TEKTRONIX PHYSICAL LAYER METHOD OF IMPLEMENTATION

USB 3.1 RX

Version 0.9

30 September 2014

NOTICE: This is a living document. Contents are subject to change in subsequent releases, as incremental refinements/ improvements are made, and supplemental material is added.

Tektronix®

TABLE OF CONTENTS

MODIFICATION RECORD

No modifications.

ACKNOWLEDGMENTS

Tektronix would like to acknowledge the efforts of the following individuals in the development of this test suite:

- Randy White
- Jim Dunford
- Pete Tomaszewski

INTRODUCTION

This particular suite of tests has been developed to help implementers evaluate the **USB 3.1** physical layer functionality of their **USB 3.1** products. This test suite is aimed at validating products in support of the work being directed by the **USB-IF**.

These tests are designed to determine if a product conforms to specifications defined in the **USB-IF Universal Serial Bus 3.1 Specification, Version 1.0 – 11 August 2014** (hereafter referred to as the "USB Specification"). Successful completion of all tests contained in this suite does not guarantee that the tested device will successfully operate with other USB products. However, when combined with a satisfactory level of interoperability testing, these tests provide a reasonable level of confidence that the Device Under Test (DUT) will function properly in many USB environments.

This test suite document is part of the support and resources Tektronix provides to its customers to help them determine whether a product conforms to a specification, but does not attempt to provide any legal basis for claims of compliance.

The scope of this v0.9 document is to provide overview, guidance, and information regarding test setup, operation, and best practices for receiver (RX) jitter tolerance tests. These areas include symbol filtering, LFPS and loopback initiation, BERT jitter & stress calibration, and jitter tolerance testing.

The test definitions themselves are intended to provide a high-level description of the motivation, resources, procedures, and methodologies specific to each test. Formally, each test description contains the following sections:

REFERENCES

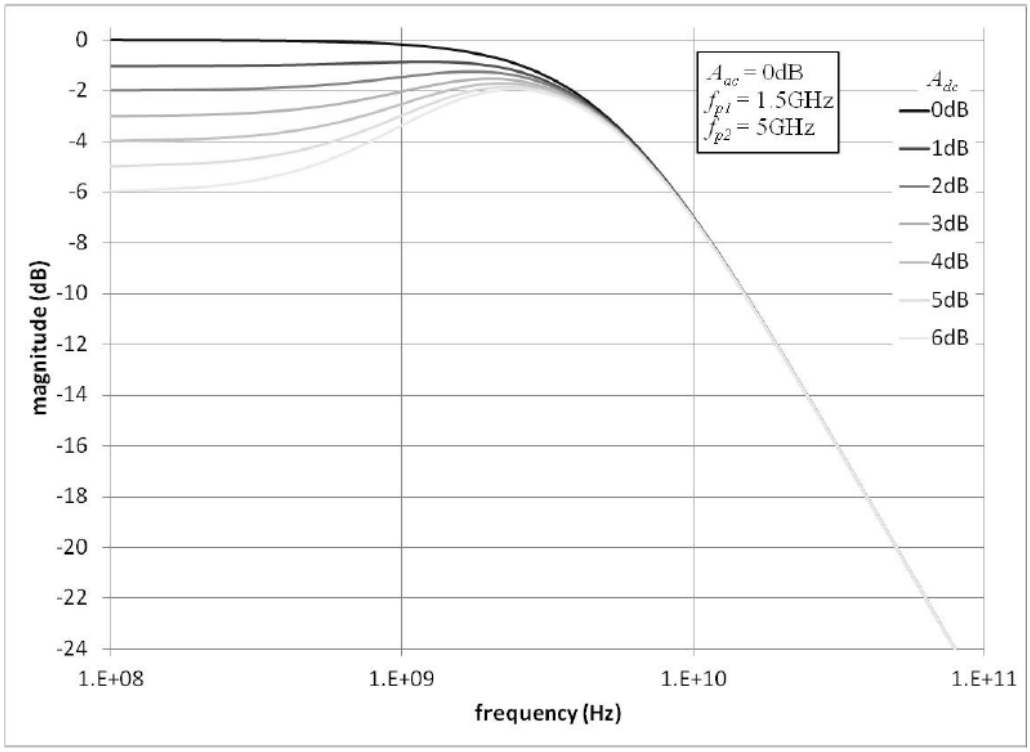
The following documents are referenced in this text:

- Universal Serial Bus 3.1 Specification, v1.0, DTD 11 August, 2014
- Universal Serial Bus Compliance Test Specification v0.1

USB 3.1 RECEIVER JITTER TOLERANCE TEST OVERVIEW

Receiver tests for USB 3.1 are described in section 6.8 of the Universal Serial Bus 3.1 Specification, v1.0, DTD 11 August, 2014. This test verifies that the receiver properly functions in the presence of deterministic and random jitter at multiple frequencies. In order to reduce test time, the receiver is tested to a bit error ratio (BER) of 10-10. This test is performed without cross talk from other links. No receiver testing is done with multiple downstream ports active on hosts/hubs. Loopback training must have all jitter sources on. The receiver test is performed with asynchronous SSC clocks in the test system and the device under test. The test system SSC shall be triangular at the maximum specified SSC frequency (33 kHz) and down-spread -5000 ppm. The test system SSC shall meet the specification limits. For tethered devices replace the 1 meter cable with the short USB 3.1 cable for receiver testing. These cables are provided by the USB-IF with the official test fixture kit. Note: When the DUT is in loopback for this test it shall not exit loopback unless it receives a power on reset, warm reset or an LFPS Exit Handshake.

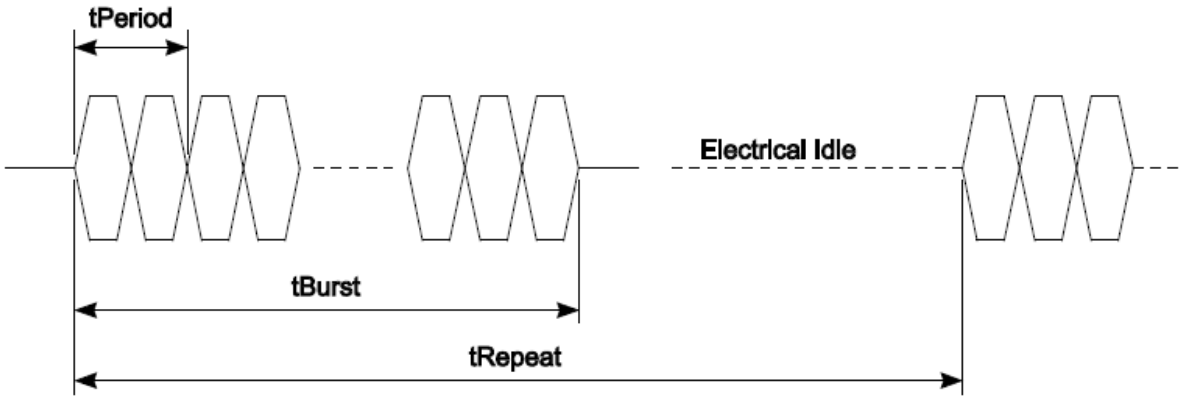
USB 3.1 allows the use of receiver equalization to meet system timing and voltage margins. For long cables and channels the eye at the Rx is closed, and there is no meaningful eye without first applying an equalization function. The Rx equalizer may be required to adapt to different channel losses using the Rx EQ training period. The exact Rx equalizer and training method is implementation specific. Receivers may optimize their performance by sweeping through seven pre-defined Rx EQ settings including CTLE gain and 1-tap DFE.



USB 3.1 LFPS

LFPS entry and exit process for receiver loopback is described in Chapter 7 of the USB 3.1 specification. Receiver loopback must be re-timed. Direct connection from the Rx amplifier to the transmitter is not allowed for loopback mode. The receiver shall continue to process SKPs as appropriate. SKP symbols shall be consumed or inserted as required for proper clock tolerance compensation. Over runs or under runs of the clock tolerance buffers will reset the buffers to the neutral position. During loopback the receiver shall process the Bit Error Rate Test (BERT) commands. Loopback shall occur in the 10-bit domain for Gen 1 operation and in the 132-bit domain for Gen 2 operation. No error correction is allowed. All symbols shall be transmitted as received with the exception of SKP and BERT commands.

Specifically Polling.LFPS is discussed in section 7.5.4.3 of the USB 3.1 specification. The BERT will have the capability to generate compliant USB 3.1 LFPS signaling to initiate port loopback, then provide a calibrated 10 Gb/s test signal for RX jitter tolerance testing.



USB 3.1 SYMBOL FILTERING

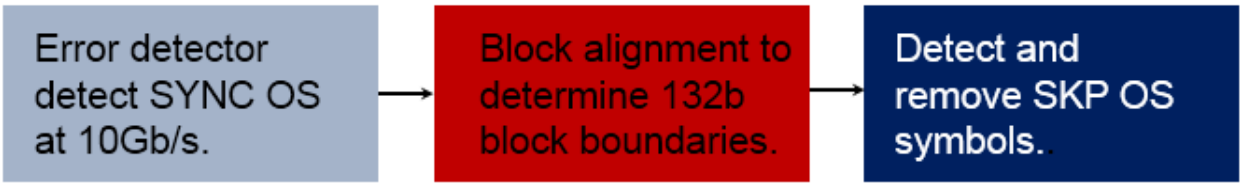
A USB 3.1 host or device has the ability to dynamically insert skip ordered set (SKP OS) symbols in the presence of SSC to keep the bit rate constant over time. The BERT must have the ability to both provide these symbols for the purpose of keeping the receiver in lock, and removing these symbols while detecting errors in the bit payload. Section 6.4.3.2 of the USB 3.1 spec provides details related to SKP OS rules when using 128b/132b encoding. Several of these rules include:

1. Granularity for which SKP Symbols can be added or removed by a Port is two symbols. A port may add or remove more than 2 SKP symbols, but the number of SKP symbols that is added or removed shall be a multiple of two.
    a. This includes re-timers within the signal path.
2. A receiver (or BERT error detector) may receive a SKP OS with anywhere from 0 to 36 SKP symbols with the number of SKP symbols being a multiple of two. A SKP OS with 0 SKP symbols has only a SKPEND symbol followed by the three symbols that describe the LFSR state. Another impact of receiving variable length SKP OS is that a receiver is always allowed to add up to 12 SKP symbols to any SKP OS regardless of the length of the received SKP OS.
3. The SKPEND Symbol indicates the last four Symbols of SKP Ordered Set so that receivers can identify the location of the next Block Header in the bit stream. The three Symbols following the SKPEND Symbol contain different information depending on the LTSSM state.
4. A receiver (or BERT error detector) must always perform single bit error correction on the SKP and SKPEND (and all other special) symbols. However, since the Hamming distance between the SKP and SKPEND symbols is 8, once a receiver has determined that it is dealing with a non-empty SKP OS (by proper detection of a first SKP symbol) it may be beneficial to use multiple bit (up to 3-bit) error correction in differentiating between a SKP and a SKPEND symbol.

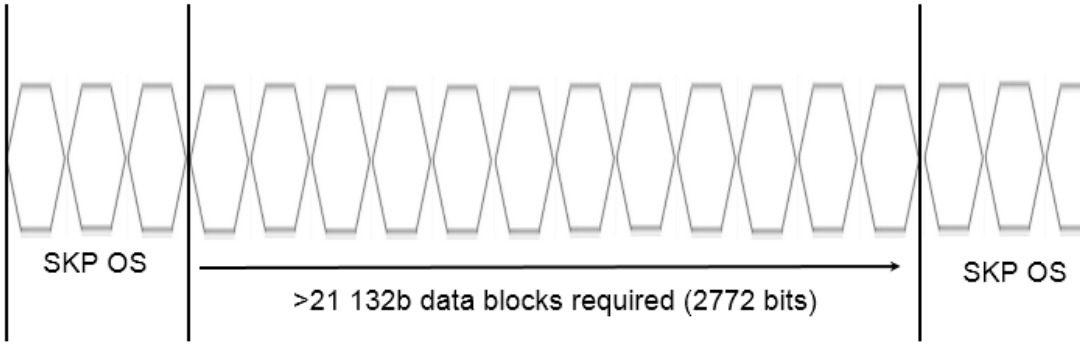
According to Table 6-12 of the USB 3.1 specification, there are multiple combinations of SKP OS that need to be handled (e.g. variable SKP sets). The BERT must support filtering of SKP OS's to differentiate between stress pattern payload bits and those dynamically inserted by the DUT.

Symbol Number	Value	Description
0 through 2*N-1 [N can be 0 through 18]	CCh	SKP Symbol Symbol 0 is the SKP Ordered Set Identifier Note: for an empty SKP OS, the first symbol will be a SKPEND.
2*N	33h	SKPEND Symbol
2*N+1	00-FFh	(i) If prior block was a Data Block: Bit[7] = Even Data Parity Bit[6:0] = LFSR[22:16] (ii) Else: Bit[7] = ~LFSR[22] Bit[6:0] = LFSR[22:16]
2*N+2	00-FFh	(i) If LTSSM state is Polling.Compliance: Error_Status[7:0] (ii) Else LFSR[15:8]
2*N+3	00-FFh	(i) If LTSSM state is Polling.Compliance: ~Error_Status[7:0] (ii) Else LFSR[7:0]

The BERT performs SKP OS filtering in a two-step process. First, the BERT performs a block alignment using SYNC ordered set (SYNC OS) symbols using the error detector. This block alignment frames each 132b data block to determine when each 132b data block starts and ends. The BERT requires SYNC OS symbols to be present at the error detector input for SKP OS filtering to be functional. Second, the BERT searches for valid SKP OS sequences within each 132b block, and removes these SKP OS bits when detecting bit errors.



The BERT requires a minimum of (21) 132b data blocks between successive SKP OS events. This minimum number ensures that all combinations of SKP OS are removed by the error detector at 10 Gb/s speeds. This minimum number also matches the minimum specified for a USB 3.1 receiver in the electrical spec.



USB 3.1 BERT STRESS CALIBRATION

BERT calibration is required to ensure accurate receiver test under conditions of stress. Table 6-27 in the USB 3.1 spec lists the stress parameters under which the RX should be tested. The methodology is similar to USB3.0, with the following differences:

- Test through a 1-meter cable (ISI channel) versus through 3-meter cable.
- Test SJ @ frequencies 4 MHz and 100 MHz
  - In addition to 500 kHz, 1 MHz, 2 MHz, PJ_F1, and 50 MHz
- Pre-shoot and de-emphasis used, versus de-emphasis only

Symbol	Parameter	Gen 1	Gen 2	Units	Notes
f1	Tolerance corner	4.9	7.5	MHz	
J_{Rj}	Random Jitter	0.0121	0.01	UI rms	1
$J_{Rj\_p\text{-}p}$	Random Jitter peak- peak at 10^{-12}	0.17	0.141	UI p-p	1,4
J_{Pj_500kHz}	Sinusoidal Jitter	2	4.76	UI p-p	1,2,3
J_{Pj_1Mhz}	Sinusoidal Jitter	1	2.03	UI p-p	1,2,3
J_{Pj_2MHz}	Sinusoidal Jitter	0.5	0.87	UI p-p	1,2,3
J_{Pj_4MHz}	Sinusoidal Jitter	N/A	0.37	UI p-p	1,2,3
J_{Pj_f1}	Sinusoidal Jitter	0.2	0.17	UI p-p	1,2,3
J_{Pj_50MHz}	Sinusoidal Jitter	0.2	0.17	UI p-p	1,2,3
J_{Pj_100MHz}	Sinusoidal Jitter	N/A	0.17	UI p-p	1,2,3
V_full_swing	Transition bit differential voltage swing	0.75	0.8	V p-p	1
V_EQ_level	Non transition bit voltage (equalization)	-3	Pre=2.2 Post= -3.1	dB	1

Notes:

1. All parameters measured at TP1. The test point is shown in Figure 6-18.
2. Due to time limitations at compliance testing, only a subset of frequencies can be tested. However, the Rx is required to tolerate Pj at all frequencies between the compliance test points.
3. During the Rx tolerance test, SSC is generated by test equipment and present at all times. Each J_{Pj} source is then added and tested to the specification limit one at a time.
4. Random jitter is also present during the Rx tolerance test, though it is not shown in Figure 6-1
5. The JTOL specs for Gen 2 comprehend jitter peaking with re-timers in the system and has a 25 dB/decade slope.

Jitter calibration steps:

1. Performing timing calibration to optimize generator delay (Note, for early development or if PHY only (no controller, manual loopback control) remove switch from setup. In other words, connect DPP output directly to CR).

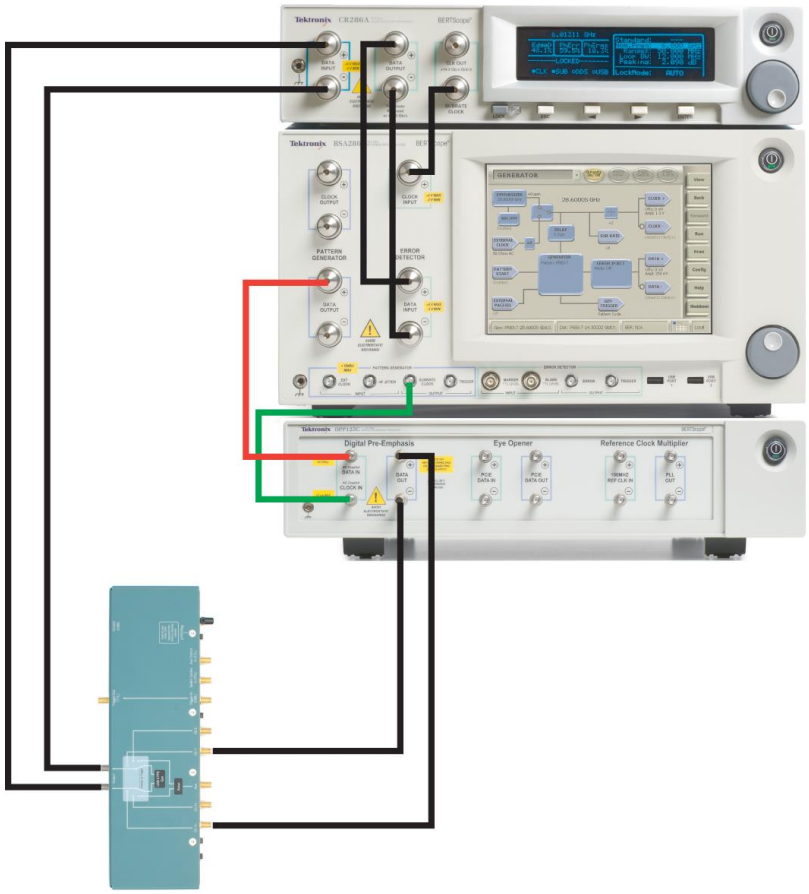


Figure 1. Timing Calibration Setup

2. Connect the receiver test signal source (BERT) directly to a high-speed oscilloscope using phase matched SMA cables for calibration of de-emphasis and pre-shoot. SSC is off for calibration. CTLE is off for de-emphasis calibration.

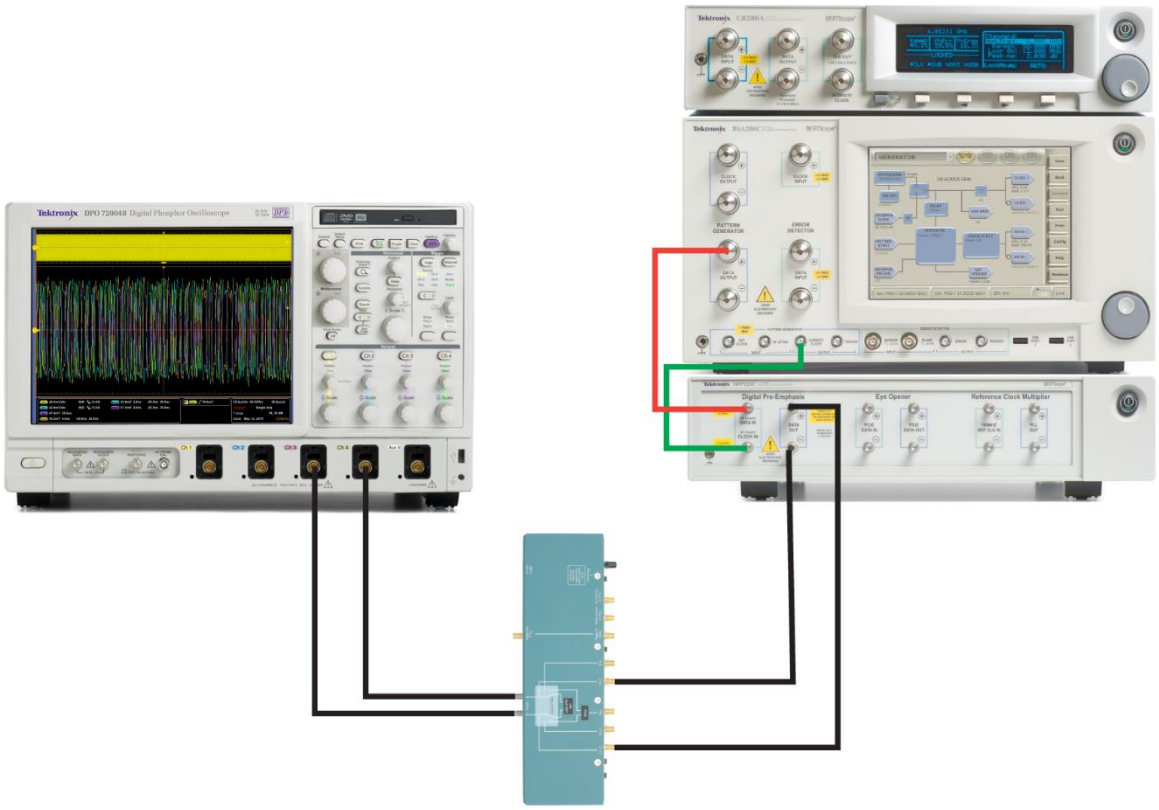


Figure 2. Transmitter Launch Amplitude and Equalization Calibration setup

3. Calibrate the differential amplitude of the measured signal to 800mV p-p.

4. Calibrate de-emphasis to -3.1dB +5/-0% dB, and pre-shoot to +2.2 +/-0.5dB with the signal source set to full bit -3.1dB de-emphasis / +2.2dB pre-shoot using CP9 pattern.

5. Calibrate RJ and SJ as follows:
    a. Calibration is performed using DPOJET using the following patterns:
        i. CP9 random data pattern is used for all SJ calibration steps.
        ii. CP10 clock pattern is used for RJ calibration

    b. Calibrate RJ (1.0 +/-10% ps rms / 12.73 +/-10% ps p-p at a BER of 10^{-10}) with clock pattern CP10. SSC and all other noise sources are off for this step.

c. Calibrate SJ frequencies in the table below to +0/-10% tolerance using CP9. (Calibration is done by testing measured maximum peak to peak jitter without extrapolation (measured TJ) without Sj and then adding Sj until measured maximum peak to peak jitter without extrapolation (measured Tj) increased by the required amount). All other noise sources are off during this calibration. Note, a low frequency JTF setting of 50 kHz is applied so as to allow low frequency SJ to pass through but filter SSC.

Frequency	SJ
500 kHz	476ps
1 MHz	203ps
2 MHz	87ps
4 MHz	37ps
7.5 MHz	17ps
50 MHz	17ps
100 MHz	17ps

Table 1. SJ Frequencies

6. Connect the receiver test signal source to a high-speed oscilloscope through the compliance channel.
   a. 1-meter USB cable for USB 3.1 and required Host or Device calibration channel
   (-14.5 dB loss, recommend using 31 inch trace on BSA12500ISI board)

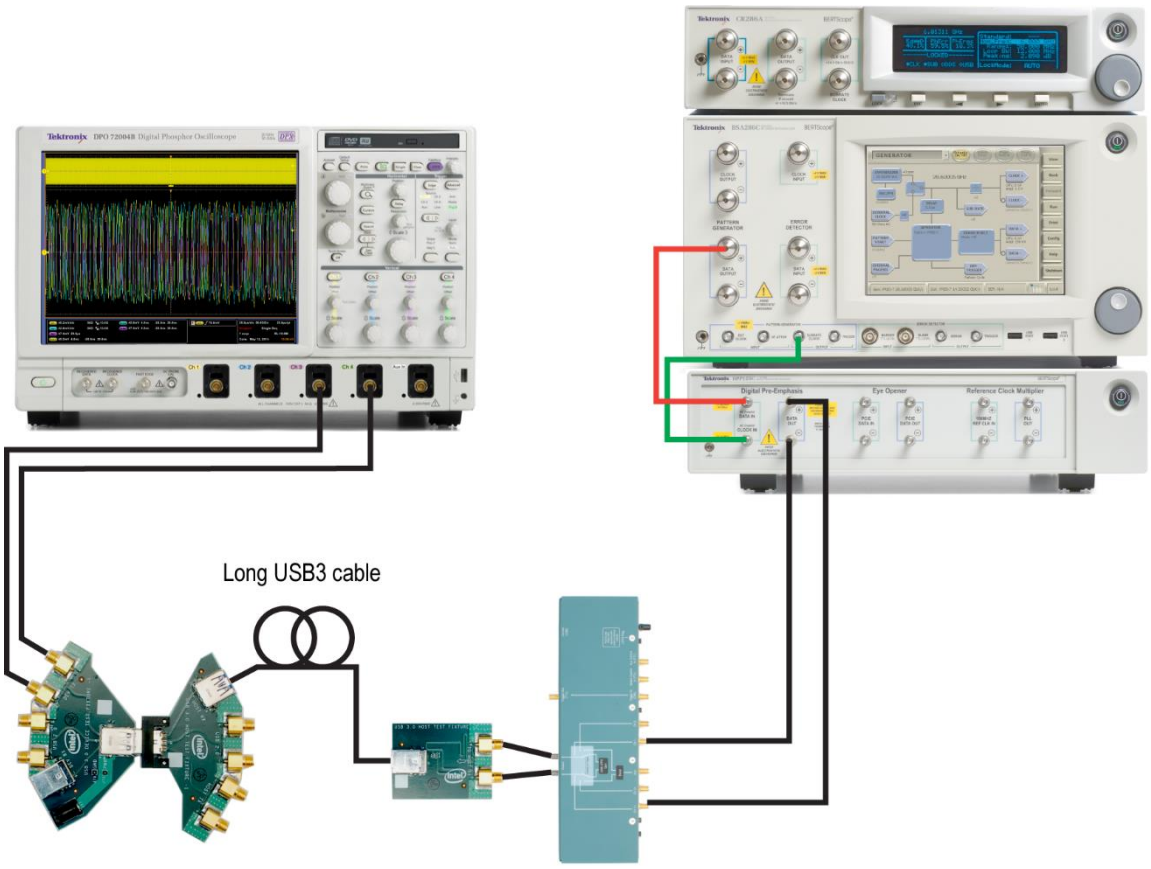


Figure 3. Host Stress Calibration Setup

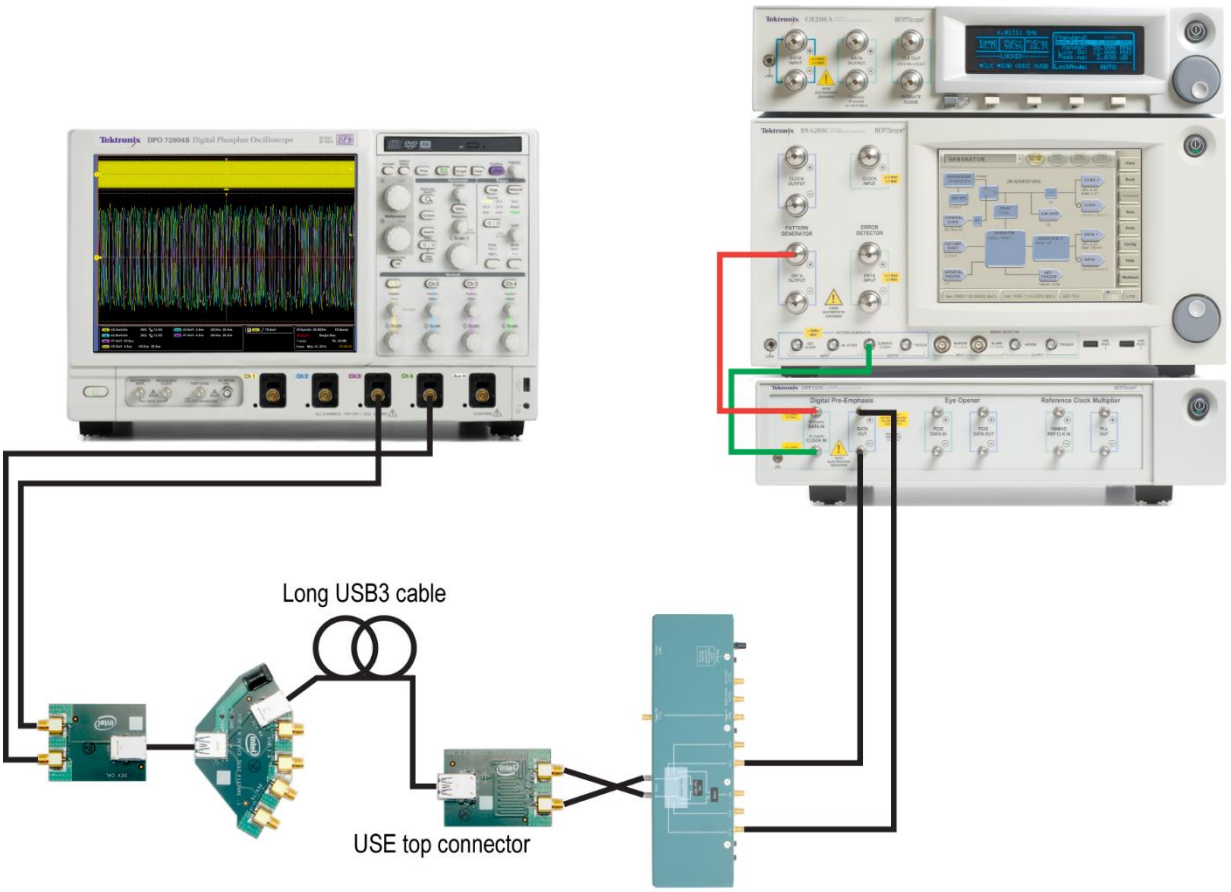


Figure 4. Device Stress Calibration Setup

7. Calibrate eye width using DPOJET and SDLA as follows:
    a. Measure eye width with CP9 at a BER of 10^{-12} at the end of the channel with the following conditions:
        i. RJ and 100 MHz SJ on.
        ii. SSC on
        iii. Apply the compliance JTF (7.5 MHz) and CTLE (as determined from 100 MHz SJ cal).
        iv. Adjust the signal source Tx EQ dB de-emphasis to provide an eye width of 50 ps +/-3ps
        v. Using SDLA analyze CP9 waveform with the below CTLE settings and a 1-tap DFE. This is automatically configured by using the 'USB 3.1 Gen2' preset in SDLA.

Aac =	0	db	1	(linear)		fz =	(Adc/Aac) x f	
fp1 =	1.5	GHz		Bit Rate =	10	Gb/s	DFE Taps	1
fp2 =	5.0	GHz		JTF BW =	7.5	MHz	Autoset V =	Enabled
Adc =	0	db	1.00	(linear)		fz0 =	1.50E+00	
	1	db	0.89	(linear)		fz1 =	1.34E+00	
	2	db	0.79	(linear)		fz2 =	1.19E+00	
	3	db	0.71	(linear)		fz3 =	1.06E+00	
	4	db	0.63	(linear)		fz4 =	9.46E-01	
	5	db	0.56	(linear)		fz5 =	8.44E-01	
	6	db	0.50	(linear)		fz6 =	7.52E-01	

SDLA CTLE Settings

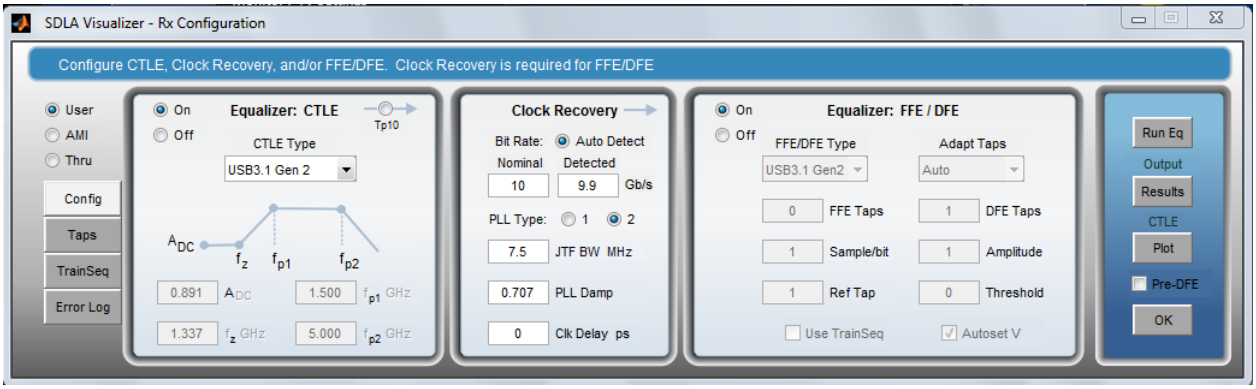


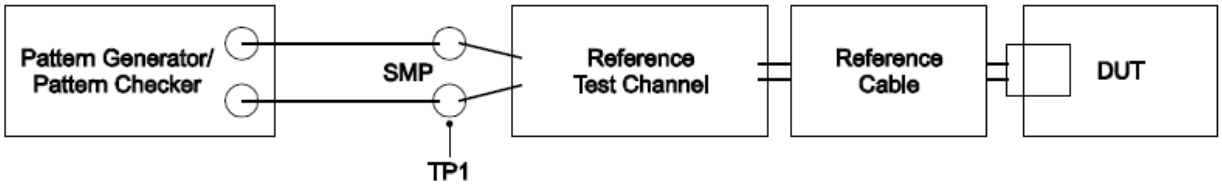
Figure 5. SDLA Rx Block Configuration

8. Calibrate eye height using DPOJET as follows:
   a. Measure eye height with CP9 at a BER of 10^{-12} at the end of the channel with the following conditions:
      i. Calibrated RJ and 100 MHz SJ on.
      ii. SSC on
      iii. Apply the compliance JTF (7.5 MHz) and CTLE (as determined from 100 MHz SJ cal).
      iv. Adjust the signal source amplitude to provide the following:
         1. For Device testing adjust amplitude to provide 100mV +5mV/-0mV of eye height.
         2. For Host testing adjust amplitude to provide 120mV +5mV/-0mV of eye height.
9.

Note: Amplitude should be calibrated to be as close to the minimum value as possible w/o going under the minimum value.
Note: De-emphasis & pre-shoot values at the instrument output must remain at -3.1dB +5/-0% and 2.2dB +5/-0% after the calibration process is complete.

USB 3.1 JITTER TOLERANCE TESTING

Per section 6.8.5 of the USB 3.1 specification, the receiver tolerance test is performed using the appropriate compliance reference channel for Gen 1 or Gen 2 operation depending upon the rate being tested. A pattern generator shall send the rate appropriate compliance test pattern with added jitter through the compliance reference channels to the receiver. The receiver shall loop back the data and any difference in the pattern sent from the pattern generator and returned will be an error. When running the compliance tests, the receiver shall be put into loopback mode.



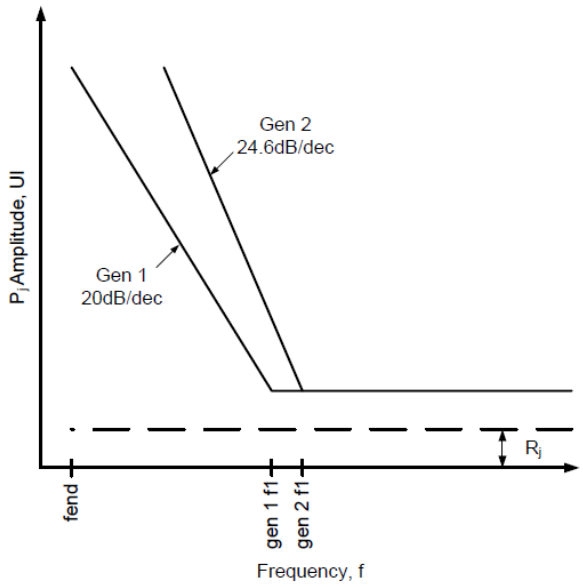
Loopback shall occur in the 10-bit domain for Gen 1 operation and in the 132-bit domain for Gen 2 operation. No error correction is allowed. All symbols shall be transmitted as received with the exception of SKP and BERT commands.

Overview of receiver test steps:
The test runs in the Polling.Loopback substate.

1. Set receiver test signal output to calibrated values.
2. Power on the device under test.
3. Connect the DUT to the receiver test signal source through the USB-IF receiver compliance channel.
4. Vbus shall be on for Device testing. The channel to the test equipment receiver is kept as short and clean as possible.
5. TransmitPolling.LFPS until SCD1.LFPS handshake is established.
6. TransmitPolling.LFPSPlus until SCD2.LFPS handshake is established.
7. TransmitPolling.PortMatch until PHY Capability LBPM handshake is established.
8. TransmitPolling.PortConfig until PHY Ready LBPM handshake is established. Note that all jitter sources are added during all transmissions to the device under test.
9. Transmit 524,288 TSEQ control blocks (132-bits/block).
10. Transmit 256 TS1.
11. Transmit 256 TS2 with loopback bit set.
12. Start transmitting the BDAT test pattern.
13. Transmit the BDAT sequence from the signal source for a total of 3x109 symbols (3xE10 bits). A single SKP ordered set is inserted in the sequence every xx data blocks.
14. Repeat test for all values of Sj listed in Table 6-27 of the USB 3.1 specification.
15. The DUT fails if more than one error is encountered.

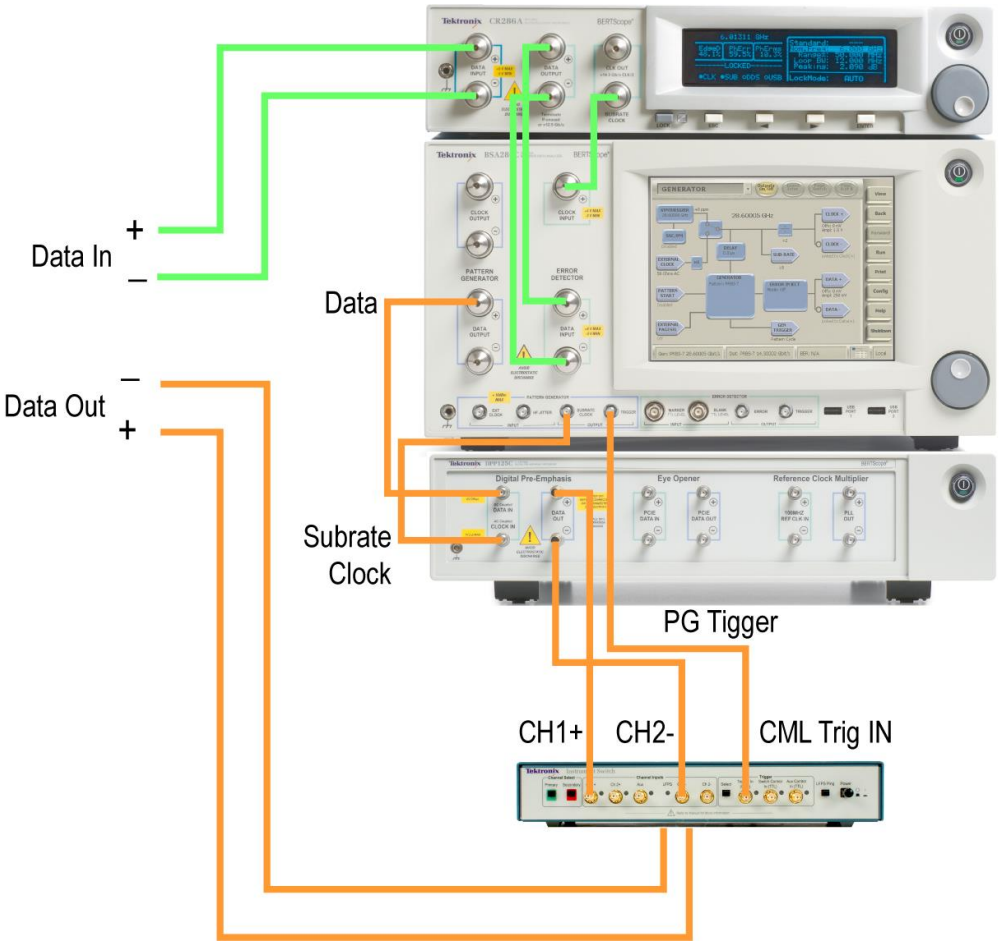
If the device does not go into loopback it fails the test. If a DUT drops out of loopback during or after a specific test point retrain the link with jitter at the SJ test frequency and retest.

Note: For steps 4-8 the specific number of sequences are intended to provide a nominal guideline for receiver loopback training. Some hosts or devices may support test modes that can force loopback without LFPS negotiation. These test modes may be used for characterization, but cannot be used for compliance verification.



Appendix A: BERTScope Loopback Implementation

1. DUT Rx is open -> BERTScope transmits temp pattern and ED is set to AllZeros patterns (and is 'locked' since there is no signal with a ~50mV threshold)
2. DUT Tx connected to fixture/CR input ->CR passes data output to ED input and starts showing errors (any signal will trigger errors when comparing AllZeros pattern). Not starting at pass through mode but rather LFPS idle.
3. ED recognizes LFPS activity -> ~80us later PG Trigger is asserted (trigger starts LFPS)
4. Transmit LFPSPlus (SCD1/2, LBPM), LFPS messaging timing based on midpoint of max/min range
5. Switch in Pass Through Mode (done automatically) -> Data Out now transmits TSEQ and then TS1/TS2
6. DUT receives TS1/TS2 -> DUT enters Loopback
7. DUT_Loopback = true -> CR detects CP9 pattern
8. CP9 Pattern detected -> ED compares to user pattern for ED Sync (If DUT_Mode = Tx compliance-> No sync, ElseIf DUT_Mode = Rx_Loopback -> Sync +SKP filtering)



Appendix B: Conversion of DPP Tap Settings

Computation of DPP tap gains required to implement specified levels of pre-shoot and/or de-emphasis is not entirely trivial. For those not already familiar with the concepts and definitions of pre-shoot and de-emphasis as applied to digital signaling it can be rather confusing. This short note hopes to make things a bit more clear.

Figure 1 on the next page shows that pre-shoot and de-emphasis are defined in terms of signal voltage levels.

In the plot on the right side of the differential waveforms (blue and red signals), the level Vb occurs in the middle of a string of at least 3 positive ones (or it could be negative ones, but we'll assume they are positive ones here). The output level is then equal to the sum of all three tap gains. Using the tap naming convention in figure 1,

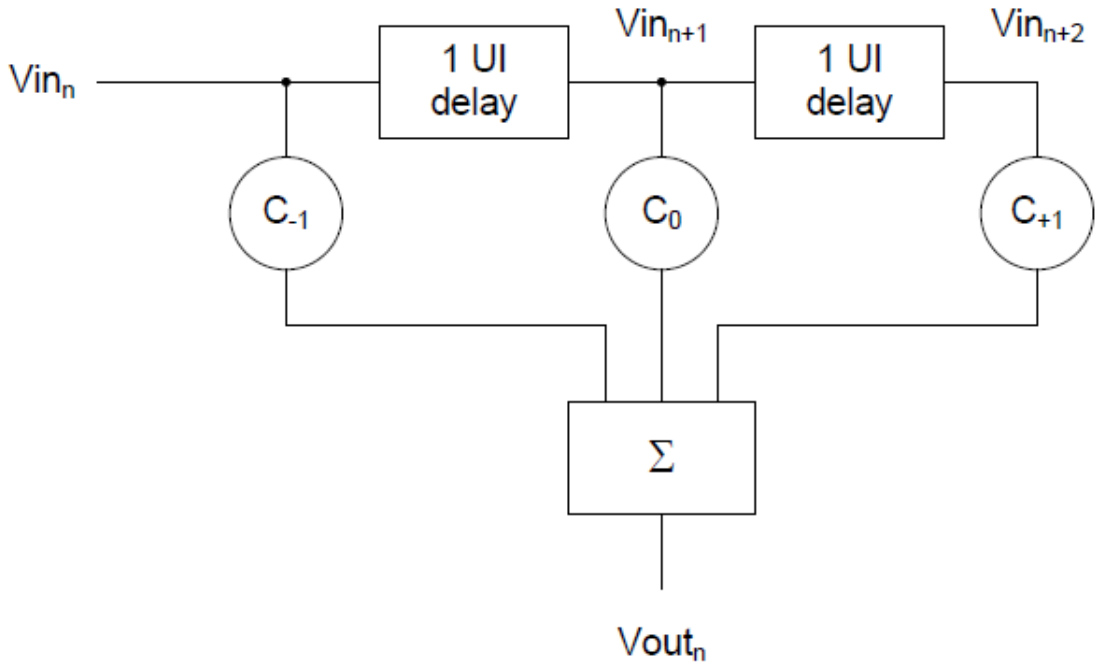
$$V_b = C_{-1} + C_0 + C_1 \quad (1)$$

Pre-shoot (V_c) occurs at the end of the period of ones, where we have a single negative one present at the input to the first FIR delay, but it is not yet clocked through that delay. The output during pre-shoot is

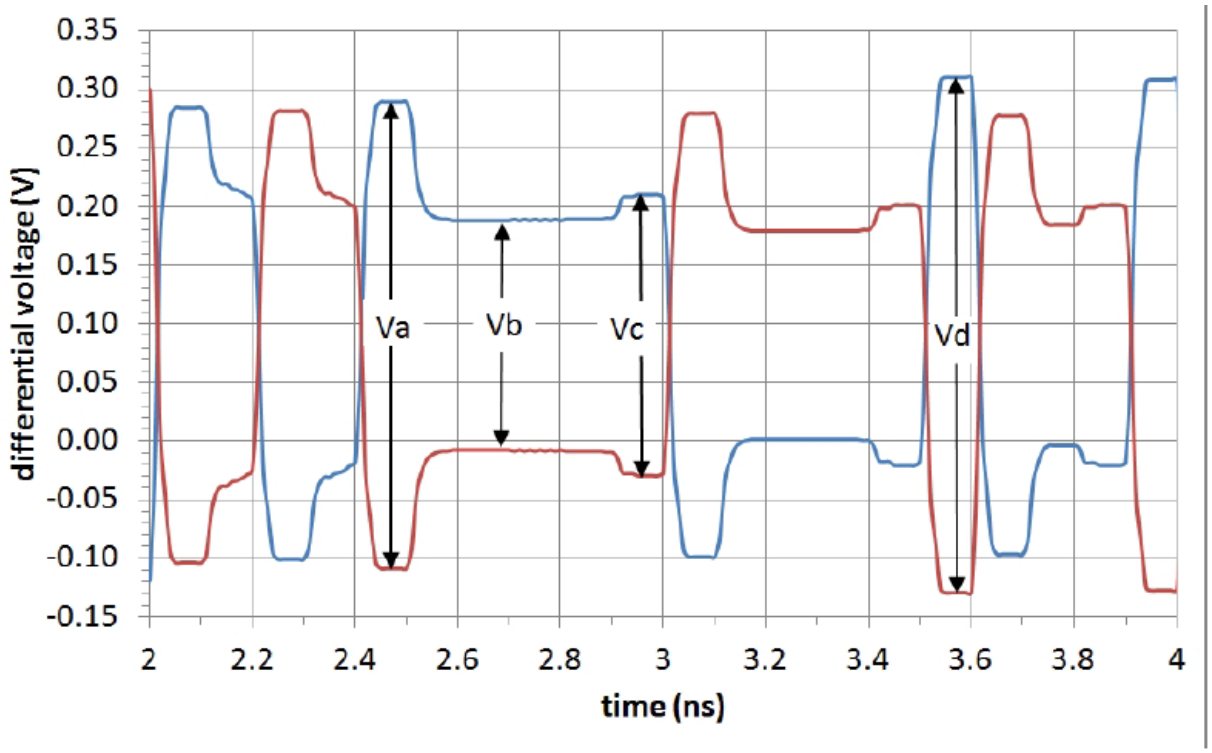
$$V_c = -C_{-1} + C_0 + C_1 \quad (2)$$

In the example graph, the pre-shoot ratio is greater than one.
De-emphasis occurs as a string of positive ones is being shifted into the taps but before the last tap is populated with a positive one (i.e. that tap is still at -1). The output during de-emphasis is

$$V_a = C_{-1} + C_0 - C_1 \quad (3)$$



$$Vout_n = \sum_{n=1}^{3}(Vin_n c_n) \quad \text{and} \quad \sum_{n=1}^{3} | c_n | = 1$$



Preshoot = 20log(Vc/Vb)
De-emphasis = 20log(Vb/Va)

Figure 6. Definition of Pre-shoot and De-emphasis

In the example graph, the de-emphasis ratio is less than one.
Given pre-shoot and de-emphasis in dB, it is easy to calculate the corresponding linear ratios.

$$P = 10^{0.05\,P_{dB}} \quad (4)$$

$$D = 10^{0.05\,P_{dB}} \quad (5)$$

Now, the linear ratios for pre-shoot and de-emphasis are easily written.

$$D = \frac{V_A}{V_B} = \frac{C_{-1}+C_0+C_1}{C_{-1}+C_0-C_1} \quad (6)$$

Appendix C: Taking the Mystery out of DPP Tap Settings

Computation of DPP tap gains required to implement specified levels of pre-shoot and/or de-emphasis is not entirely trivial. For those not already familiar with the concepts and definitions of pre-shoot and de-emphasis as applied to digital signaling it can be rather confusing. This short note hopes to make things a bit more clear.

Figure 1 shows that pre-shoot and de-emphasis are defined in terms of signal voltage levels. One crucial detail is that the signal levels are assumed to be {+1, -1}; they are NOT {+1, 0}; Get this detail wrong and it will be very difficult to understand FIR tap settings!

In the plot on the right side of Figure 1, the level Vb occurs in the middle of a string of at least 3 positive ones (or it could be negative ones, but we'll assume they are positive ones here). The output level is then equal to the sum of all three tap gains. Using the tap naming convention in Figure 1,

$$V_b = C_{-1} + C_0 + C_1 \quad (1)$$

Pre-shoot (Vc) occurs at the end of the period of ones, where we have a single negative one present at the input to the first FIR delay, but it is not yet clocked through that delay. The output during pre-shoot is

$$V_c = -C_{-1} + C_0 + C_1 \quad (2)$$

In the example graph, the pre-shoot ratio is greater than one.
De-emphasis occurs as a string of positive ones is being shifted into the taps but before the last tap is populated with a positive one (i.e. that tap is still at -1). The output during de-emphasis is

$$V_a = C_{-1} + C_0 - C_1 \quad (3)$$

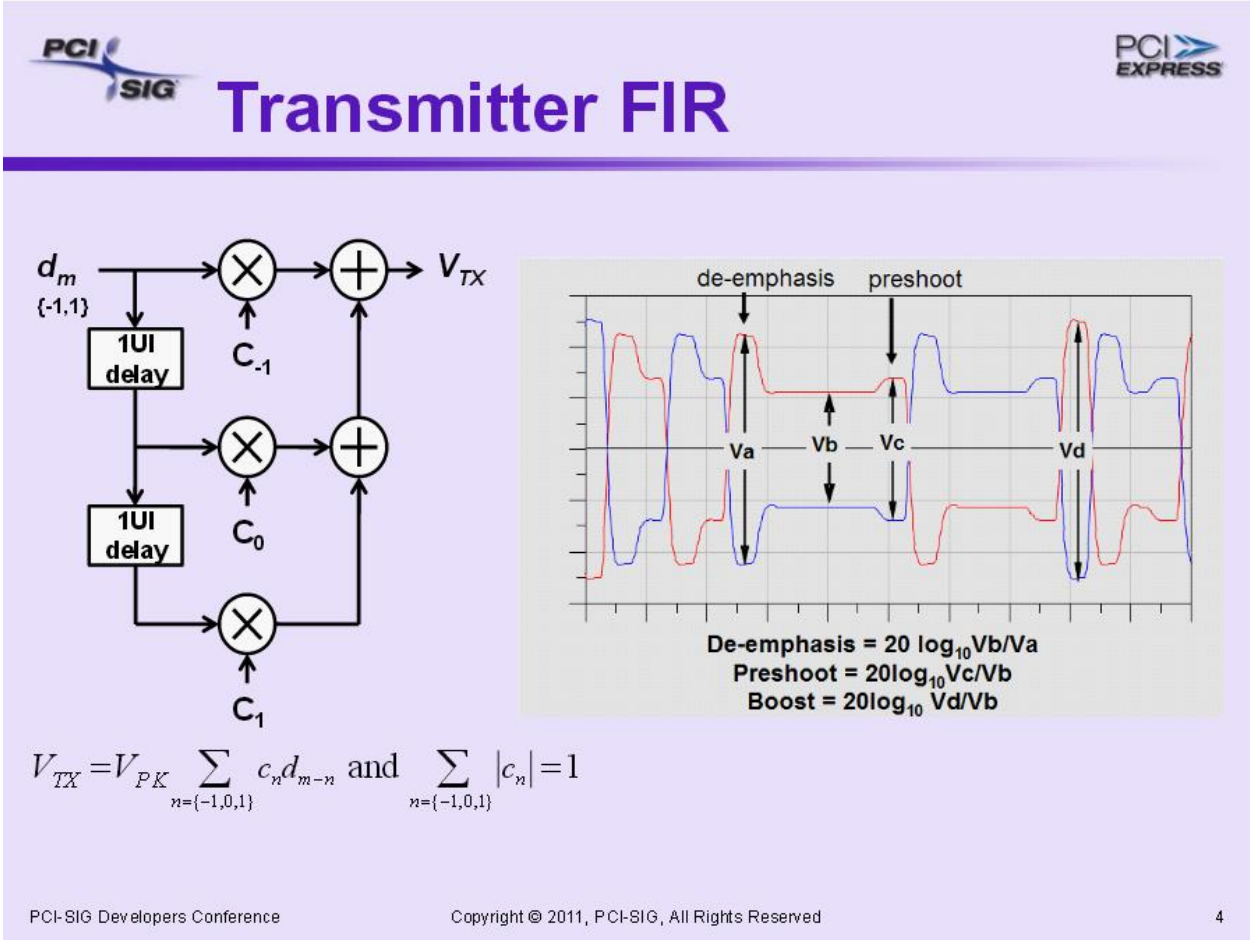


Figure 1. Definition of Pre-shoot and De-emphasis.

In the example graph, the de-emphasis ratio is less than one.
Given pre-shoot and de-emphasis in dB, it is easy to calculate the corresponding linear ratios.

$$P = 10^{0:05\ P_{dB}} \quad (4)$$

$$D = 10^{0:05\ D_{dB}} \quad (5)$$

Now, the linear ratios for pre-shoot and de-emphasis are easily written.

$$\begin{aligned} D &= \frac{V_b}{V_a} = \frac{C_{-1} + C_0 + C_1}{C_{-1} + C_0 - C_1} \\ P &= \frac{V_c}{V_b} = \frac{-C_{-1} + C_0 + C_1}{C_{-1} + C_0 + C_1} \end{aligned} \tag{7}$$

To determine tap values from pre-shoot and de-emphasis ratios, there are two equations but three unknowns. One easy solution here is to pick an arbitrary value for the sum of the tap values (V_b).

$$V_b = 1 \tag{8}$$

Now we have a system of three equations in three unknowns:

$$\begin{bmatrix} 1-D & 1-D & 1+D \\ P+1 & P-1 & P-1 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} C_{-1} \\ C_0 \\ C_1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \tag{9}$$

Alternatively, the third equation could simply specify the value of the cursor tap, $C_0 = 1$. Then we have these equations:

$$\begin{bmatrix} 1-D & 1-D & 1+D \\ P+1 & P-1 & P-1 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} C_{-1} \\ C_0 \\ C_1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \tag{10}$$

In this particular case (specified value for cursor tap), it is feasible to work out some closed-form expressions for tap values as a function of pre-shoot and de-emphasis gains. This is an exercise left to the reader.

Here are a couple of examples that can be verified against other sources.
For a pre-shoot value of 3.00dB and de-emphasis of -10.00dB, P = 1:4125 and D = 0:3162. The tap value equation is

$$\begin{bmatrix} 0.6838 & 0.6838 & 1.3162 \\ 2.4125 & 0.4125 & 0.4125 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} C_{-1} \\ C_0 \\ C_1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

The solution to this is

$$\begin{bmatrix} C_{-1} \\ C_0 \\ C_1 \end{bmatrix} = \begin{bmatrix} -0.0902 \\ 1 \\ -0.4726 \end{bmatrix}$$

For a second example, take a pre-shoot of 6.00dB and de-emphasis o -4.00dB.

$$\begin{bmatrix} 0.3690 & 0.3690 & 1.6310 \\ 2.9953 & 0.9953 & 0.9953 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} C_{-1} \\ C_0 \\ C_1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

The solution to this is

$$\begin{bmatrix} C_{-1} \\ C_0 \\ C_1 \end{bmatrix} = \begin{bmatrix} -0.2780 \\ 1 \\ -0.1634 \end{bmatrix}$$