- data mismatches <ch12 ch13 ch14 ch15> 00000000 00000000 00000000 00000000
-----
-- dsc delay                          1C
-- data delay <ch00 ch01 ch02 ch03>      1D          1D          1D          1D
-- data delay <ch04 ch05 ch06 ch07>      1D          1D          1D          1C
-- data delay <ch08 ch09 ch10 ch11>      1D          1D          19          1C
-- data delay <ch12 ch13 ch14 ch15>      1D          1D          1D          1D
-----
>

```

X871\_20\_121407

**Figure 20: Command Line Interface and Current Link Statistics of SFI-5 Testbench**

The second section of the reported statistics shows eight flags that track the condition of the TX and RX interfaces. All of these flags are historical bits and can only be cleared by issuing a *c* command (see Table 4). The third section of the reported statistics shows the number of total frames received, the number of frames with errors in the frame header, and the number of mismatches between each channel and the corresponding deskew channel data fragment for that channel. This section is not a true error detector because it only tracks mismatches between the data and deskew channels. However, mismatches are a good measure of the robustness of the link. In the fourth section of the reported statistics, the current barrel shifter setting of each of the 16 data channels and the deskew channel is shown. The frame count, mismatch counts, and barrel shifter settings are expressed in hexadecimal format.

## Characterization

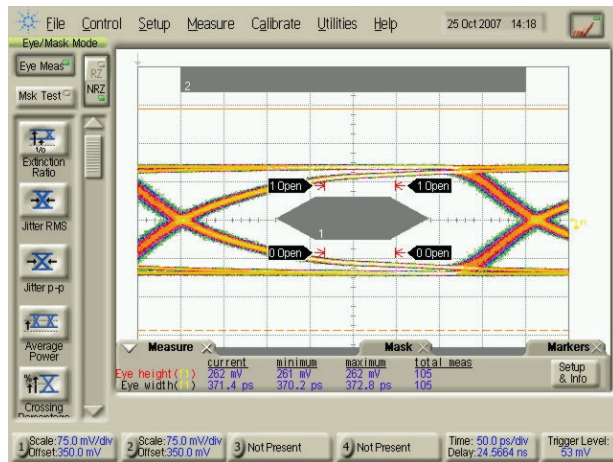
In this section, the SFI-5 interface is tested on several devices to verify these parameters of operation:

- Interface meets receiver eye mask requirement as specified in the OIF specification.
- Deskew capability meets or exceeds the maximum skew compensation requirement.
- Two SFI-5 interfaces on two different FPGAs pass traffic at  $< 10^{-12}$  bit error rate for an extended period of time.

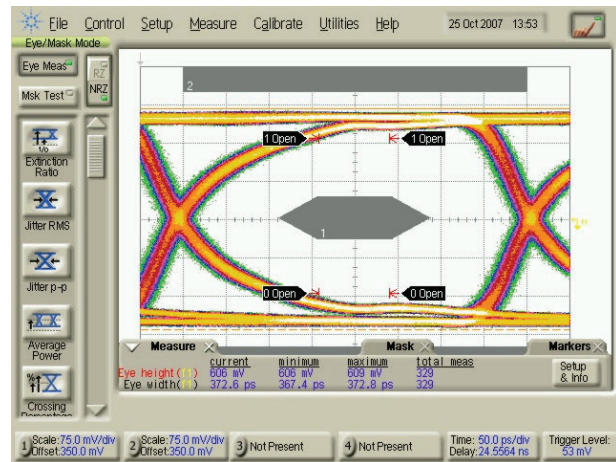
Figure 21 shows that the SFI-5 transmitter meets the receiver eye mask requirements with several transmission media at 2.5 Gb/s. The longest medium is 40 inches of FR4 and 60 inches of coaxial cable terminated with SMA connectors. In the 40-inch case, maximum signal pre-emphasis and the highest output swing setting are required to satisfy the receiver eye mask.



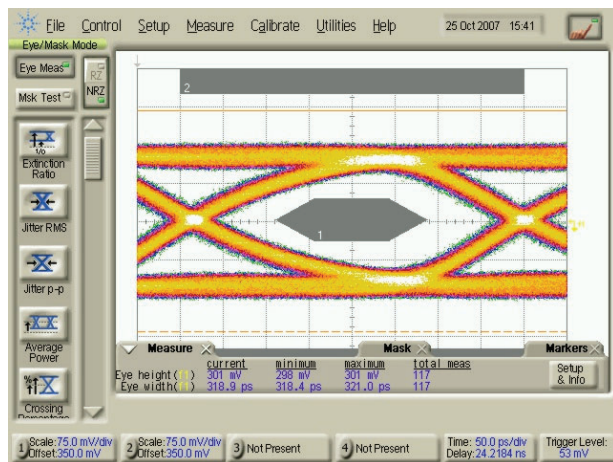
**S/N 12027**, 2.5 Gb/s, Output Swing = 400 mV,  
30-inch Coaxial Cable, Pre-emphasis = 000



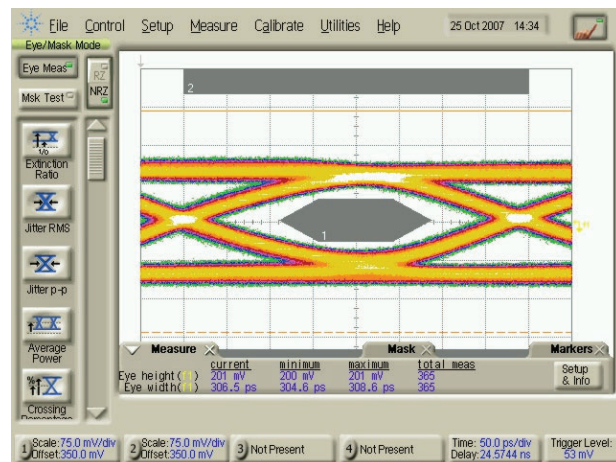
**S/N 12027**, 2.5 Gb/s, Output Swing = 1000 mV,  
30-inch Coaxial Cable, Pre-emphasis = 000



**S/N 12027**, 2.5 Gb/s, Output Swing = 1000 mV,  
60-inch Coaxial Cable + 20-inch FR4,  
Pre-emphasis = 101



**S/N 12027**, 2.5 Gb/s, Output Swing = 1100 mV,  
60-inch Coaxial Cable + 40-inch FR4,  
Pre-emphasis = 111



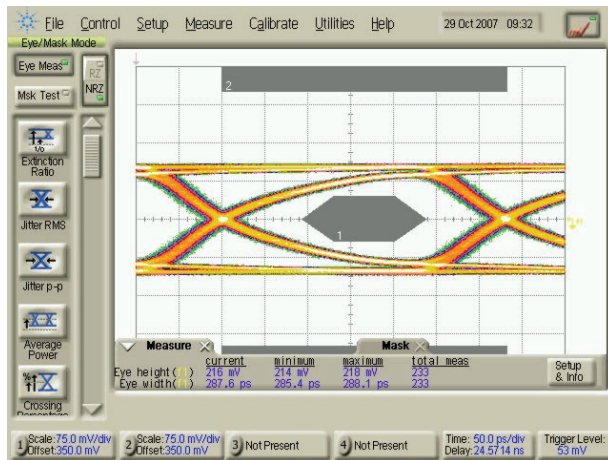
X871\_21\_121407

**Figure 21: Eye Masks of Device Number 12027 for Various Media at 2.5 Gb/s**

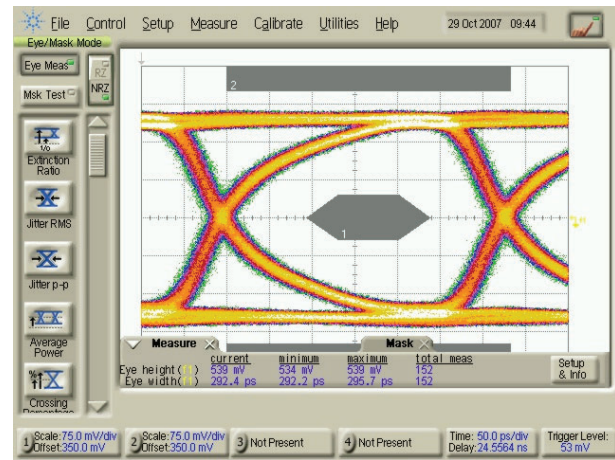
Figure 22 shows the same cases as Figure 21, except that the data rate is 3.0 Gb/s, corresponding to a total interface speed of 48 Gb/s. At 3.0 Gb/s, three of the four cases still meet the receiver eye mask requirement, but the fourth case fails the requirement. This indicates that the transmission medium load in the fourth case is too large for the transmitter to meet the input specification at 3.0 Gb/s.



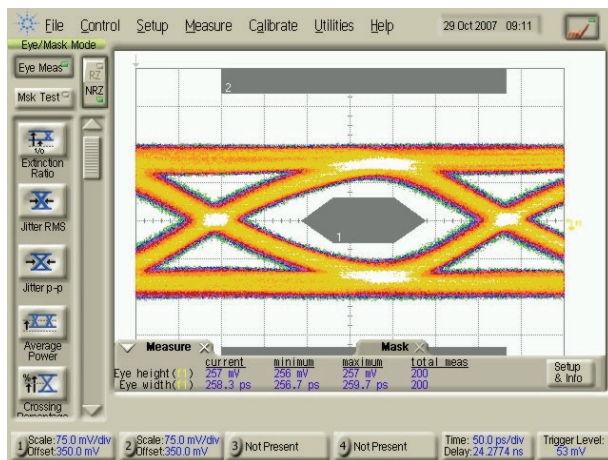
**S/N 12027**, 3.0 Gb/s, Output Swing = 400 mV,  
30-inch Coaxial Cable, Pre-emphasis = 000



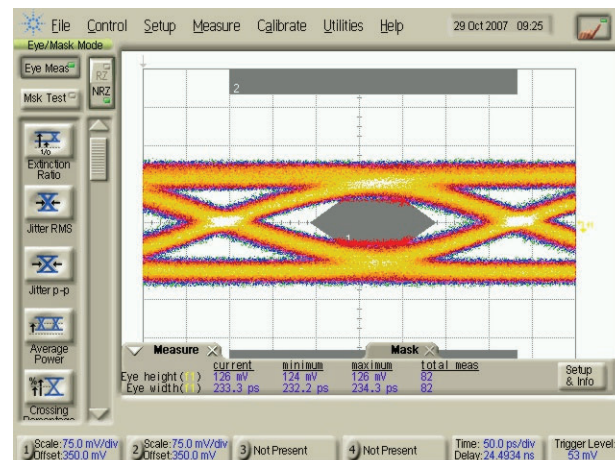
**S/N 12027**, 3.0 Gb/s, Output Swing = 1000 mV,  
30-inch Coaxial Cable, Pre-emphasis = 000



**S/N 12027**, 3.0 Gb/s, Output Swing = 1000 mV,  
60-inch Coaxial Cable + 20-inch FR4,  
Pre-emphasis = 101



**S/N 12027**, 3.0 Gb/s, Output Swing = 1100 mV,  
60-inch Coaxial Cable + 40-inch FR4,  
Pre-emphasis = 111



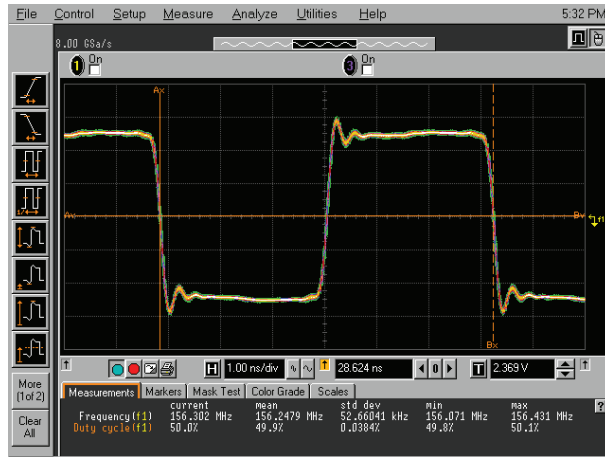
X871\_22\_121407

**Figure 22: Eye Masks of Device Number 12027 for Various Media at 3.0 Gb/s**

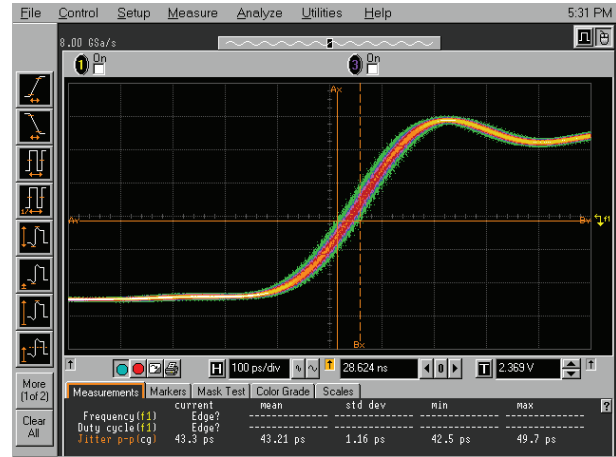
The reference clock of the SFI-5 interface can come from an oscillator or from another device in the SFI-5 link. Every SFI-5 transmitter is responsible for providing a reference clock, and the receiver can optionally use that clock. This clock is provided on the TXDCK port. For a 40 Gb/s interface, the reference clock is 156.25 MHz. For a 48 Gb/s interface, the reference clock is 187.5 MHz. This is shown in [Figure 23](#).



**S/N 12027**, Period and duty cycle distortion of reference clock (TXDCK) at 156.25 MHz



**S/N 12027**, Total jitter of reference clock (TXDCK) at 156.25 MHz (43.3 ps peak-to-peak)



X871\_23\_121407

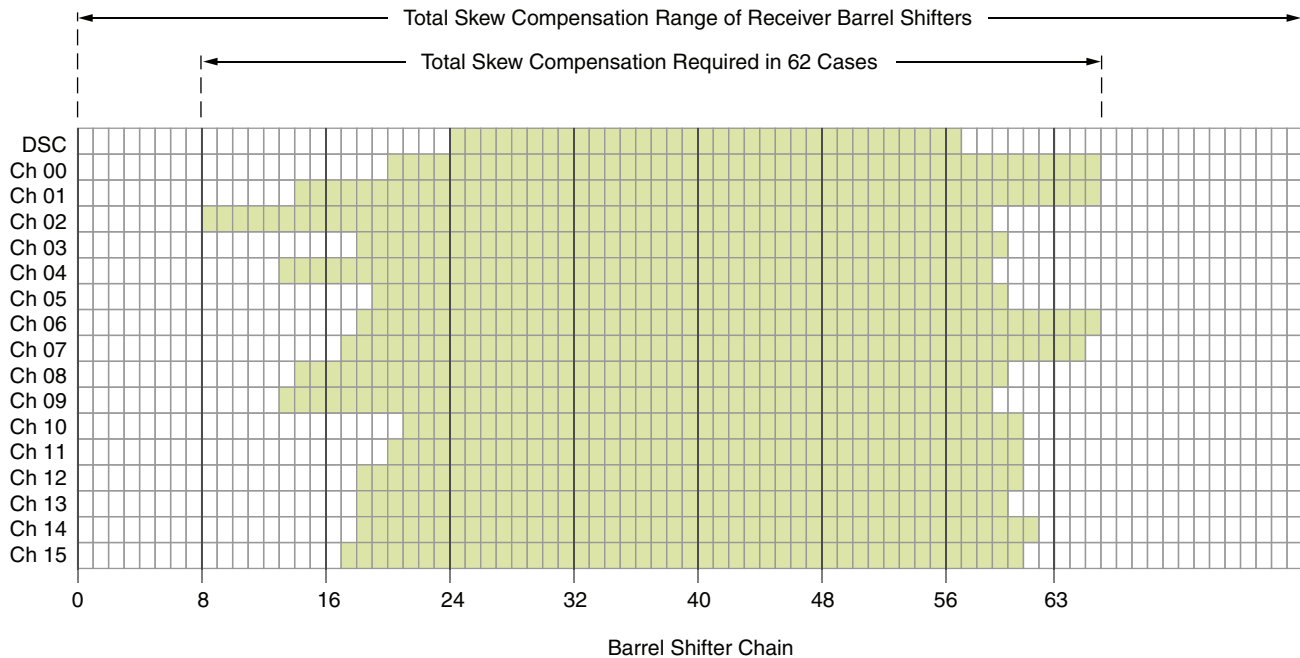
**Figure 23: Reference Clock (TXDCK) Provided by Transmitter for Optional Use by Receiver**

The deskewing capability of the receiver is tested in every combination of these conditions:

- Four XC5VLX330T devices (serial numbers 12027, 12010, 027, and 029)
- 24 inches of extra skew only on deskew channel (~8 UI of skew at receiver)
- 24 inches of extra skew only on data channel 2 (~8 UI of skew at receiver)
- 2.5 Gb/s and 3.0 Gb/s operation
- Multiple resets in some cases

In all of these conditions, the device is looped back externally to itself. There are several additional cases in the deskew testing in which two independent XC5VLX330T devices exchange data via an SFI-5 interface. In these cases, there is no loopback. The total sample size of deskew results is 62, each of which corresponds to a different case. [Figure 24](#) shows all 62 results overlapping one another. This overlapping shows how much of the barrel shifter chain in each datapath is needed to satisfy all 62 cases. This demonstrates that in all cases, the link compensates the maximum skew of 8 UI specified at the receiver inputs, and that there is margin to compensate more skew. The largest skew compensated in any of the 62 cases is 21 bit times. The specific test case is detailed in [Figure 27, page 33](#).

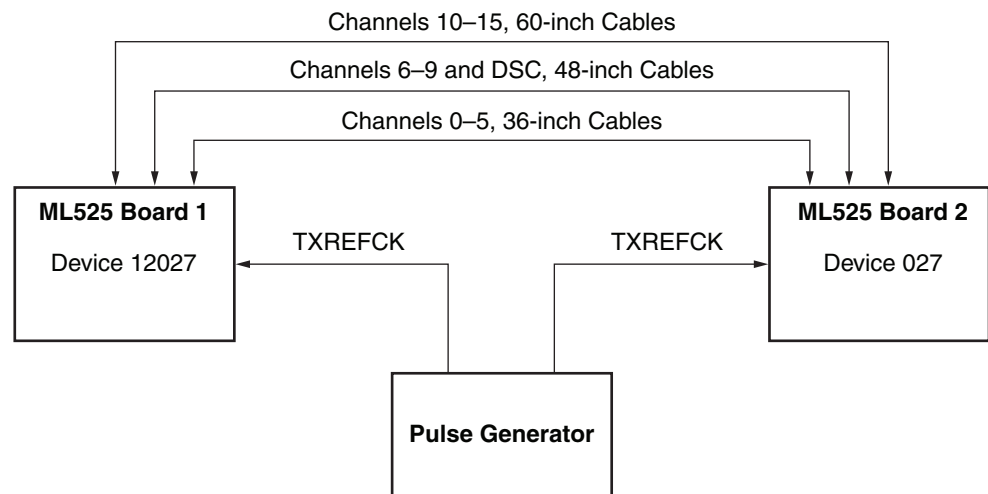




X871\_24\_022808

**Figure 24: Cumulative Barrel Shifter Chain Usage Showing Full Range of Settings over 62 Cases**

The test setup used for the SFI-5 system test is shown in [Figure 25](#). The reference clocks provided to both boards were synchronous. Three different cable lengths were used, creating a maximum skew of 24 inches, which corresponds roughly to 8 UI. The optional interface settings were set to the default values given in [Table 2](#), [page 4](#).

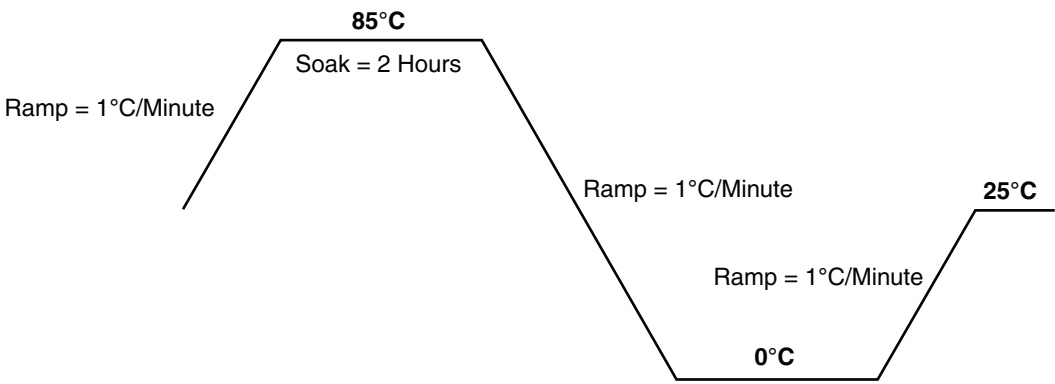


X871\_25\_121407

**Figure 25: System Test Setup**

A temperature forcing unit was applied to the FPGA on ML525 board 1. The unit was programmed to perform the temperature ramp shown in [Figure 26](#), which takes approximately eight hours to complete. The system test was performed twice: once with device number 12027 as the DUT, and once with device number 027 as the DUT. In both cases, there were zero mismatches, zero frame errors, and no alarms after the temperature ramp completed.

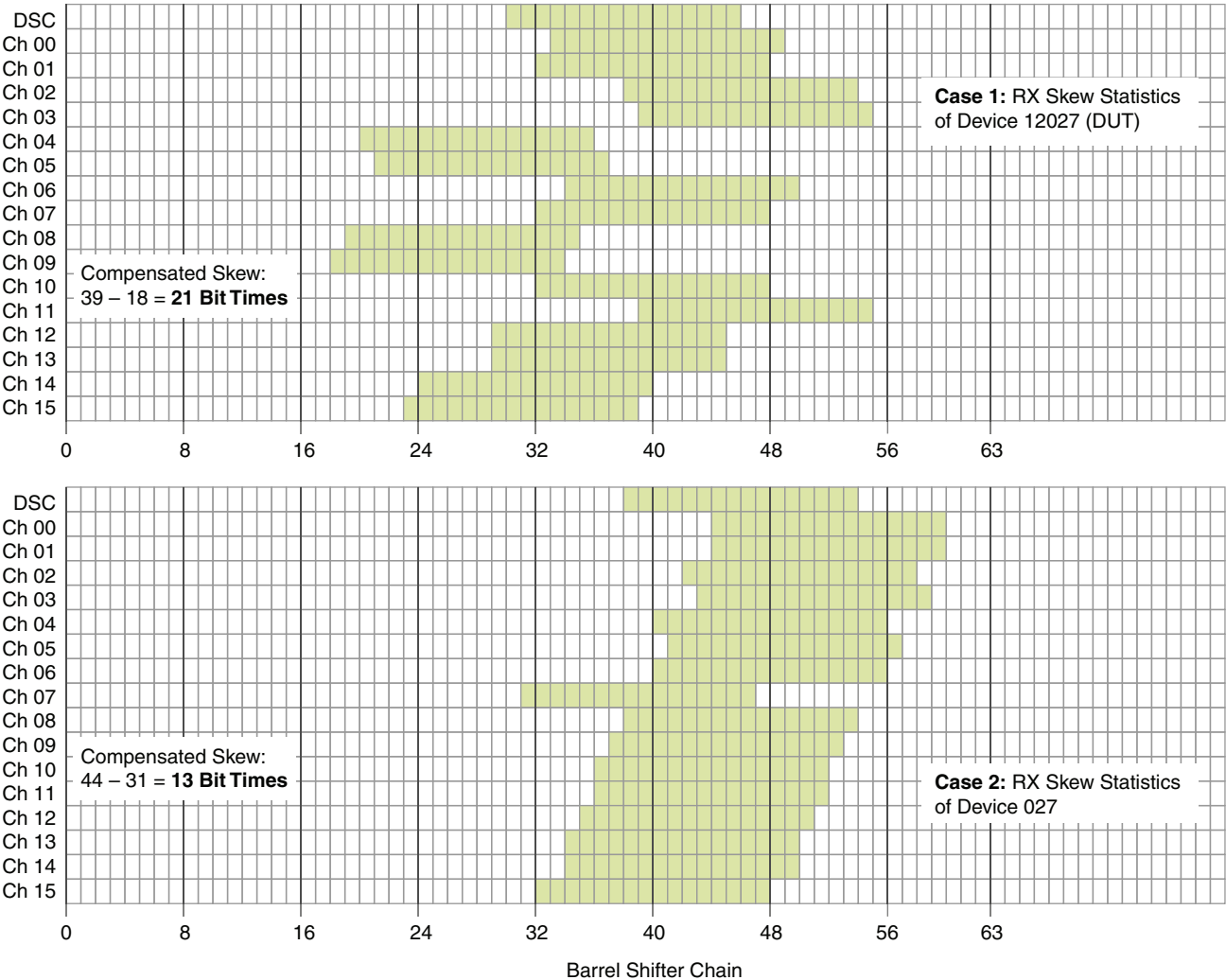




X871\_26\_121807

Figure 26: Temperature Ramp for System Test

The deskew results for device numbers 12027 and 027 are shown in Figure 27 and Figure 28, respectively, at a data rate of 40 Gb/s (2.5 Gb/s x 16 channels).



X871\_27\_022808

Figure 27: Barrel Shifter Selections For All Channels Showing Skew Results For System Test



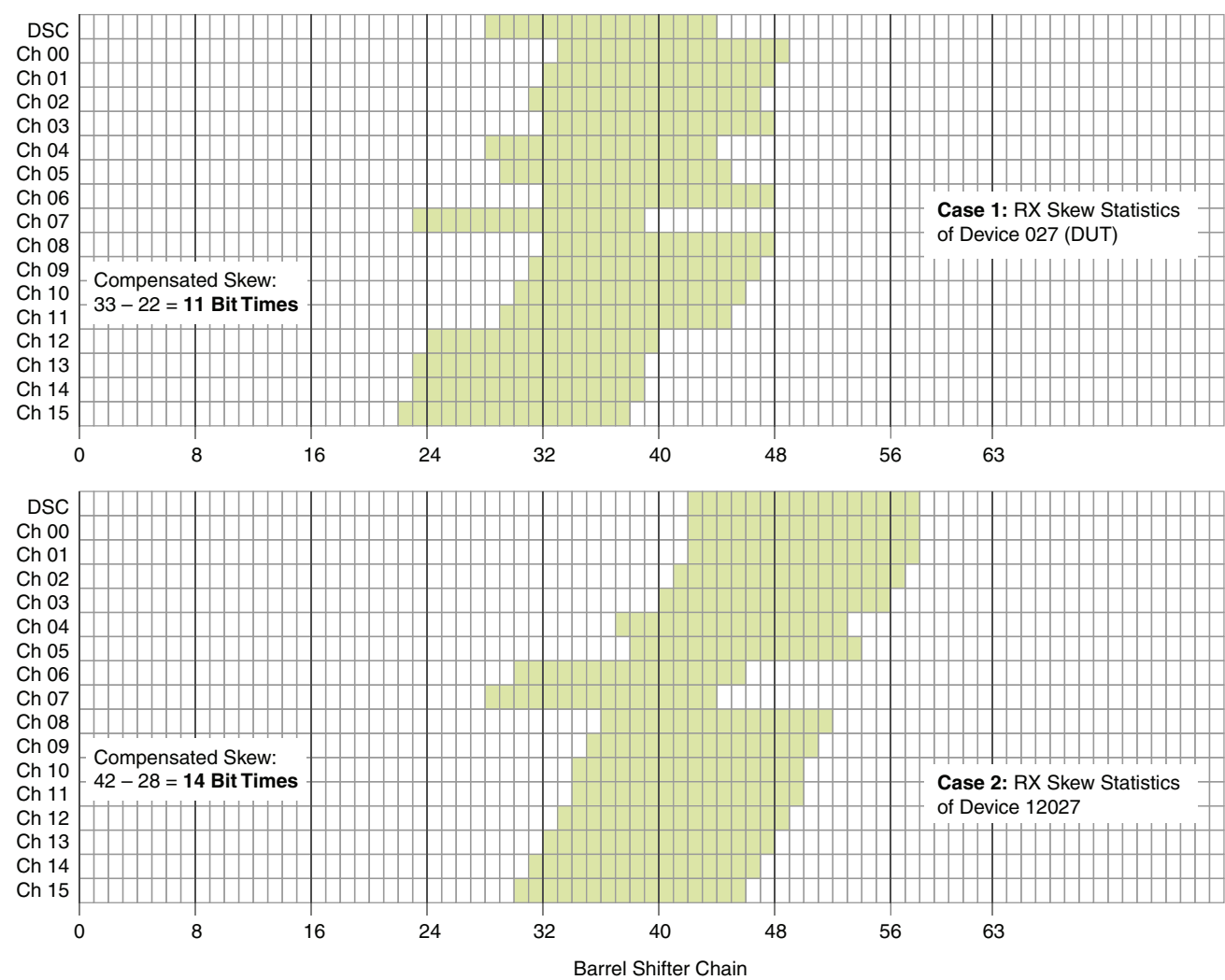


Figure 28: Barrel Shifter Selections For All Channels Showing Skew Results For System Test

Reference Design Matrix

The reference design matrix for this application note is shown in Table 5.

Table 5: Reference Design Matrix

Parameter	Description
General	
Developer Name	Xilinx
Target Devices (Stepping Level, ES, Production, Speed Grades)	XC5VLX330T
Source Code Provided?	Y
Source Code Format	Verilog
Design Uses Code or IP from Existing Reference Design, Application Note, 3rd party, or CORE Generator™ Software?	N
Simulation	
Functional Simulation Performed?	Y
Timing Simulation Performed?	N



Table 5: Reference Design Matrix (Cont'd)

Parameter	Description
Testbench Provided for Functional and Timing Simulations?	Y
Testbench Format	Verilog
Simulator Software and Version	ModelSim SE 6.2g
SPICE/IBIS Simulations?	N
<b>Implementation</b>	
Synthesis Software Tools and Version	ISE 9.2 SP3
Implementation Software Tools and Version	ISE 9.2 SP3
Static Timing Analysis Performed?	Y
<b>Hardware Verification</b>	
Hardware Verified?	Y
Hardware Platform Used for Verification	ML525

The reference design contains a simulation bug. The DLL\_FREQUENCY\_MODE attribute of the DCM called tx\_mgt\_clock\_multiplier should be set to HIGH to meet the hardware specification. However, in simulation it needs to be set to LOW, or else the CLK2X output does not toggle. Simulation behaves normally as long as the attribute is set to LOW. When the design is moved to hardware, the attribute should be set to HIGH.

## Reference Design

The reference design for this application note can be found at:

<https://secure.xilinx.com/webreg/clickthrough.do?cid=104178>

## Revision History

The following table shows the revision history for this document:

Date	Version	Description of Revisions
02/28/08	1.0	Initial Xilinx release.

## Notice of Disclaimer

Xilinx is disclosing this Application Note to you "AS-IS" with no warranty of any kind. This Application Note is one possible implementation of this feature, application, or standard, and is subject to change without further notice from Xilinx. You are responsible for obtaining any rights you may require in connection with your use or implementation of this Application Note. XILINX MAKES NO REPRESENTATIONS OR WARRANTIES, WHETHER EXPRESS OR IMPLIED, STATUTORY OR OTHERWISE, INCLUDING, WITHOUT LIMITATION, IMPLIED WARRANTIES OF MERCHANTABILITY, NONINFRINGEMENT, OR FITNESS FOR A PARTICULAR PURPOSE. IN NO EVENT WILL XILINX BE LIABLE FOR ANY LOSS OF DATA, LOST PROFITS, OR FOR ANY SPECIAL, INCIDENTAL, CONSEQUENTIAL, OR INDIRECT DAMAGES ARISING FROM YOUR USE OF THIS APPLICATION NOTE.