NXP Semiconductors
Application Note

Document Number: AN5119
Rev. 2, 01/2016

SerDes Configuration and Validation Tool Companion

Contents

1 Introduction

This application note is a companion document to the SerDes Configuration & Validation Tool User Guide. It is provided to assist those engineers wishing to use the Tx Equalization, Built-In Self Test (BIST), and Jitter Scope test features of the QCVS SerDes validation tool. The 10 G SerDes block is the basis for describing the technical topics. The 10 G SerDes is in the T4240, B4860, T2080, P5040, and T1xx QorIQ multicore processors families.

The fundamental blocks of a SerDes are a transmitter and a receiver. The transmitter serializes the parallel data, performs transmit equalization, and sends serial differential pair signals to the output pad and then across the channel (the path between the transmitter and receiver, that is, electrical components, printed circuit board, traces, cables, and so on) to the receiver. The receiver collects the serial data, extracts a clock from the data, de-serializes the data, and performs equalization if necessary.

The NXP 10 G SerDes implements both transmit equalization and receive equalization. Transmit and receive equalization is provided to compensate for lossy channels. An example is to take a single pulse (1 bit) and send it through a lossy channel. It starts out as a definite single pulse, but, by traversing the channel, it develops “tails” on either side of the received pulse, as shown in the following figure.

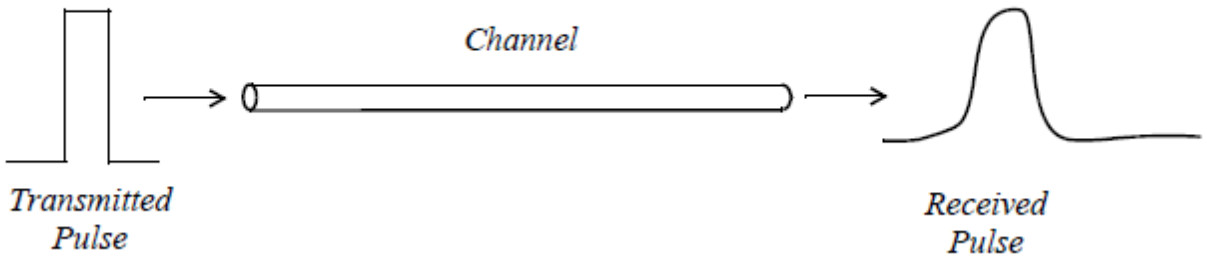

Figure 1. Single-pulse response with no equalization

In a data stream built of many pulses, these tails start to affect the bits on either side, making it difficult, if not impossible, for the receiver to extract the data stream. These effects are known as inter-symbol interference (ISI). At this point, an overlay graph of all data samples collected, called the data eye, would be closed. The Data Eye is discussed later in this document.

2 Transmit equalization

The 10 G SerDes is equipped to provide transmit equalization, as required by the protocol specification, for user configuration and debug. Transmit equalization pre-conditions the signal to compensate for any channel losses. The 10 G SerDes transmitter provides the Feedforward Equalization (FFE) scheme in the form of 2-tap equalization and 3-tap equalization. The transmit FFE implementation in the 10 G SerDes transmit path requires knowledge of three variables:

- The bit stream being serialized
- The amount of equalization required to invert channel losses
- Knowledge of the channel

Knowledge of the channel is beyond the scope of this document.

2.1 Transmit feedforward equalization

As mentioned previously, the FFE scheme is in the form of 2-tap equalization and 3-tap equalization. A tap is the point at which the weighting is applied to each cursor. 2-tap equalization applies/detracts energy at the first bit after the main bit. 3-tap equalization applies energy at the bit before and the bit after the main bit.

On the Lane(n) Configuration tab of the SerDes configuration and validation tool, tap equalization is selected at the option *Type*, where the choices are:

- *No Transmit Equalization*
- *2 Levels* (2-tap equalization)
- *3 Levels* (3-tap equalization)

Programmable differential peak-to-peak transmission swings and de-emphasis are the techniques used to adjust the overall output amplitude of the signal. Serialization is done inside the 10 G SerDes such that the data prior (pre) to the cursor and the data after (post) the cursor can be extracted and presented to the output amplifier cell. The SerDes configuration and validation tool allows pre-cursor and post-cursor adjustments on the Lane(n) Configuration tab, as shown in the following figure.

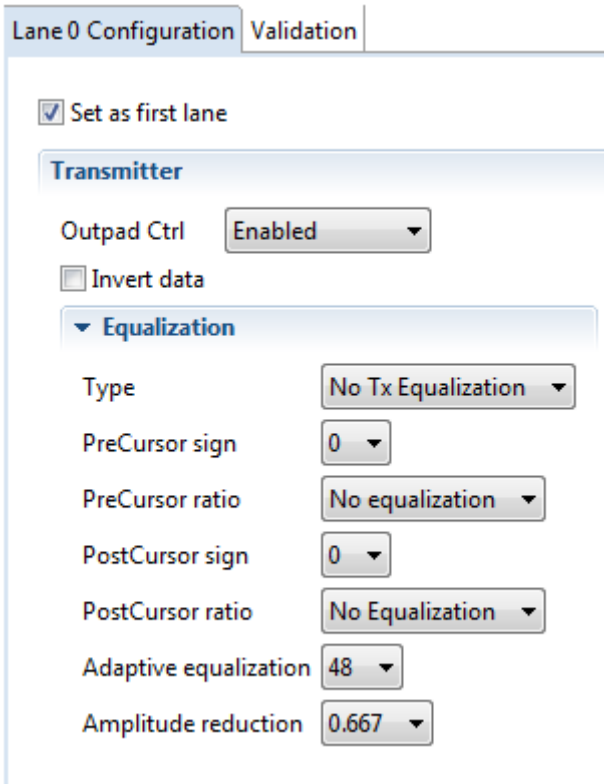

Figure 2. Lane 0 configuration showing PreCursor and PostCursor equalization options

The 10 G SerDes is constructed with 48 equalization units that can be applied across the transmit equalization taps. By adjusting the number of units, the output voltage swing is increased or reduced. For equalization control, the contributions of these 48 identical equalization units are steered to either the:

- Main cursor (20 dedicated units)
- Pre-cursor (12 dedicated units)
- Post-cursor (16 dedicated units)

If a pre-cursor or post-cursor is not used as a pre-cursor or post-cursor, then the unused units are used by the main cursor.

The number of equalization units, or weighting, applied to each cursor is controlled by the PreCursor ratio and PostCursor ratio options on the Lane(n) Configuration tab. 1-tap equalization consists only of the main cursor; 2-tap equalization and 3-tap equalization are addressed individually in the next sections.

This table shows examples of default transmit equalization settings for the T4240 PCIe, XFI, and 10GBaseKR. Default transmit equalization settings are device-specific. See the device reference manual for the default transmit equalization settings for each protocol and bit rate.

NOTE

None of the transmitter controls can be modified at the bit rate.

Table 1. Examples of default transmit equalization settings for T4240 PCIe, XFI, and 10GBaseKR

Protocol bit rate	Type	PreCursor sign	PreCursor ratio	PostCursor sign	PostCursor ratio	Adaptive equalization	Amplitude reduction
PCIe, 8	3-tap	1	No equalization	1	2	48	1.0
PCIe, 5, 6dB	2-tap	0	No equalization	1	2	48	1.0
PCIe 2.5, 3dB	2-tap	0	No equalization	1	1.5	48	1.0

Table continues on the next page...

Table 1. Examples of default transmit equalization settings for T4240 PCIe, XFI, and 10GBaseKR (continued)

XFI, 10.3125	2-tap	0	No equalization	1	1.14	48	0.585
10 GBaseKR, 10.3125	3-tap	1	1.09	1	2.18	32	1.000

At project creation, the SerDes Validation tool initially uploads the current equalization settings. However, if you re-configure protocol options using the SerDes validation tool, you must manually set the transmit equalization settings for any new protocol or data rate on the Lane(n) Configuration tab.

2.1.1 2-tap equalization

The SerDes block can be configured for 2-tap equalization by selecting "2 Levels" in the Type field on the Lane(n) Configuration tab.

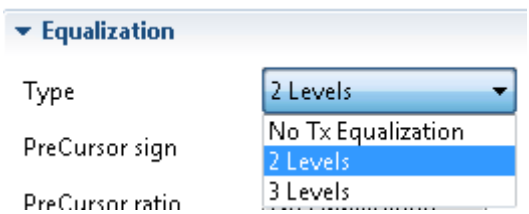

Figure 3. Transmit equalization type levels for 2-tap equalization

2-tap equalization consists of a main cursor and first post-cursor. In 2-tap equalization, transition bits swing the full programmed amplitude, while non-transition bits are de-emphasized and only swing a portion of the full programmed amplitude.

This figure shows the 2-tap equalization with amplitude reduction.

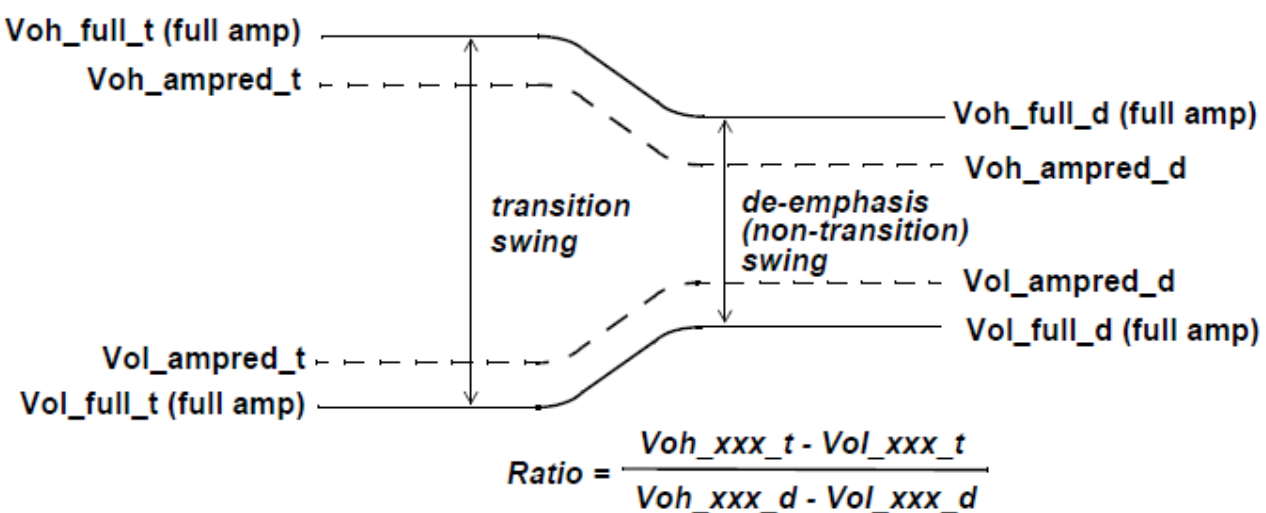

Figure 4. 2-tap equalization with amplitude reduction

The overall amplitude is set by using the Amplitude Reduction option on the Lane(n) Configuration tab. The drop down menu shows the relative amplitude reduction. This amplitude reduction applies to all bits in the bit stream.

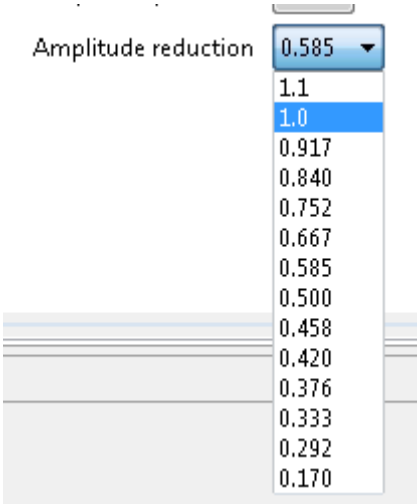

Figure 5. Amplitude reduction values

The relative de-emphasis levels can be adjusted via PostCursor ratio, which, in turn, controls the number of the 16 equalization units that are applied to the post-cursor. The remaining post-cursor equalization units are applied to the main cursor.

This figure shows the post-cursor ratios of full-swing transition bit to non-transition bit for each value of the firt post-cursor. These ratios only apply if the first PostCursor sign is set to "1", PreCursor is set to "No equalization", and Adaptive Equalization is set to "48".

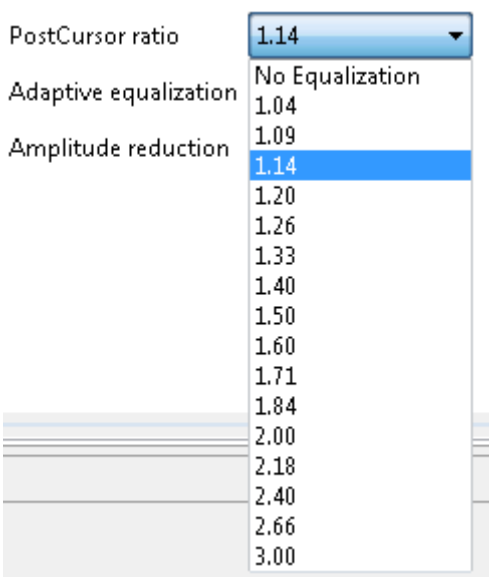

Figure 6. PostCursor ratio options

Adaptive Equalization is not used for 2-tap equalization. For the tool, Adaptive Equalization must be set to 48 to enable all equalization units.

2.1.2 3-tap equalization

In 3-tap equalization, both a pre-cursor and a post-cursor are available to help shape the data stream. The SerDes block can be configured for 3-tap equalization by selecting "3 Levels" in the Type field on the Lane(n) Configuration tab.

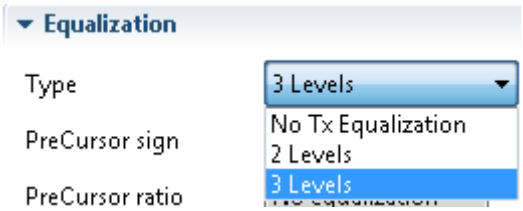

Figure 7. Transmit equalization type levels for 3-tap equalization

There are 12 equalization units available for the pre-cursor. Any unused pre-cursor equalization units are shifted to the main cursor. As with 2-tap equalization, there are 16 post-cursor equalization units and 20 main cursor equalization units for a total of 48 equalization units. Any unused post-cursor equalization units are also shifted to the main cursor. The total number of equalization units is always 48. Post-cursor sign and pre-cursor sign direction depends on whether the bit is a logic '1' or logic '0'. Selecting '1' for the sign bit increases the height of the transition with respect to the non-transition bits by the ratio selected. A '0' decreases the height of the transition bit with respect to the non-transition bit by the ratio value.

The following is an example of 3-tap equalization:

- Type = 3 Levels (3-tap equalization)
- Amplitude reduction = 1.0 (full amplitude swing)
- Adaptive equalization = 48 (all 48 equalization units in operation)
- PreCursor sign = 1 (open the eye with the pre-cursor)
- PostCursor sign = 1(open the eye with the post-cursor)
- PreCursor ratio = 1.09 (use 3 of the 12 equalization units for the pre-cursor)
- PostCursor ratio = 1.71 (use 10 of the 16 equalization units for the pre-cursor)

In other words, the transmitter has been configured to use 3-tap equalization with all 48 equalization units operating at their full swing. Both the pre-cursor and post-cursor have been set to open the eye. The pre-cursor uses 3 of its available 12 equalization units and the post-cursor uses 10 of its available 16 equalization units; thus, [(12-3) + (16-10)] = 15 additional equalization units are added to the main cursor for a total of (20+15) = 35.

This table shows the summing of the equalization units across the data stream.

Table 2. K28.5 and 3-tap equalization example

		bit time 1	bit time 2	bit time 3	bit time 4	bit time 5	bit time 6	bit time 7	bit time 8	bit time 9	bit time 10	bit time 11	bit time 12	bit time 13	bit time 14	bit time 15	bit time 16	bit time 17	bit time 18	bit time 19	bit time 20		
	0	*1*	*1*	*0*	*0*	*0*	*0*	*0*	*1*	*0*	*1*	*0*	*0*	*1*	*1*	*1*	*1*	*1*	*0*	*1*	*0*	1	
3	-35	10																					
	-3	35	-10																				
		-3	35	-10																			
			3	-35	10																		
				3	-35	10																	
					3	-35	10																
						3	-35	10															
							3	-35	10														
								-3	35	-10													
									3	-35	10												
										-3	35	-10											

Table continues on the next page...

Table 2. K28.5 and 3-tap equalization example (continued)

											3	-35	10										
												3	-35	10									
													-3	35	-10								
														-3	35	-10							
															-3	35	-10						
																-3	35	-10					
																	-3	35	-10				
																		3	-35	10			
																			-3	35	-10		
																				3	-35	10	
																					-3	35	-10
Total		*32*	*28*	*-42*	*-22*	*-22*	*-22*	*-28*	*48*	*-48*	*48*	*-42*	*-28*	*42*	*22*	*22*	*22*	*28*	*-48*	*48*	*-48*		

2.1.3 Adaptive equalization

A feature of the 10 G SerDes block is adaptive equalization. It is recommended to use the default settings for each protocol.

3 SerDes receive path

For certain high speed protocols, the Rx Equalization Boost bit is enabled by default to provide extra equalization for high loss channels. However, many customer channels are not high loss and enabling the Boost bit may adversely enhance the return loss due to some discontinuities presented in the channel. This may further cause more reflection. Unless the channel is high loss, the Rx Equalization Boost bit should be disabled, meaning, the Boost box is unchecked, for all lanes for better performance. Board level simulation should always be performed to determine whether the SerDes channels are in high loss condition so that the Rx Equalization Boost bit is configured properly.

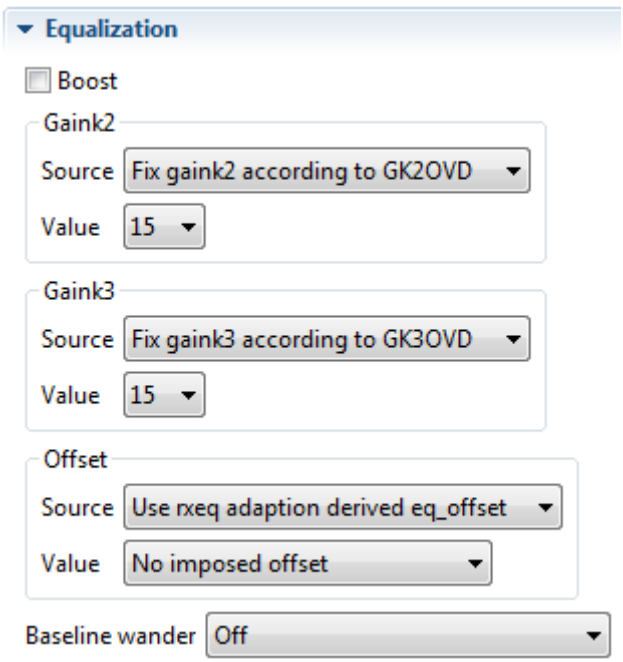

Figure 8. Receive equalization recommended default settings

4 Electrical idle

Electrical Idle settings are factory configured for BIST and Jitter Scope. It is recommended not to change any settings.

4.1 Built-in self test (BIST)

The 10 G SerDes block allows for Built-In Self Test (BIST) on a lane-by-lane basis. Serial received test data can be generated via either the built-in pattern generator on the transmit side of the SerDes or by an external pattern generator. The SerDes configuration validation tool allows the user to run the BIST, which can be selected on the Validation tab shown in the following figure.

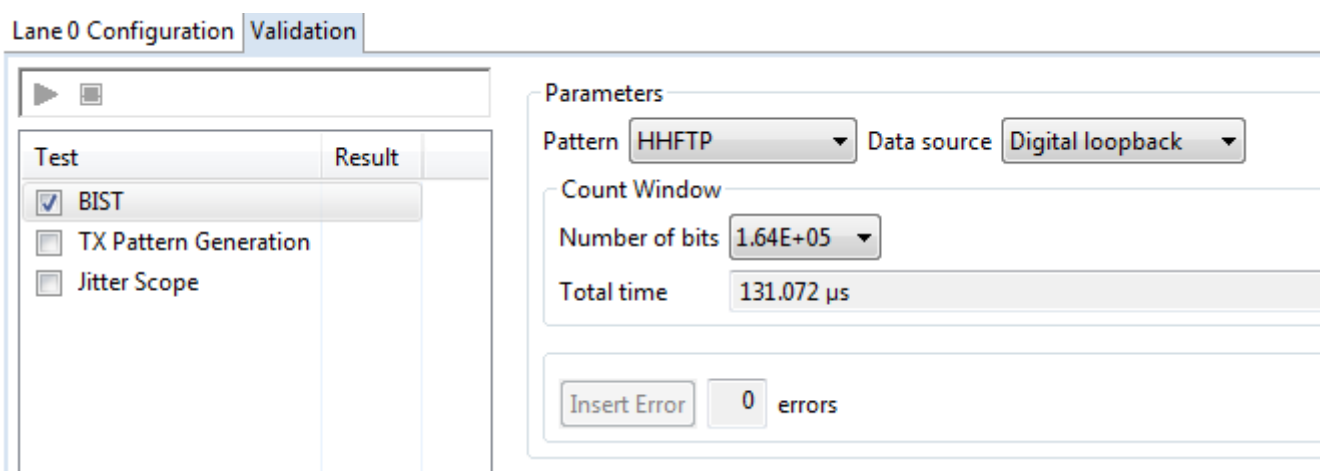

Figure 9. Validation tab

The following figure shows a number of Built-In Test patterns, which can be generated for testing a SerDes link. These patterns can be used either with the on-chip BIST circuit or in system testing.

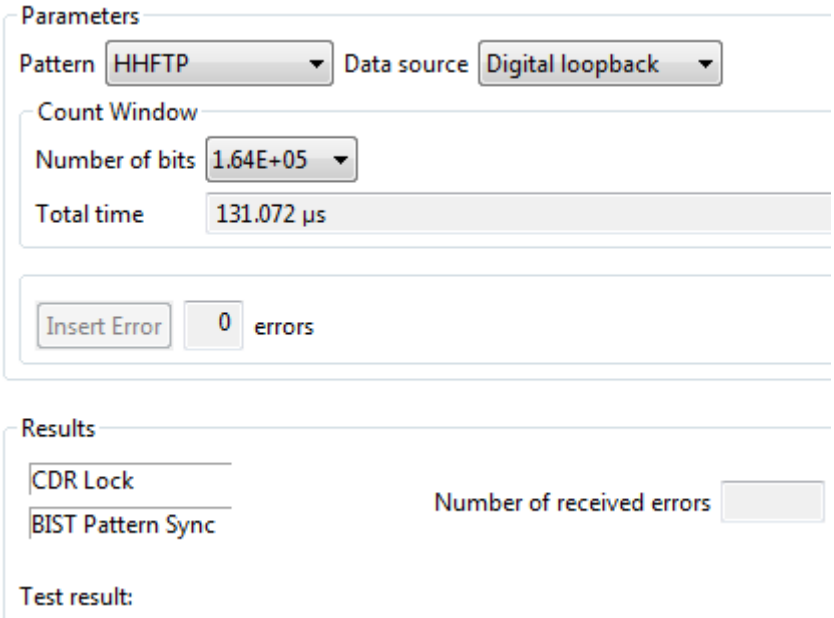

Figure 10. Parameters for BIST on SerDes configuration and validation tool

Full descriptions of test patterns are located in Appendix A. Patterns available for BIST are selected via the Pattern drop-down menu.

NOTE

DIV## patterns are for NXP lab use only.

The following figure shows the number of bits, which correspond to different amounts of time depending on data rate. One UI is a bit, even if it represents a different number of picoseconds at different data rates, and it is used in many calculations in serial communications.

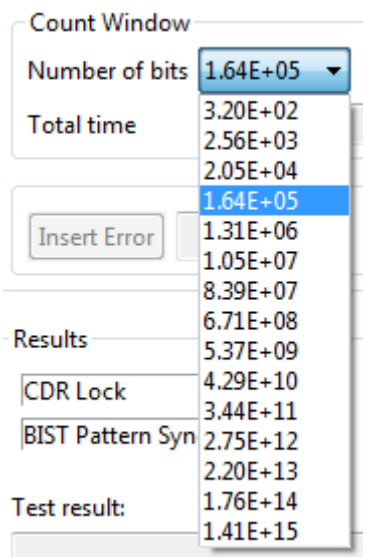

Figure 11. Count window number of bits range

4.1.1 Built-In Self Test modes

There are three data source options to run BIST in the SerDes validation tool:

- Digital loopback
- External loopback
- External mode

Digital loopback is an internal loopback. External loopback generates the selected pattern and transmits it to the Tx pin, which is connected to the Rx pin through an external user-supplied connection. In External mode, the generated pattern is transmitted to the TX pin of the device. To connect to an external device, connect the Tx pin to the necessary cable for communication with the partnering device.

4.1.2 Insert Error Button feature

The Insert Error Button feature allows you to insert an error into the transmitted test pattern during the BIST testing. You must select the largest Number of Bits in the Count Window before starting the BIST test. Errors can be inserted only after the BIST pattern sync has successfully passed. This is indicated by a green checkmark as shown in the following figure.

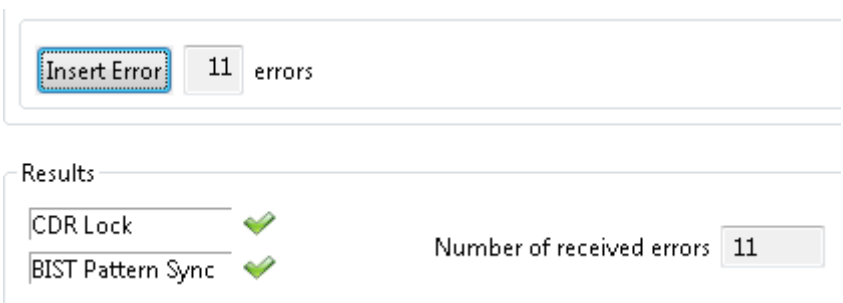

Figure 12. Insert error passing BIST test

The error is not linked to any specific point in any transmitted BIST pattern, but occurs whenever the Insert Error Button is clicked. The BIST test has passed using this feature only if the number of inserted errors is equal to the number of receive errors. If they are not equal, the test fails.

4.1.3 Results

After the BIST has completed, the test success or failure can be observed from the Results section of the Validation tab. Results are broken down into four items:

- Clock and Data Recovery (CDR) lock
- BIST pattern sync
- Number of received errors
- Test result

If the test is successful, the output matches the following figure.

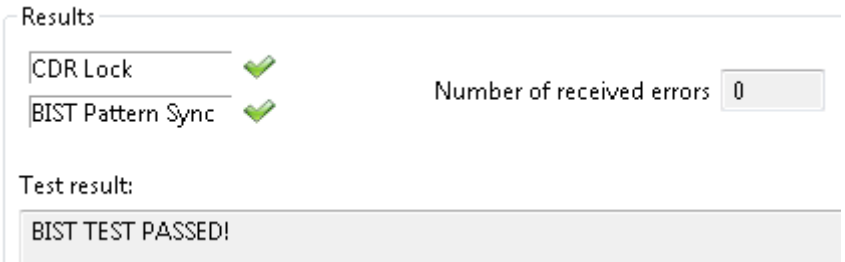

Figure 13. Example of BIST test passed

If the test is unsuccessful, the green checkmark is replaced by a red "x" and the error appears in the Test result field.

4.1.3.1 Clock and data recovery

The SerDes Receive Path receives serial data, extracts a clock from the data, and deserializes the data to either 20 bits, 16 bits, or 10-bits of parallel receive data. The 10 G SerDes receive path uses data and edge samples to extract the received clock and data. There is a CDR lock circuit that asserts CDR Lock when it has acquired a valid Rx clock.

4.1.3.2 BIST pattern sync

BIST pattern Sync is asserted when the serial received test data matches the selected BIST pattern generated by either the built-in test pattern generator on the transmit side of the SerDes or by an external pattern generator.

4.1.3.3 Number of received errors

After the BIST has initiated, the lane pattern error analyzer is started. Any reported mismatch errors are reported in the Results section.

4.1.3.4 Test results

As described in section 4.1.3 if the test is successful, the output is *"BIST TEST PASSED*"shown in the above Figure 14. If test is unsuccessful, *Test result* will state the error message.

5 Transmit pattern generation

Tx Pattern Generation allows the user to select a built-in pattern to transmit from the SerDes. The SerDes configuration and validation tool allows the user to run the Tx Pattern Generation, as shown in the following figure.

NOTE

DIV## patterns are only used for internal NXP testing.

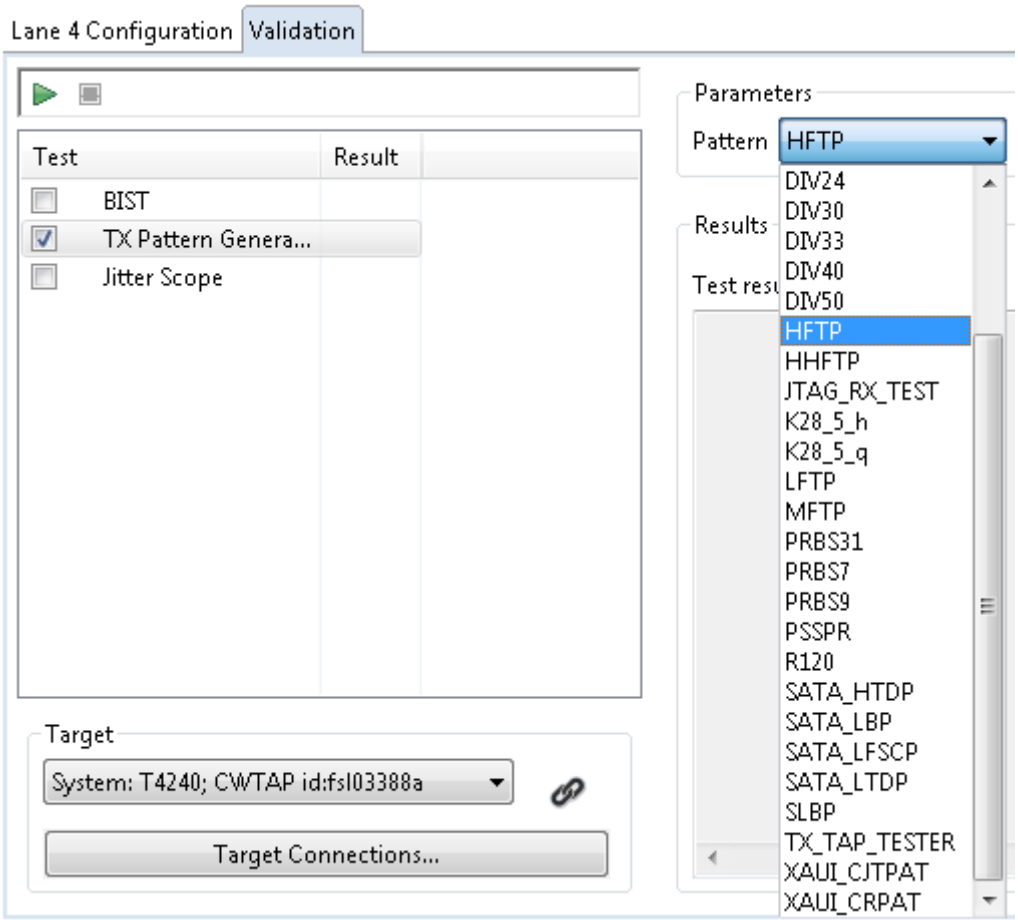

Figure 14. Pattern selection list for Tx pattern generation

After the Pattern is selected, begin transmitting the pattern by clicking the Start button. The pattern continues to transmit until the Stop button is clicked.

6 Jitter Scope mode

Jitter Scope mode is a characterization circuitry built within the 10 G SerDes receive path that makes it possible to construct a zero versus one probability curve over a repeating sequence of bits. This curve, from the point of view of the sampler after receiver equalization and data tracking, provides a measurement of eye opening in the receive path.

Controls for Jitter Scope mode testing are done on a lane-by-lane basis. This figure shows the options for Lane 0 configuration in Jitter Scope mode.

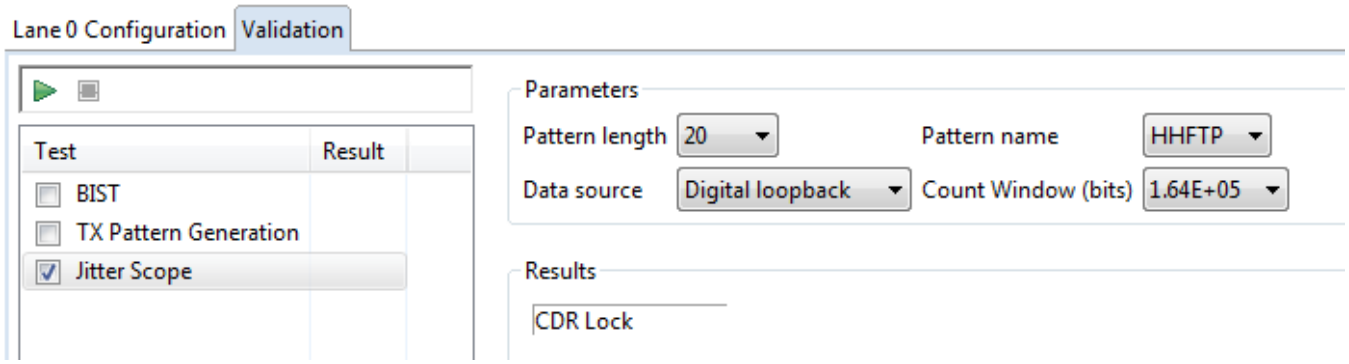

Figure 15. Jitter scope options on the validation tool

The pattern put into the receiver can be generated from either a resident 10 G SerDes BIST pattern via digital loopback or external loopback, or from an external source. Any externally sourced patterns must match in length to one of the resident 10 G SerDes BIST patterns.

This table provides the options for Pattern Length and Pattern Name in the Parameters section of the Jitter Scope options. Several of the pattern lengths listed are not exact matches or multiples of the lengths of the BIST patterns, but are modified to produce expected results as well as allow flexibility in programming external patterns.

Table 3. BIST pattern descriptions and pattern lengths

Pattern	Pattern Description	Pattern Length (Number of Bits)
D21.5 (HFTP - Nyquist))	High Frequency Pattern (Alternating '1' and '0')	10
D24.3 (HHFTP - Nyquist/2)	Half High Frequency Pattern (Alternating ‘11’ and ‘00’)	20
K28.7 (LFTP)	Low Frequency Pattern (Alternating ‘11111’ and ‘00000’)	10
K28.5 (MFTP)	Mixed Frequency Pattern	20
D27.5 (SLBP)	Simple Lone Bit Pattern	20
D8.7 (ASLBP)	Another Simple Lone Bit Pattern	40
XAUI CRPAT	XAUI Compliant Random Data Pattern	3780
XAUI CJTPAT	XAUI Compliant Jitter Data Pattern	3820
CRPAT	MJSQ Compliant Random Data Pattern	3360
CJTPAT	MJSQ Compliant Jitter Data Pattern	2640
R120	120 ‘1’s followed by 120 ‘0’s	240
PRBS9	PRBS9	511
PRBS7	PRBS7	127

The Jitter Scope count window sets the number of bits collected during sampling.

This figure shows the Jitter Scope count window values.

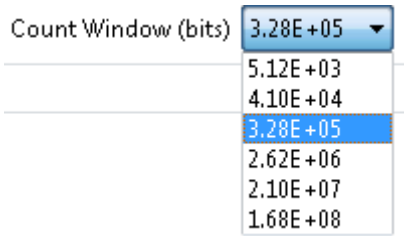

Figure 16. Jitter scope count window values

Similar to BIST, CDR Lock must be obtained before a Data Eye or a Recovered Data Stream can be acquired.

This figure shows the Recovered Data Stream.

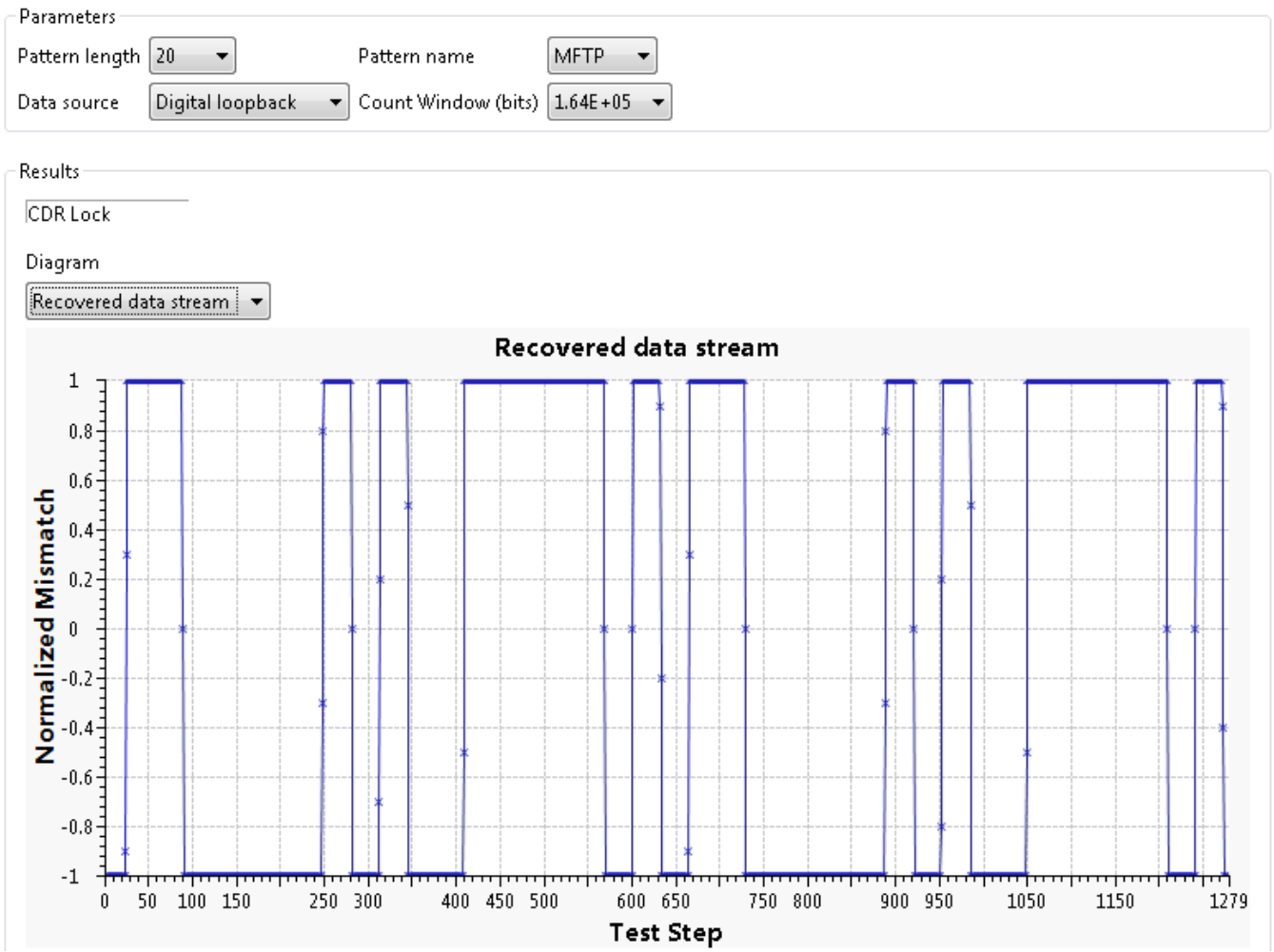

Figure 17. Recovered Data Stream of MFTP pattern, length 20 over digital loopback

This figure shows the Data Eye plot, which represents the received signal after all analog processing and sampling (that is, after the signal has been captured in the digital part of the SerDes).

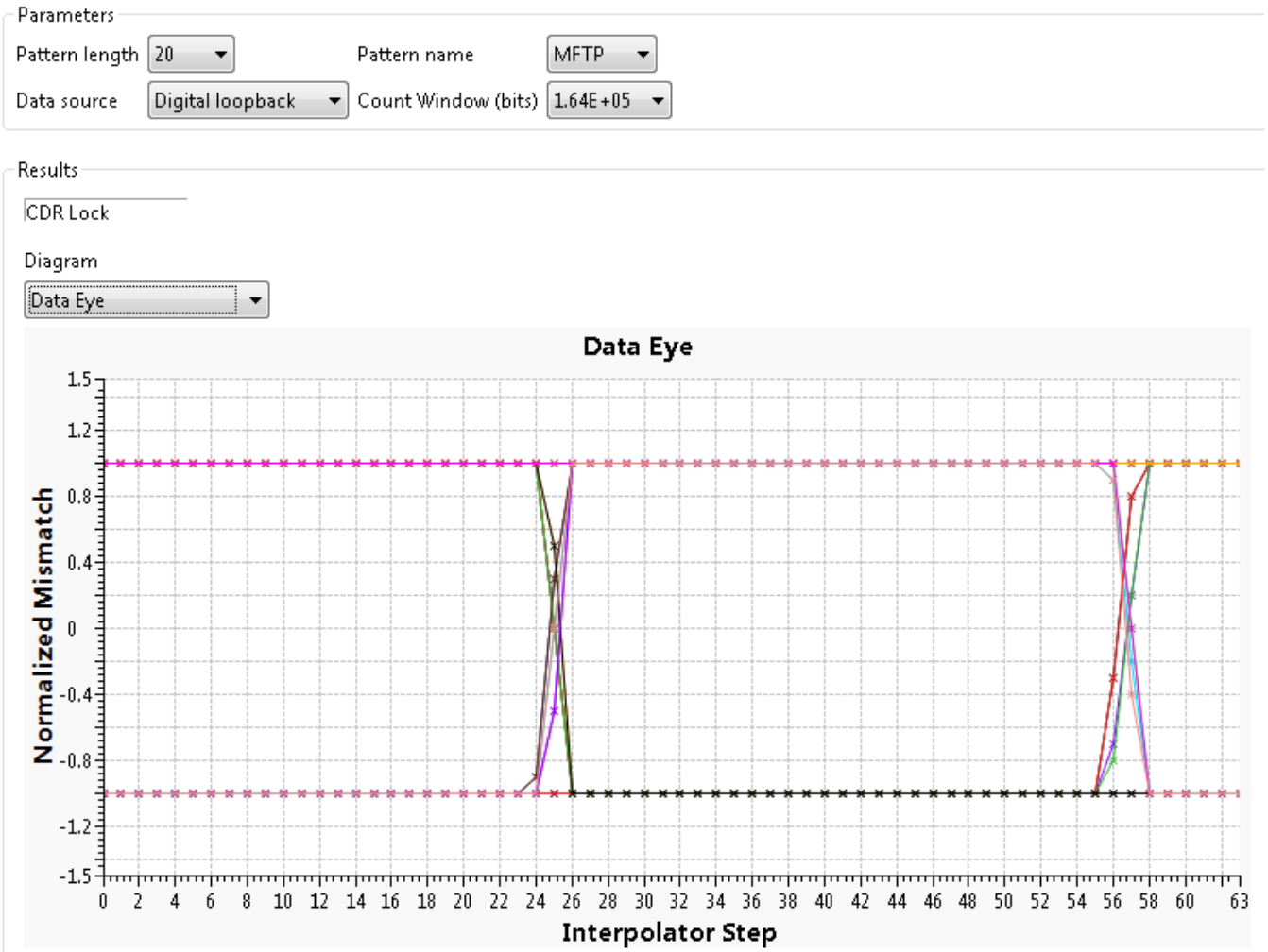

Figure 18. Data Eye of MFTP pattern, length 20 over digital loopback

The x-axis is time in units of steps, where each bit time (or unit interval (UI)) is broken into 16 or more steps based on rate selection. The steps are relative to where the SerDes currently observes the median data transition time. Each point in the plot represents samples taken from many bits at the same time position in a repeating bit pattern.

The y-axis is not the amplitude of the signal. Rather, it represents what percentage of the samples taken at that time step are ones versus zeros. The axis is normalized such that +1 is 100% “ones” at that time step, 0 is 50% ones and 50% zeros, and -1 is 100% “zeros” at the time step.

The data sample is taken from the step located in the middle of the data eye. In the absence of bit errors, the y-value would be +1 or -1 at that step. The jitter plot can take a long time to run, so typically the test is run with a small fraction of the number of bits needed to demonstrate the specified bit error rate. Still, it is a quick visual indication of initial signal integrity or configuration issues that interfere with the link working properly.

7 Debugging tips

This section describes debugging tips to help identify configuration issues that interfere with the link working properly or initial signal integrity issues.

7.1 Check power and clocks

If the SerDes link is not working properly or if the SerDes PLL is not locked (SRDSxPLLncR0[PLL_LCK] = 0), perform the following tests:

1. Measure all core/platform supply voltages and all PLL supply voltages, and ensure they meet requirements as specified in the device hardware specification.
2. Check all SerDes reference clock inputs and ensure they meet requirements as specified in the device data sheet.
3. Check the following SerDes register programming:
    - SRDSxPLLnCR0[POFF] = 0: Ensure the PLL is powered on for lanes referencing this PLL.
    - SRDSxPLLnCR0[RFCLK_SEL]: Check that the clock input SD_REF_CLK/SD_REF_CLK_B inputs match the programming.
    - SRDSxPLLnCR0[FRATE_SEL]: Check that the requested PLL VCO frequency matches the programming.
    - SRDSxPLLnCR1[PLLBW_SEL] Check that the PLL bandwidth configuration matches the programming.
    - SRDSxPLLnCR1[VCO_EN]: Check that the PLL VCO matches the selected bit rate.

7.2 Verify actual bit rate

Measure the actual bit rate and ensure it is as expected:

1. Set the SerDes Configuration and Validation Tool to Tx Pattern Generation mode to transmit an HFTP pattern.
2. Use an oscilloscope to measure the bit rate at the transmitter output (SD_TXn/SD_TXn_B).

7.3 Run BIST

Set the SerDes Configuration and Validation Tool to run BIST in digital loopback mode:

1. Start with a simple pattern, such as the HFTP, to check for basic functionality.
2. Progress to a more challenging pattern, such as the MFTP.
3. Re-check the bit rate and ensure both Tx and Rx are running at the expected rate.

If the digital loopback test passes, set the SerDes Configuration and Validation Tool to run BIST in external loopback mode, which has the added advantage of the effects of ISI being removed:

1. Start with the HFTP.
2. If the HFTP passes, run the MFTP. If errors are received and the bit rate is correct, there may be a signal integrity problem. In this case, proceed to the Jitter Scope test.
3. Run the CJTPAT pattern. This test fails on a bad channel or twisted connection, that is, SD_TX_P is connected to SD_RX_N. If you set SRDSxLNmGCR1[RDAT_INV] = 1 and the CJTPAT test passes, then the connection is twisted. If CJTPAT fails, set the RDAT_INV field back to 0 and proceed to the Jitter Scope test.

If you have the capability to generate patterns from an external data source to transmit to the device, use the SerDes Configuration and Validation Tool to run in external mode:

1. Ensure the selected pattern matches the pattern from the external data source.
2. Run BIST.

7.4 Run Jitter Scope

Run the Jitter Scope mode to construct a data eye:

1. Use the same flow as the BIST procedure, that is, start with HFTP then proceed to MFTP.
2. View the data eye and recovered data stream for a quick check of the eye opening in the receive path. These graphs provide a quick visual indication of initial signal integrity.

7.5 Identify signal integrity issues

If the Jitter Scope indicates signal integrity issues, use the following techniques to identify and improve on problem areas:

1. Configure the link partner to generate appropriate patterns and adjust the Tx equalization settings to see if a setting can get a board design to work. You may need to perform a manual sweep of the Tx equalization settings or perform automatic Tx adaptive equalization. Use the Jitter Scope mode in external mode to monitor improvements to the data eye.
2. If it is not possible to control the link partner device, remove the link partner device and replace it with a signal source bit error rate tester (BERT) to test the channel to our receiver. Use Time Domain Reflectometry (TDR) of the channel to confirm the quality of the probe or the hard connection to the board. The TDR can also look for discontinuities in the channel.
3. Record the TDR measurement of the channel to identify impedance discontinuities that may be indicators of potential signal integrity issues. Model the channel with improvements to the channel for those discontinuities shown by the TDR.

8 Appendix A: BIST pattern descriptions

8.1 High Frequency Test Pattern (HFTP)

This pattern is a continuous stream of single-bit alternating ones and zeros. In real-time it appears as a single frequency equal to the 1/(2*UI) or Nyquist pattern. It is used to test random jitter (RJ) at a BER of 10-12 (or lower) and to test the asymmetry of transition times. It can be coded by the repeated transmission of either of the 8B/10B D21.5 code-group. This pattern corresponds to the High Frequency Test Pattern called out in IEEE Std 802.3ae-2002, Annex 48A, Section 48A.1.

Table 4. D21.5 coding

Pattern name	Repeating data pattern	8B/10B 10-bit codes (D21.5)	
		Current RD-	Current RD+
HFTP	10	101010 1010	101010 1010

8.2 Half High Frequency Test Pattern (HHFTP)

This pattern is a continuous stream of double-bit alternating ones and zeros. In real-time it appears as a single frequency equal to the 1/(4*UI). It can be coded by the repeated alternate transmission of both of the 8B/10B D24.3 code-groups.

Table 5. D24.3 coding

Data name	Repeating data pattern	8B/10B 10-bit codes (D24.3)	
		Current RD-	Current RD+
HHFTP	1100	110011 0011	001100 1100

8.3 Low Frequency Test Pattern (LFTP)

This pattern is a continuous stream of five-bit alternating ones and zeros. It can be used to test low frequency random jitter (RJ) and to test PLL tracking error. In real-time it appears as a single frequency equal to the 1/(10*UI). It can be coded by the repeated transmission of either the RD+ or RD- of the 8B/10B K28.7 code-group. This pattern corresponds to the Low Frequency Test Pattern called out in IEEE Std 802.3ae-2002, Annex 48A, Section 48A.2.

Table 6. K28.7 coding

Data name	Repeating data pattern	8B/10B 10-bit codes (K28.7)	
		Current RD-	Current RD+
LFTP	1111100000	001111 1000	110000 0111

8.4 Mixed Frequency Test Pattern (MFTP)

This pattern can be used to test the combination of RJ and deterministic jitter (DJ) by transmitting five bits strings of either ones or zeros mixed with single bits. It can be coded by the repeated alternate transmission of both of the 8B/10B K28.5 code-groups. This pattern corresponds.

Table 7. K28.5 coding

Data name	Repeating data pattern	8B/10B 10-bit codes (K28.5)	
		Current RD-	Current RD+
MFTP	00111110101100000101	001111 1010	110000 0101

8.5 Simple Lone Bit Test Pattern (SLBP)

This pattern can be coded by the repeated alternate transmission of both of the 8B/10B D27.5 code-groups.

Table 8. K27.5 coding

Data name	Repeating data pattern	8B/10B 10-bit codes (K27.5)	
		Current RD-	Current RD+
SLBP	11011010100010011010	110110 1010	001001 1010

8.6 Another Simple Lone Bit Test Pattern (ASLBP)

This pattern can be coded by the repeated alternate transmission of both of the 8B/10B D8.7 code-groups.

Table 9. D8.7 coding

Data name	Repeating data pattern	8B/10B 10-bit codes (D8.7)	
		Current RD-	Current RD+
ASLBP	11100100010001101110	111001 0001	000110 1110

8.7 XAUI CRPAT

This pattern corresponds to the continuous random data pattern (CRPAT) described in the IEEE 802.3ae specification, Annex 48A.4. The IEEE 802.3ae specification is also referred to as the XAUI specification.

Table 10. XAUI CRPAT coding

Data name	Encoding
XAUI CRPAT	*See IEEE 802.3ae, Annex 48A.4*

8.8 XAUI CJPAT

This pattern corresponds to the continuous jitter data pattern (CJPAT) described in the IEEE 802.3ae specification, Annex 48A.5. The IEEE 802.3ae specification is also referred to as the XAUI specification.

Table 11. XAUI CJPAT coding

Data name	Encoding
XAUI CJPAT	*See IEEE 802.3ae, Annex 48A.5*

8.9 Compliant random data pattern (CRPAT)

This pattern corresponds to the CRPAT described in the MJSQ T11.2 Specification, Annex A (A.2.2.4). Although the MJSQ T11.2 is a FibreChannel specification and 10 G SerDes does not support FibreChannel, this is also a very common pattern used in jitter analysis systems as a standard CRPAT and since the 10 G SerDes block itself has no knowledge of encoding or protocol framing, this CRPAT is a valid pattern for jitter testing of the 10 G SerDes block.

Table 12. CRPAT coding

Data name	Encoding
CRPAT	*See the MJSQ T11.2 specification, Annex A (A.2.2.4)*

8.10 Compliant jitter tolerance data pattern (CJTPAT)

This pattern corresponds to the CJTPAT described in the MJSQ T11.2 specification, Annex A (A.2.3.3). Although the MJSQ T11.2 is a FibreChannel specification and the 10 G SerDes block does not support FibreChannel, this is also a very common pattern used in jitter analysis systems as a standard CJTPAT. Because the 10 G SerDes block itself has no knowledge of encoding or protocol framing, this CJTPAT is a valid pattern for jitter testing of the 10 G SerDes block.

Table 13. CJTPAT coding

Data name	Encoding
CJPAT	*See the MJSQ T11.2 specification, Annex A (A.2.3.3)*

8.11 PRBS9

This 2^9 pseudo-random sequence is based on the equation shown in the following table.

Table 14. SerDes PRBS9 pattern

Data name	Encoding
PRBS9	f = 1 + x5 + x9

8.12 PRBS7

This SerDes Traditional BIST 27-1 pseudo-random sequence is based on the equation shown in the following table.

Table 15. SerDes PRBS7 pattern

Data name	Encoding
PRBS7	f = 1 + x6 + x7

This table lists the PRBS7 serial data sequence, which contains 127 different states before repeating.

Table 16. 2^7-1 BIST PN sequence

Bit number	Bit value	Bit number	Bit value	Bit number	Bit value	Bit number	Bit value
Bit	Binary	Bit	Binary	Bit	Binary	Bit	Binary

Table continues on the next page...

Table 16. 2^7-1 BIST PN sequence (continued)

Bit number	Bit value	Bit number	Bit value	Bit number	Bit value	Bit number	Bit value
Number	Value	Number	Value	Number	Value	Number	Value
1	1	33	1	65	0	97	1
2	1	34	1	66	0	98	0
3	1	35	1	67	0	99	0
4	1	36	0	68	1	100	0
5	1	37	0	69	1	101	1
6	1	38	1	70	1	102	1
7	1	39	0	71	0	103	0
8	0	40	0	72	0	104	1
9	0	41	0	73	0	105	0
10	0	42	1	74	1	106	0
11	0	43	0	75	0	107	1
12	0	44	1	76	0	108	0
13	0	45	1	77	1	109	1
14	1	46	0	78	0	110	1
15	0	47	0	79	0	111	1
16	0	48	1	80	1	112	0
17	0	49	1	81	1	113	1
18	0	50	1	82	0	114	1
19	0	51	0	83	1	115	1
20	1	52	1	84	1	116	0
21	1	53	0	85	0	117	0
22	0	54	1	86	1	118	1
23	0	55	0	87	0	119	1
24	0	56	0	88	1	120	0
25	0	57	1	89	1	121	0
26	1	58	1	90	0	122	1
27	0	59	1	91	1	123	0
28	1	60	1	92	1	124	1
29	0	61	1	93	1	125	0
30	0	62	0	94	1	126	1
31	0	63	1	95	0	127	0
32	1	64	0	96	1		

8.13 CDR stressor

This is useful for testing the stress on the Clock/Data Recovery Loop. This pattern is constructed of four full-rate ‘1’s followed by four full-rate ‘0’s.

NOTE

This pattern is not a valid 8b10b pattern and has no application at the protocol level.

Table 17. CDR stressor coding

Data name	Encoding
CDRSTRS	Four full-rate ‘1’s; four full-rate ‘0’s

8.14 Half Rate Mixed Frequency Test Pattern (K28.5_h)

This pattern corresponds to the Half Rate K28.5. The idea is to run the block at full speed but the data at half the rate. This test is useful for assessing the impact of changing sample depth.

NOTE

This pattern is not a valid 8b10b pattern and has no application at the protocol level.

Table 18. K28.5_h coding

Data name	10-bit encoding
K28.5_h	000011 1111 111100 1100 111100 0000 000011 0011

8.15 Quarter Rate Mixed Frequency Test Pattern (K28.5_q)

This pattern corresponds to the Quarter Rate K28.5. The idea is to run the block at full speed but the data at quarter the rate. This test is useful for assessing the impact of changing sample depth.

NOTE

This pattern is not a valid 8b10b pattern and has no application at the protocol level. Because of the long strings of ‘1’s and ‘0’s, there may also be issues with baseline wander due to the AC-coupling capacitors and may only be useful during digital loopback.

Table 19. K28.5_q coding

Data name	Encoding
K28.5_q	000000 0011 111111 1111 111111 1100 001111 0000 111111 1100 0000 000000 0000 000011 110000 1111

8.16 PSSPR

This pattern is based on the SSPR pattern described in OIF CEI Short Stress Patterns White Paper. There are some slight modifications to fit the pattern into a 10-bit or 20-bit wide parallel pattern. The original SSPR pattern was chosen to have baseline wander and timing content that are at least as stressful as 10,000 years of random binary at 10 Gbit/s. The pattern has a total length 32,760 bits. All 2^28-1 PRBS28 sequences are generated using taps 25 and 28. Block 1, shown in Table 20, is

5437 bits of PRBS28 with a seed = 28'h008_0080 and begins with 8 x 0, 1, 11 x 0, 1, 12 x 0, 1 … Block 2 CID (Consecutive Identical Digit) contains a 1 followed by 71 x 0. Block 3 is 5437 bits of PRBS28 seed = 28'hFFF_FFFF and begins 28 x 1, 25 x 0, 3 x 1, 22 x 0 … Block 4 takes the same sequence as block 1 (omitting the last 3 bits) and codes it:

- A zero is encoded as a change of output
- A one is encoded as no change of output
- The output before the first bit is assumed to have been a 0
- This block begins 10101010010101010101101010101011011010 …

Blocks 5 to 8 are the inverse of 1 to 4.

Table 20. PSSPR coding

Data name	Encoding							
	Block 1	Block 2	Block 3	Block 4	Block 5	Block 6	Block 7	Block 8
SSPR	PRBS28	CID	PRBS28	PRBS28	PRBS28	CID	PRBS28	PRBS28
	Seed = 28h'008_0080	1, 71 x 0	Seed = 28h'FFF_FFFF	Seed = 28h'008_0080 Diff encoded	Seed = 28h'008_0080	1, 71 x 0	Seed = 28h'FFF_FFFF	Seed = 28h'008_0080 Diff encoded
	5437 bits	72 bits	5437 bits	5434 bits	5437 bits	72 bits	5437 bits	5434 bits

8.17 SATA LBP

This pattern corresponds to the Lone Bit Pattern described in Section 7.2.4.3.5 of the SerialATA Rev 2.6 specification.

Table 21. SATA LBP coding

Data name	Encoding
SATA LBP	*See the SATA Rev 2.6 specification, Section 7.2.4.3.5*

8.18 SATA LTDP

This pattern corresponds to the Low Transition Density Pattern found in Section 7.2.4.3.1 of the SerialATA Rev 2.6 specification.

Table 22. SATA LTDP coding

Data name	Encoding
SATA LTDP	*See the SATA Rev 2.6 specification, Section 7.2.4.3.1*

8.19 SATA High Transition Density Pattern (HTDP)

This pattern corresponds to the High Transition Density Pattern found in Section 7.2.4.3.2 of the SerialATA Rev 2.6 specification.

Table 23. SATA HTDP coding

Data name	Encoding
SATA HTDP	*See the SATA Rev 2.6 specification, Section 7.2.4.3.2*

8.20 SATA Low Frequency Spectral Content Pattern (LFSCP)

This pattern corresponds to the Low Frequency Spectral Content Pattern found in Section 7.2.4.3.3 of the SerialATA Rev 2.6 specification.

Table 24. SATA LFSCP coding

Data name	Encoding
SATA LFSCP	*See the SATA Rev 2.6 specification, Section 7.2.4.3.3*

8.21 Tx tap tester

This pattern is required for testing Tx tap increment and decrement function. This pattern is defined in IEEE Std 802.3ap-2007, Section 72.7.1.11.

NOTE

This pattern is not a valid 8b10b pattern and has no application at the protocol level.

Table 25. Tx tap tester

Data name	Encoding
Tx tap tester	Eight full-rate ‘1’s; eight full-rate ‘0’s

9 Revision history

This table provides a revision history for this application note.

Table 26. Document revision history

Rev. number	Date	Description
2	01/2016	Updated SerDes receive path.
1	06/2015	Added Section 7, "Debugging tips"
0	04/2015	Initial public release

How to Reach Us:

Home Page:
nxp.com

Web Support:
nxp.com/support

Information in this document is provided solely to enable system and software implementers to use NXP products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits based on the information in this document. NXP reserves the right to make changes without further notice to any products herein.

NXP makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does NXP assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. "Typical" parameters that may be provided in NXP data sheets and/or specifications can and do vary in different applications, and actual performance may vary over time. All operating parameters, including "typicals," must be validated for each customer application by customer's technical experts. NXP does not convey any license under its patent rights nor the rights of others. NXP sells products pursuant to standard terms and conditions of sale, which can be found at the following address: nxp.com/SalesTermsandConditions.

While NXP has implemented advanced security features, all products may be subject to unidentified vulnerabilities. Customers are responsible for the design and operation of their applications and products to reduce the effect of these vulnerabilities on customer's applications and products, and NXP accepts no liability for any vulnerability that is discovered. Customers should implement appropriate design and operating safeguards to minimize the risks associated with their applications and products.

Document Number AN5119
Revision 2, 01/2016