High-speed Serial Bus Repeater Primer

Re-driver and Re-timer Micro-architecture, Properties, and Usage

Revision 1.2, Oct. 2015

By Samie B. Samaan, Dan Froelich, and Samuel Johnson, Intel Corporation

1. Introduction

1.1 Target Audience

This primer aims to inform High-speed Serial bus Signal Integrity engineers and system designers on the characteristics of SerDes repeaters (re-drivers and re-timers), in order to enable proper and effective use of those devices. It also provides useful background and insights for hardware validation and debugging practitioners.

1.2 Motivation and Scope

Multi-gigahertz serial links, with progressively increasing bitrates, suffer from signal distortion (Inter-symbol Interference, or ISI) due to PCB and package copper and dielectric losses. Other distortions also occur due to impedance discontinuities in the channels, such as vias, connectors, and packages. While a Serializer-Deserializer (SerDes) receiver (Rx for short) is designed to compensate for most of these distortions, and create an internal eye open enough for reliable sampling, bit rates are rising faster than Rx, PCB, or package High Density Interconnect (HDI) technologies can keep up with. The result is that total channel reach is decreasing. Using expensive PCB materials to reduce loss will be reaching its dielectric limits soon, and adds significant cost. Next generation buses, such as PCI Express (PCIe) 4.0 (16 Gb/s) and USB3.10 (10 Gb/s), which aim to double bit rates, will support shorter channel lengths (with similar high volume PCB materials) than their existing 8 Gb/s and 5 Gb/s predecessors, respectively.

For the above reasons, and also various OEMs' desires for differentiation through larger/modular systems, entailing longer channels, there seems to be a rising need for active devices –or **Repeaters**-- which "restore" the signals mid-flight, thus extending channel reach. This paper describes the two main classes of SerDes repeaters: The analog "**Re-drivers**", and the mixed-signal (analog/digital) "**Re-timers**" (both protocol-aware, and protocol-un-aware re-timers).

This work describes the internal micro-architecture, properties, applications, and limitations of both types of SerDes repeaters. It explains the subtle reasons why the presence of a re-driver in certain busses which require adaptive transmitter (Tx) Linear Equalization (TxEQ), such as PCIe 3.0, and 10G-KR, causes such links to be non-openly interoperable. It also addresses the special case of so-called "Closed" Systems which do not require open interoperability (Section 10.1). Furthermore, this document describes the functionality of PCIe 3.0 and 10G-KR re-timers, in detail.

1.3 Signal Distortion in High-speed Copper Links

Present multi-Gigahertz serial busses carry bit streams ranging in their rates from around 1 Gb/s and up to (or exceeding) 32 Gb/s. The fundamental (Nyquist) or clock frequency of such patterns is half the bit rate. PCBs, packages, cables, and connectors make up the majority of copper-based serial links, due to their cost effectiveness. Optical links are also used at high data rates and long reach, but they have different distortion mechanisms, and are not addressed directly here. Copper (or metal) suffers from a resistance which increases with the square root of frequency, due to skin effect [1]. In addition, PCBs have losses in the dielectrics (resin and glass) used to make them [1]. The dielectric loss increases approximately linearly with frequency. Hence, depending on the dielectric loss constant (Dissipation Factor (Dk), a.k.a. tan(δ)), a PCB's trace loss ranges from having square root to linear dependence on frequency. For present day FR4 PCBs (whose Dk might range from 0.03 to 0.015) , the loss is dominated by the square root function at low frequencies, then becomes dominated by the linear dielectric loss at higher frequencies, as seen in Figure 1.

Increasing attenuation with frequency causes signal distortion, and such distortion is worse for longer lines. When higher frequency components are attenuated, the bits (Unit Intervals, or UIs) begin to lose their sharpness, their tails extend beyond any single symbol (or bit), and start spilling over (interfering) with subsequent symbols, hence the term *Inter-symbol Interference (ISI)*.

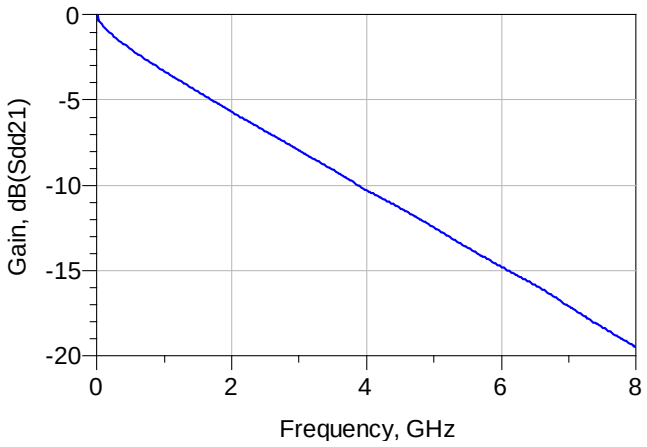

Figure 1 Measured differential Insertion Loss (Gain to be precise) of a terminated PCB strip-line trace, showing an initial square-root-like dependence on frequency up to ~0.5 GHz (skin-loss-dominated), then a linear dependence (dielectric-dominated).

In addition to attenuation, there are also internal reflections in a channel, due to impedance discontinuities, causing further signal distortion. The impedance discontinuities in the channel are caused by inter-layer via transitions, connectors, decoupling capacitor parasitics, packages, imperfect terminations, etc. The insertion loss of such channels (e.g. Figure 2) is usually more complex than that of a simple transmission line.

It is noteworthy that a significant fraction of the smooth (frequency-dependent) distortion due to channel loss could be compensated for by using simple equalization circuits in the transmitter or the

receiver, whereas distortion caused by reflections is usually compensated by the more complex *Decision Feedback Equalization (DFE)* technique [2].

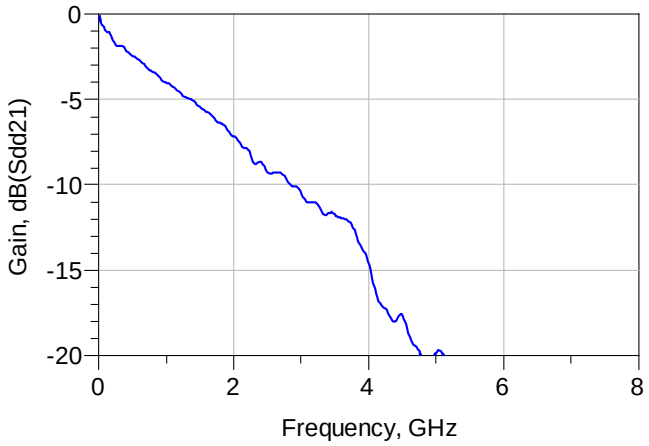

Figure 2 Measured differential Insertion Loss (Gain to be precise), of a practical well-terminated channel employing 2 connectors, vias, and PCB traces.

1.4 Introductory Description of Linear and Non-linear Systems

In order to understand two main sub-types of repeaters called re-drivers, this section provides an initial brief definition of linear and non-linear systems (or circuits). More will be given on this topic in section 3.2.

A Linear System: Is one where the amplitude of the output response is directly proportional to the amplitude of the input, irrespective of the shape of the output. The output and input do not have to have the same shape for a system to be linear (since shape is controlled separately by the transfer function of the system, *H(s)*). See section 3.2.2 for more details.

A Non-linear System: Is one where the amplitude of the output is not directly proportional to the input. The relationship between input and output amplitudes could be anything other than a straight line. In extreme cases, it could be a step function, i.e. as the input amplitude rises, there is no output initially, but then suddenly an output appears, and remains at about the same amplitude even if the input amplitude keeps rising. Such a system is a "Sharply-limiting" non-linear system. In more moderate cases the output amplitude reaches saturation gradually. See section 3.2.1 for more details.

2. Repeater Types

Fundamentally, high-speed serial **repeaters are of two types: Re-drivers, and Re-timers.**

2.1 Re-driver

This is usually a high-gain-bandwidth amplifier, employing input Continuous Time Linear Equalization (CTLE, [3]) and sometimes also output Transmitter Linear Equalization (TxEQ [3] [4]). The amplifier could be either linear or non-linear (limiting). These devices have no clock, and are pure analog devices, except for the presence sometimes of a sideband low-frequency bus to program their analog settings. Limited

programming is usually also achievable by using strap pins. Re-drivers do not store data digitally, nor could they be protocol-aware. They just compensate for ISI, cause a delay of the signal by a few 100s of pico-seconds, and usually add some jitter. Re-drivers are not specified directly in most Hi-speed serial bus standard specifications.

If not addressed explicitly in a bus's specification, then in general, using re-drivers renders a link non-compliant, strictly speaking. Their use in such busses as USB3 and SATA3 is "Extra-spec" (which might be surprising to some, but can be gleaned from careful reading of those specifications), but appears to be tolerated practically. More details on these devices' usage and subtleties will be given in later sections.

2.2 Re-timer

A device which has a Clock & Data Recovery circuit (CDR [5] [6]), which is the main component of a SerDes Physical Layer (PHY). A re-timer has a Phase Locked Loop (PLL [7]), may require an input Reference clock, and is a mixed-signal analog/digital device. It converts an incoming analog bit stream into purely digital bits that are stored (or staged) internally. The internal digital data has no analog information left in it from the incoming original bit stream. A re-timer re-transmits data anew, with new equalization, and new jitter content, unrelated analog-wise to the input bit stream. Thus, a re-timer breaks a link into two distinct sub-links, which are completely independent from each other, from a Signal Integrity (analog amplitude and timing) perspective. Some Re-timers also provide debug capabilities, such as eye margining, link status, and link health indicators. In addition, sophisticated re-timers, which are protocol-aware, comprise digital logic to manage link initialization, training, data encoding and decoding, and clock domain frequency differences.

Re-timers are more complex than re-drivers, larger in die and package size, are more expensive, and, as of this writing, are offered by fewer vendors than re-drivers. Re-timers are, in turn, of two types: **Simple** "**Bit Re-timers"** and "**Intelligent Re-timers". Bit re-timers** are usually **Protocol unaware,** & usually **TxEQ Training-incapable**, while **intelligent Re-timers** are **Protocol-aware**, and **TxEQ Training-capable.** Section 13 provides more details on re-timers.

3. Re-drivers

3.1 Re-driver Micro-architecture

This section discusses the generic internal micro-architecture of a re-driver. Any vendor's re-driver might have a different specific design, but they all share these general features. Figure 3 shows the internal components of a differential re-driver (See section 3.2). Re-drivers usually come as an identical pair (or set of pairs) in one package as shown in Figure 4, in order to accommodate the sending and receiving signal directions of one or more lanes. Typically, each direction of transmission has its own independent set of controls. The channel preceding the re-driver is referred to here as the "**Pre-channel"**, and the one following is referred to as the **"Post-channel"**.

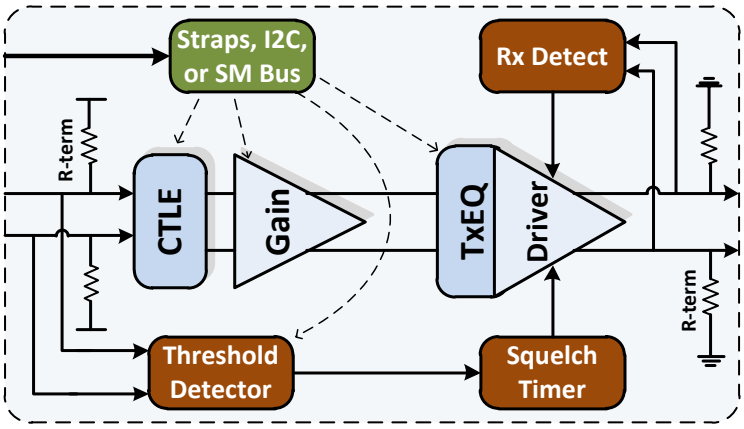

Figure 3 Conceptual micro-architecture of a differential SerDes re-driver, showing one signal direction, comprising a differential data path, with programming and control blocks.

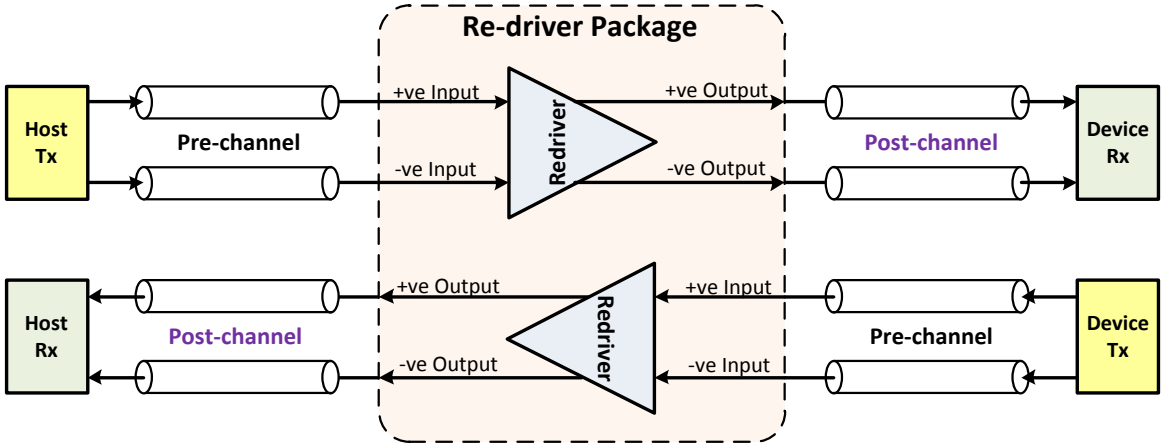

Figure 4 A typical re-driver package contains a full differential re-driver (as shown in Figure 3) in each signal direction, in order to allow repeating a full link. Also shown are the pre-channel (before) and the post-channel (after) the re-driver, in each direction.

A re-driver's main components are:-

3.1.1 Input Terminations

Depending on the circuit technology of the re-driver, an input pair of differential termination resistors is provided for matching the channel preceding the re-driver (pre-channel). These may range in value from 60 to 40 Ohms per side (120-80 Ohms differential), and are usually terminated to ground, Vdd, or another bias voltage. Re-drivers are usually DC-terminated devices at both their input and output, since they expect to operate in AC-coupled channels (using on-board capacitors).

The input impedance of a re-driver is not purely resistive, mainly due to the presence of package parasitics (inductance) and die input capacitance. A vendor's data sheet usually gives information (either

tabulated, or graphical) about the typical or worst-case SDD_{11} of the re-driver. SDD_{11} affects the shape and amplitude of the signals impinging upon a re-driver's input.

3.1.2 CTLE

A Continuous Time Linear Equalizer (CTLE) [8] is a linear amplifier whose transfer function is similar to that shown in Figure 5. It attempts to represent the *opposite* of the loss-dominated pre-channel attenuation, amplifying high-frequency components more than low-frequency ones, up to a certain frequency. Semiconductor circuits have a limited gain-bandwidth product, hence the progressive amplification of higher frequencies in a CTLE reaches a peak, then begins to decline. The peak is usually aimed to be at least as high as the Nyquist (clock) frequency corresponding to the bitrate, and preferably exceeding it as much as possible. The resulting band-pass filter opens the input eye by compensating for (reversing) ISI, only partially, since it is not an exact opposite of a channel's progressively higher attenuation shown in Figure 1 and Figure 2.

AC peaking, which is the difference in dB between the maximum amplification at the peaking frequency and the DC amplification, is usually externally programmable. In some re-drivers, the amount of DC gain is also externally, and independently, adjustable. It is important to note that a re-driver's CTLE setting is static, once programmed, irrespective of system operating conditions. This is true for both pin-strapped programming, and I2C (or SM) programming. A re-driver does not offer adaptive equalization, like some intelligent SerDes PHYs and some re-timers do.

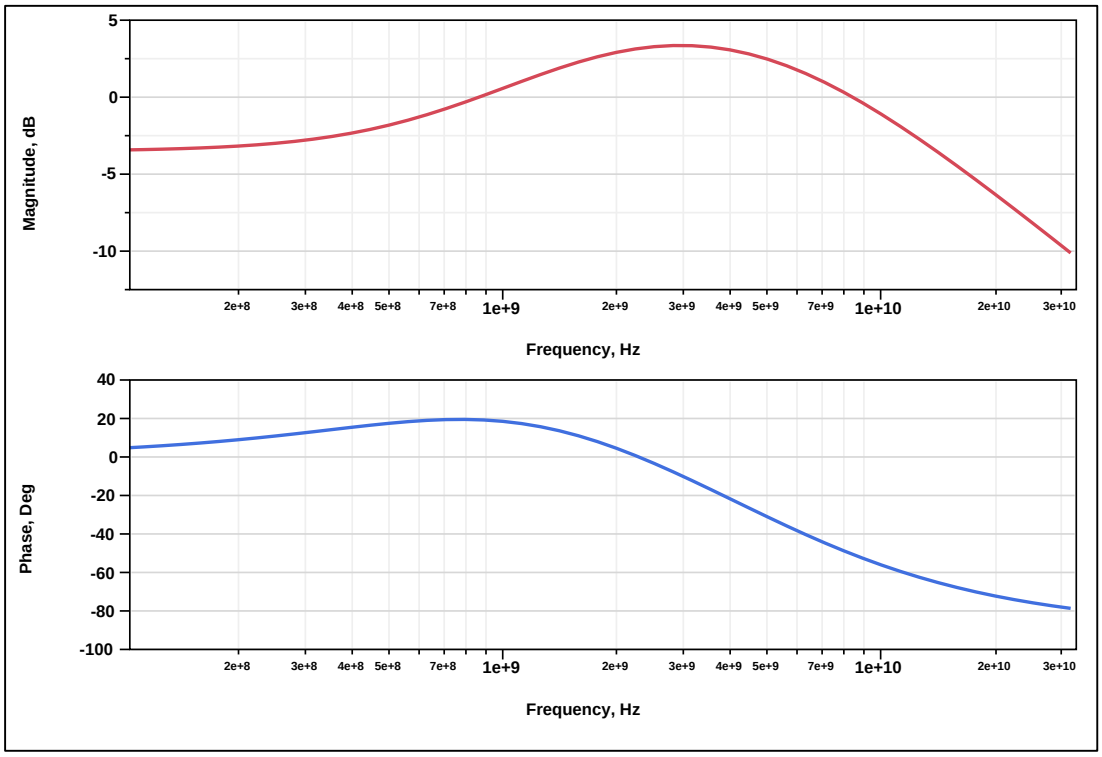

Figure 5 An illustrative example of the amplitude and phase Bode plots of a CTLE equalizer. A specific re-driver's CTLE is usually different from the above, depending on the design and bitrate. Notice the nonlinear phase response (versus frequency) of this 1-zero 2-pole equalizer.

3.1.3 Gain Stage

In many re-drivers, the amplification in the CTLE stage might be insufficient; therefore a second amplifier follows the CTLE. In addition, maximizing gain-bandwidth usually requires more than 1 or 2 stages. In the case of the limiting (or non-linear) re-drivers described in section 3.2.1, this is usually a high-gain clipping (i.e. limiting) amplifier. In the case of a linear re-driver (section 3.2.2), this stage is usually a more moderate-gain amplifier with "linear" input-output amplitude characteristics.

In the case of a limiting re-driver, there is usually a required minimum input differential voltage specification (on the order of 100-150 mV). This is the minimum voltage to guarantee that the gain stages, including the output driver, can swing completely to the full requested Output Differential Voltage (VOD).

3.1.4 Output Driver

This is the final power amplification stage which drives the output load. It can be either a limiting amplifier (section 3.2.1), or a linear amplifier (section 3.2.2). In some re-drivers, the gain of this stage is also programmable, in order to allow setting the Differential Voltage Output swing (VOD), say from 800 mV to 1200 mV, as an example.

3.1.5 Output TxEQ

This feature attempts to add de-emphasis (TxEQ) to the output. Since a re-driver lacks a clock (data is not staged digitally), an FIR-like behavior might be emulated either by delaying the output (using an analog delay circuit), scaling it, then adding it back to the main signal flow path, or alternatively through the use of an analog peaking circuit.

TxEQ capability is always present the case of a limiting (non-linear) re-driver. In the case of a linear re-driver (section 3.2.2), this function has not been observed to exist. The input CTLE is usually made strong enough to provide equalization, in order to cover what a TxEQ might add, since they both share roughly similar frequency domain peaking transfer functions.

A consequence of the methods used to achieve pseudo-FIR TxEQ behavior, is that a limiting re-driver -- which is designed for a certain bit rate-- may not be used at a significantly different bit rate. The reason is that the internal analog delays or peaking filters, which attempt to affect de-emphasis, are tuned to yield de-emphasis appearing after one UI (bit width) at the target frequency of operation. There is always a manufacturing variation in those circuits, so reasonable deviation is not detrimental.

3.1.6 Output Terminations

An output pair of differential termination resistors is provided in order to match the output impedance of the re-driver to the output post-channel. These may range in value from 40 to 60 Ohms per side (80-120 Ohms differential), and are usually terminated to the supply (Vdd) or an internal bias voltage.

Just like the input, the output impedance is not purely resistive, mainly due to the presence of package parasitics (inductance) and die output capacitance. A vendor's data sheet usually gives information (either numerical or graphical) about the typical or worst-case SDD_{22} of a re-driver.

3.1.7 Input Idle (Squelch) Threshold Detector

This circuit compares the input differential voltage to a fixed (sometime programmable) threshold on the order of about 100 mV (e.g. USB3), set for entering the Electrical Idle state. It sends a signal to a Squelch Timer if the input differential voltage is less than the idle threshold. This feature is used to detect bus inactivity, and start a process to turn off the output driver in order to save power, and reduce idle bus noise. Entering squelch is not immediate, and only happens after several nanoseconds of delay, depending on the bus protocol, and the expected bit patterns (See 3.1.8).

If the input rises above a value larger than the Idle Exit Threshold (300 mV in USB3, and called "Max LFPS Threshold", [9]), then the output of the threshold detector is reversed, and the re-driver's output is turned on. This capability is used by USB3's host's Low Frequency Periodic Signaling (LFPS), and SATA3's host's Out Of Band (OOB) signaling, in order to awaken a client device, before the resumption of data transmission. The Idle exit threshold is usually larger than the idle entry threshold.

On the other hand, a PCIe downstream device enters idle only after detection of an Electrical Idle Ordered Set (EIOS, [10]), and not due to signaling inactivity. However, it exits idle after detection of input signal activity using the Electrical Idle Exit Ordered Set (EIEOS) which has a low frequency pattern. Since re-drivers targeted for PCIe 3.0 do not have digital pattern detection ability, they are sometimes designed to turn their output off (go to sleep) after detecting that the input voltage has dropped below a certain threshold, for several nanoseconds. They do, however, exit idle upon bus activity (higher input voltage) when EIEOS reaches their input.

The author has observed, in the lab, that for some limiting re-drivers, the input voltage has to exceed the squelch exit threshold (reported in the datasheet) by tens of millivolts, before the output reaches its full desired amplitude. In addition, linear re-drivers tend not to squelch their output.

3.1.8 Squelch Timer

This circuit receives a signal form the input idle (or squelch) threshold detector if the input voltage is less than the Idle Entry Threshold, and after some time delay (typically 4-10 ns) it turns the output driver off, in order to save power, and reduce idle bus noise. Conversely, if the input rises above the Exit Threshold for a certain amount of time, then the output driver is turned back on, after a few nanoseconds. As mentioned in 3.1.7, the squelch exit threshold is typically larger than the squelch entry threshold.

Momentary crossings of the Idle Entry Threshold, by the input voltage, do not trigger shutting off of the output driver, due to the delay circuit. Such momentary drops can occur for very short periods of time in an encoded high-speed serial bit stream (e.g. "…10101...").

Also note that for the "Limiting, or "Non-linear" re-driver type describe in section 3.2.1, there is yet a third voltage threshold worthy of understanding: It is a switching threshold which must be exceeded, in order for the limiting amplifier to generate an output whose amplitude is at the full desired output swing. A well-designed re-driver would ensure that the idle exit threshold is above this switching threshold, in order to ensure that the output has reaches its full desired amplitude whenever it is turned on.

3.1.9 Receiver Detection (Rx Detect)

The PCIe and USB3 Specs provide a procedure for a transmitter to detect if a receiver is connected to the channel, thus giving it an opportunity to move to lower power states, when not. The procedure is based on sensing the different common mode charging time constants of the channel when and a load is either present or absent (Figure 6). The USB3 and PCIe procedures are similar. This is the procedure for USB3 [11]:-

1. A Transmitter must start at a stable voltage prior to the "detect common mode" shift.

2. A Transmitter changes the common mode voltage on Tx-P and Tx-N, consistent with detection of receiver high impedance which is bounded by parameter "ZRX -HIGH-IMP-DC-POS" in Table 6-13 of [11]. Furthermore, upon startup, detection is repeated periodically, until successful.

After shifting the common mode voltage, the output pins of the Tx (the Tx-P and Tx-P nodes between "R_Detect" and "C_AC" in Figure 6) start an upward voltage ramp. If there is no Rx load at the far end of the link (left circuit in Figure 6), then the Tx pins charge up rapidly, and a certain detection threshold is crossed quickly, since the isolation capacitor "C_AC" (which is the largest cap in the link) is floating. But if an "R_Term" is present (the right hand circuit in Figure 6), then the detection threshold is crossed much more slowly, as "C_AC", which is now in circuit, takes a longer time to be charged. Detection is successful if a load impedance equivalent to a DC impedance "RRX-DC" (Table 6-13 in [11]) is present. The R-term of the Rx does not have to be referenced to GND, for the technique to work correctly, since detection is an AC mechanism. This is easily proven, as the pin voltage at the bottom of "R_Detect" has to equal the applied voltage ("V_Detect") once the transient current subsides.

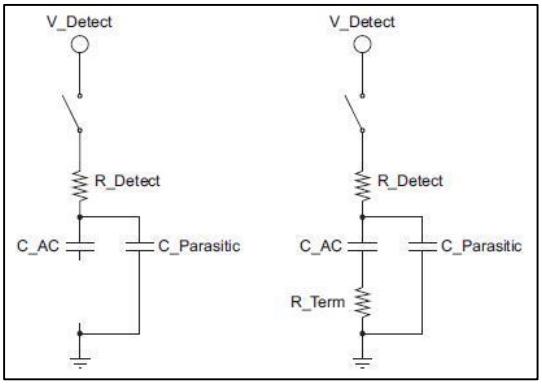

Figure 6 Circuit diagram indicating the concept of fast (left), and slower (right) charging time constant used by USB3 and PCIe for Rx detection, [11]. "R_Detect" is the common mode Tx source resistance.

Furthermore, in both PCIe and USB3, upon power-up, and as long as no load is detected at the output of the transmitter, an agent must keep its own input common mode impedance at a high value (USB3: $Z_{RX\text{-}HIGH\text{-}IMP\text{-}DC\text{-}POS} \geq$ 25 K-Ohms [11], and for PCIe: $Z_{RX\text{-}HIGH\text{-}IMP\text{-}DC\text{-}POS}$ and $Z_{RX\text{-}HIGH\text{-}IMP\text{-}DC\text{-}NEG}$, in the range of $\geq$ 10-20 K-Ohms [10]). This lets any upstream agents know, in turn, that a final destination Rx is still absent, since their Rx detection would continue to fail. Conversely, when an agent detects an Rx at its output, then it must turn its own input differential terminations and common mode input impedance ON to their nominal (low) values (USB3: 36-60 Ohms per differential side and a common mode of 18-30

Ohms [11], and for PCIe 1.0/2.0 40-60 Ohms per differential side, and for PCIe 3.0 it is bounded by R_{LRX-CM} of 6 dB, [10]).

Many PCIe and USB re-drivers possess the ability to detect an Rx, and behave similarly to a link agent, in order to allow power savings in a system. When no load is connected to a re-driver, it turns its input terminations to high-impedance, in order to let the upstream agent's transmitter (driving its input) know also that no downstream receiver is present. The re-driver's output is usually also turned off (set to a high impedance of a few 10s of kilo-ohms) in order to save power. The differential outputs reach an approximately equal voltage, with a certain common mode value.

All USB3 re-drivers seem to perform Rx detection. After power-up (or after a certain input control pin is toggled), an Rx-Detect cycle is performed periodically (typically every 12 ms). In PCIe, most re-drivers are capable of Rx detection, but not all offerings on the market now do. When a PCIe re-driver is incapable of Rx detection, then its input termination is turned on continuously upon power up, and the upstream agent's Rx detection would always succeed, even though there might not be a real load at the end of the link. In that case, the upstream agent goes immediately into training mode. But, since it does not detect a training response back form a downstream agent, it goes next into Compliance mode, where it sends a compliance pattern repeatedly. This is not a fatal error. It just implies extra power consumption in both the re-driver and the upstream agent.

There is no Rx-detection specification in both the 10G-KR and SATA specifications. In SATA, some agents turn their input terminations ON always to the 50-Ohm nominal value.

3.1.10 Sideband Programming Bus (Typically, I2C, SM Bus, or Strap Pins)

Depending on the market segment and bus specification intended for a re-driver, different means are provided to enable the designer to select the CTLE, TxEQ, Output Voltage Differential (VOD) swing, Squelch Threshold, etc. These means range from strap pins (set to ground, Vdd through a resistor, or left open circuited), to full I2C [12] or SM Bus [13] inputs. USB3 and SATA3 re-drivers tend to use strap pins, due to their relative simplicity, while re-drivers intended for PCIe and KR busses typically use I2C or SM bus programming inputs, in order to provide fine equalization resolution, and more flexible parameter and feature selection. Each direction of transmission, such as host to device (downstream), or device to host (upstream) is usually controllable independently, so the designer could select the required CTLE, TxEQ, Gain, etc. values appropriate for that direction's channel, Tx, and Rx.

3.2 Re-driver Types: Limiting and Linear

There are two main types of re-drivers in terms of input-output *amplitude* transfer functions*: **Linear, and Non-linear***. This pertains to the amplifying stage (or stages) in the re-driver.

3.2.1 Limiting (Non-linear) Re-drivers

Some re-drivers in the PCIe and KR domains, and all of the ones in the USB3 and SATA3 domains, at the time of this writing, are of the limiting amplifier type. Referring to Figure 7, this means that as the input differential amplitude rises from zero, the output amplitude (irrespective of waveform shape), starts rising very quickly, reaches saturation, and its wave shape is stable at full swing, unaffected by any further input amplitude increases. This happens at a small differential input voltage of around ~100 mV,

usually. Limiting re-drivers are easier to design. Furthermore, they can employ both an input CTLE, and an output TxEQ, as depicted in Figure 3 which showed the typical micro-architecture of a limiting re-driver.

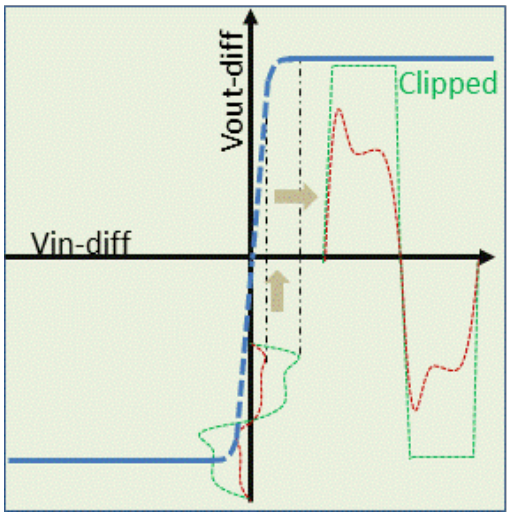

Figure 7 Relationship between input and output amplitudes of a "Limiting" re-driver, showing a quickly-growing output (red) as the low switching threshold is approached, then a regerenarted "square-like" output wave (green) once the threshold is surpassed. Jitter, added TxEQ, and bandwidth-limitted smoothing are not shown.

Figure 8 shows time domain waveforms at the output of a limiting re-driver as its input signal amplitude is raised in regular steps, where the output follows the input dis-proportionately, then quickly ceases to increase, due to the onset of limiting. Notice that the output saturates at small input amplitude. In addition, the pre-existing pre- and post-shoot equalizations of the input are obliterated. The pre-shoot in the input signal disappears from the output of the re-driver which applies only its own equalization to the output (only post-shoot, or de-emphasis in this example). Hence, a limiting re-driver *severs the analog wave shape relationship* between its input and output, but does not sever the switching-edge-crossing timing relationship, or jitter.

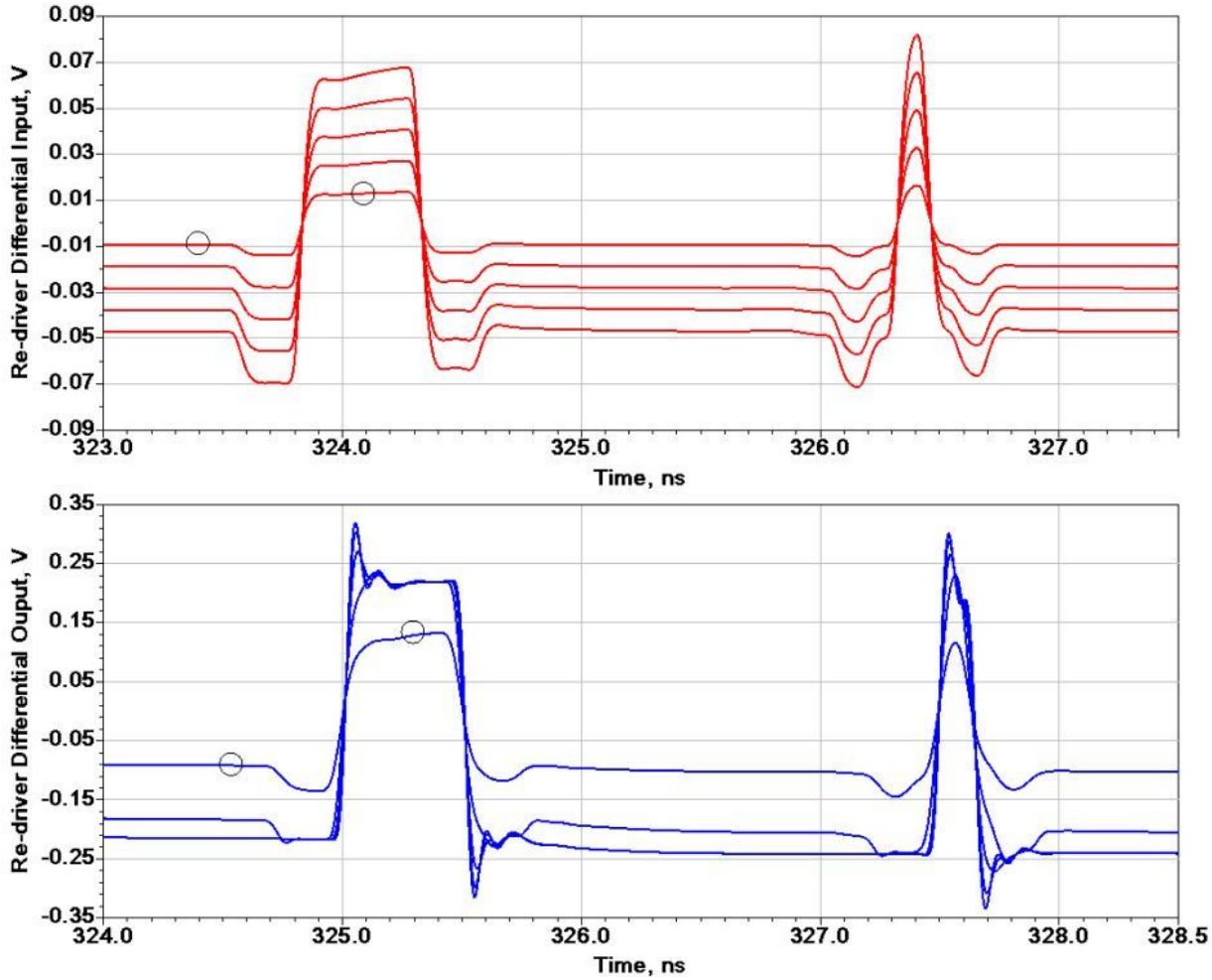

Figure 8 Input (top) and output (bottom) of a limiting re-driver as the input amplitude is raised at regular small intervals, and the re-driver output goes into saturation, ceasing to grow. Note the obliteration of pre-existing equalization in the input; the re-driver adding only its own.

For the sake of further illustration, Figure 9 depicts the simulated bit stream and eye diagrams at the input and output of a limiting re-driver placed in a lossy channel. The analog delay between the input and output is indicated by a grey dot.

With limiting re-drivers, un-equalized pre-channel ISI (under-equalization) or added ISI (over-equalization) by the re-driver's CTLE appear at the output as deterministic timing jitter (Dj), with a complete loss of signal shape information. This renders under- and over-equalization uncorrectable by a final downstream receiver. Figure 10 shows the signal at the output of a limiting re-driver's CTLE followed by its first limiting amplifier stage, where it is easy to see that un-equalized ISI results in pure Deterministic jitter (Dj) of the high-frequency type, which the Rx downstream cannot remove. This is due to the clipping (or re-shaping) nature of the amplifier which morphs all signals at its input into ideal "square-like" waves at its output, thus leaving only time axis crossing information, in the form of jitter. **This is one reason why limiting re-drivers usually extend channel reach less than linear ones, when all other aspects are equal. A limiting re-driver might extend the channel by ~40% (in total dB), while a linear one might be able to achieve ~60%.**

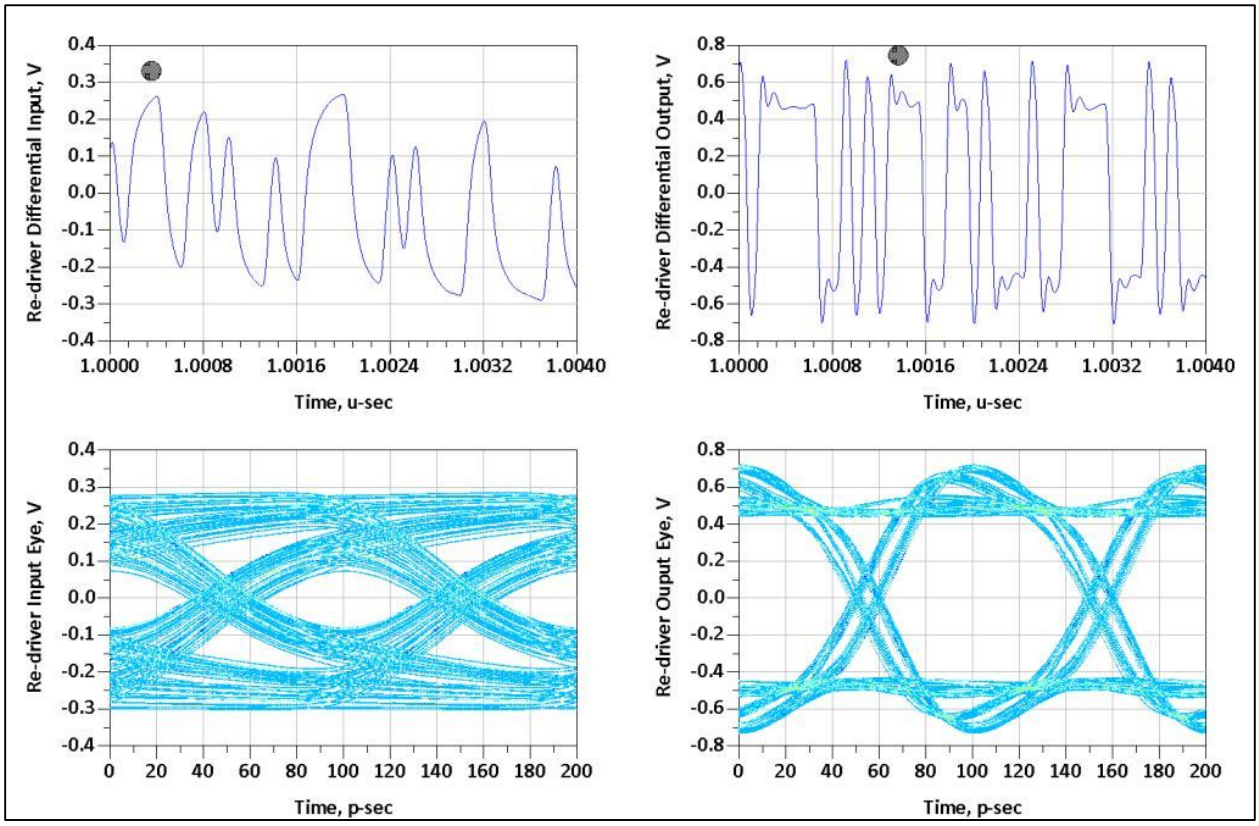

Figure 9 Simulated waveforms at the input (top left) and output (top right) of a Limiting re-driver, depicting the "regeneration" ("squaring") of wave shape, and the associated eye diagrams (bottom). TxEQ is also present at the output.

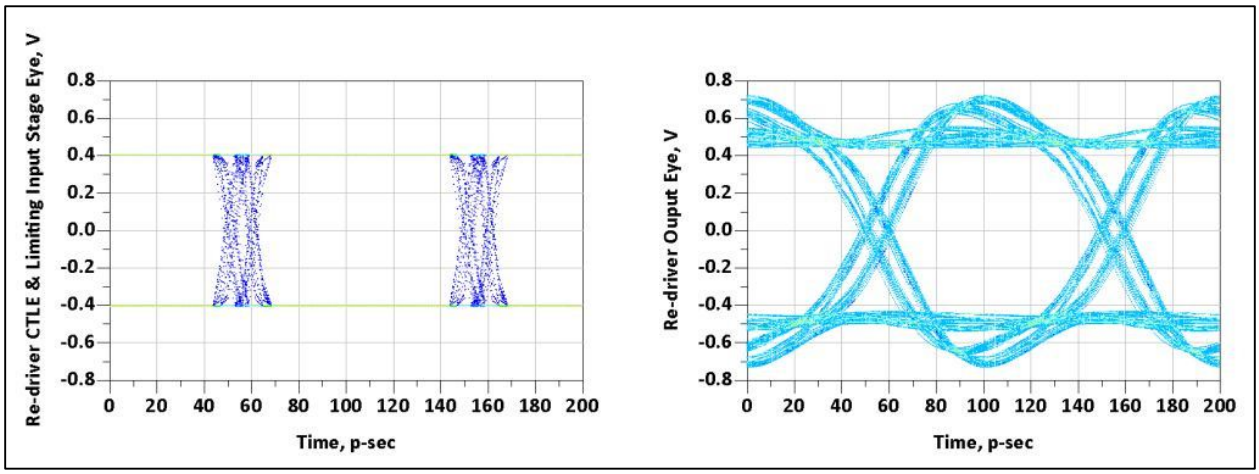

Figure 10 Simulated output of a limiting re-driver's CTLE which is followed by a limiting amplification stage (left), and the output from the final driver (right). TxEQ is present at the output. Notice the pure time axis crossing jitter which is not compensable by the final Rx at the end of a link.

3.2.2 Linear Re-drivers

Some re-drivers (especially those targeting PCIe 3.0 and KR) are built around linear amplifiers. **Linearity** means that --irrespective of incoming and outgoing signal shapes-- when the amplitude of the input signal is varied, and the amplitude of the output signal is measured, that there is a near-linear relationship between the two amplitudes, as shown in Figure 11.

Figure 11 Relationship between the input and output amplitudes of a "Linear" re-driver, showing linearity over a certain input range, and the eventual onset of saturatuon (or compression).

Relative linearity persists up to a certain value of differential input amplitude, and then compression sets in, until the output remains stable no matter how high is the input amplitude. The beginning of the end of linearity is usually indicated by a compression level, of say 1 dB, and may occur at as little as ~200 mV or up to ~600 mV of input differential amplitude, depending on the vendor, and gain settings. The compression level is defined as:

*Compression level in dB = 20 * Log10 (Actual Output Amplitude / Linearly-projected output Amplitude)*

Thus, a compression level of 1 dB means that the actual output is ~11% smaller in amplitude than the straight line projection of its value (at the same input amplitude point), as shown by the small double arrow in Figure 11.

It is worth noting that linearity does not mean that the input signal shape is identical to the output shape. It merely means that the relationship between the input amplitude and the output amplitude is a straight line, to a reasonable degree of approximation. The actual shape of the output is actually determined by the input signal shape and the equalization imposed on it by CTLE and TxEQ (if the latter is present). In other words, the output shape is a function of the input, and the overall frequency domain transfer function of the re-driver, *H(s)*.

Figure 12 shows the waveforms at the output of a linear re-driver as its input signal amplitude is increased gradually, where the output follows the input proportionately, but eventually ceases to do so,

due to the onset of compression (or end of linearity). The magnitudes of pre- and post-shoot relative to the full signal amplitude are affected as compression sets in.

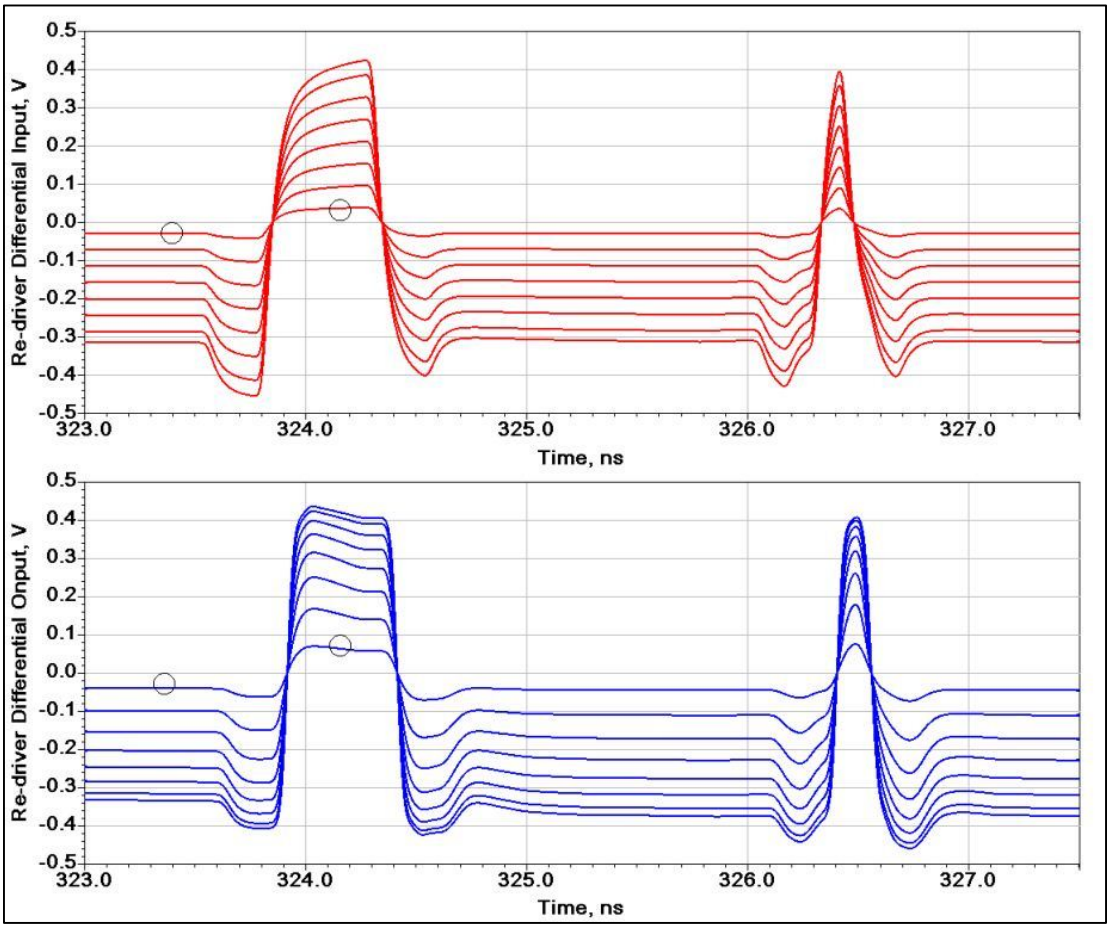

Figure 12 Input (top) and output (bottom) of a linear re-driver where the input amplitude is increased at regular intervals until the output goes into compression, i.e. ceases to grow at the same rate as the input. Compression diminishes the relative magnitude of equalization.

Linear re-drivers may or may not have TxEQ at the output (usually not). When they do not, they usually have a strong CTLE. In fact, TxEQ and CTLE perform similar (but not the same) transformations in the frequency domain, and the absence of TxEQ can be overcome by making CTLE stronger, in the linear case. Figure 13 depicts the micro-architecture of a linear re-driver which does not have output TxEQ.

For the sake of further illustration, Figure 14 shows an example of the expected wave shapes at the input and output of a linear re-driver. The output is not square-like, and has *some* resemblance to the input, and its shape is only altered by the amount of applied equalization. Both the analog amplitude and timing relationships between the input and output are maintained, albeit altered by the transfer function (*H(s)*). Unlike a limiting re-driver, the shape information connection between the pre-channel and the post-channel is not severed by a linear re-driver.

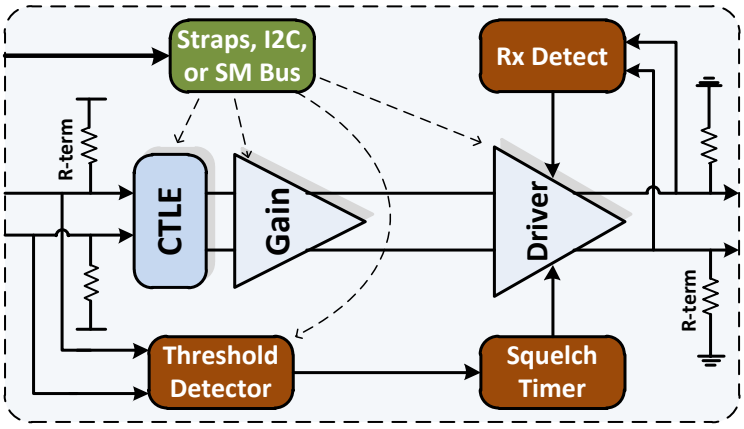

Figure 13 Conceptual micro-architecture of a linear SerDes re-driver without TxEQ, showing one signal direction, comprising a differential data path, with programming and control blocks. In this case, equalization is achieved via a strong CTLE only.

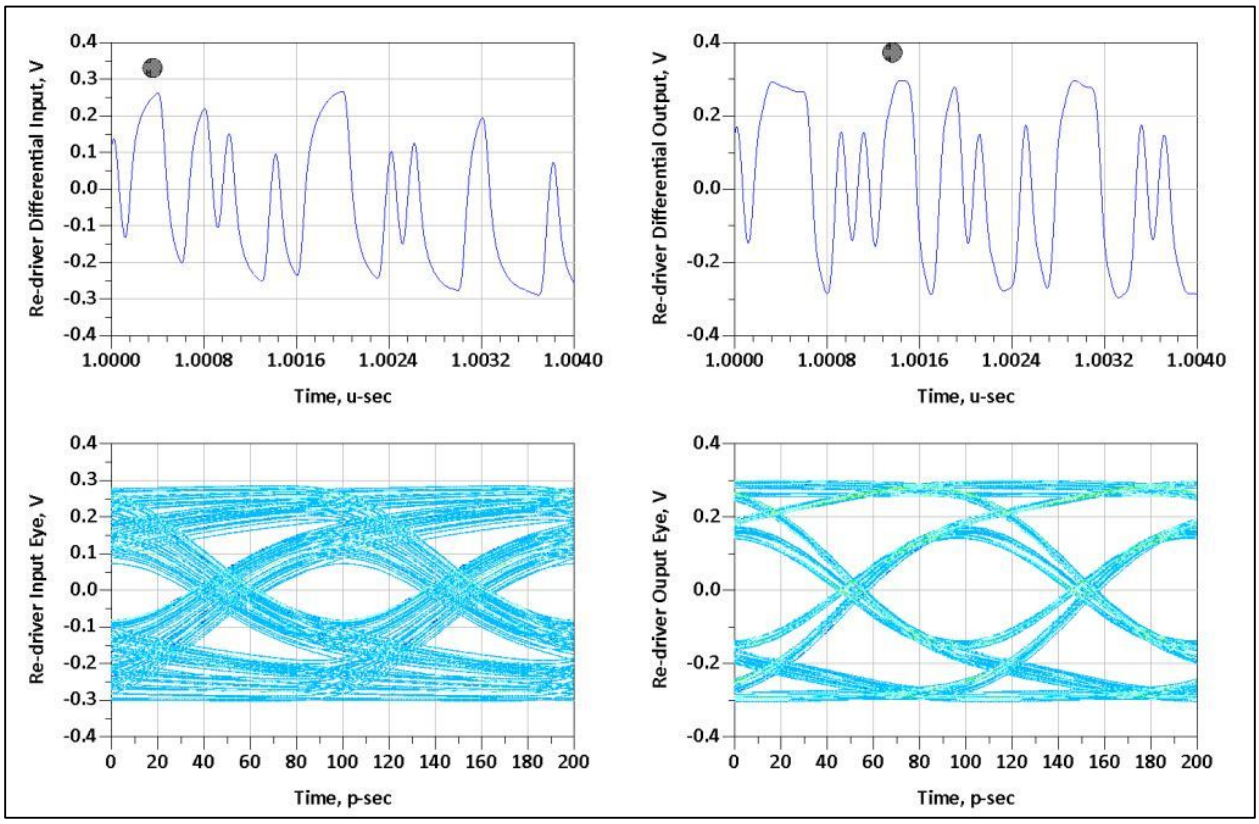

Figure 14 Simulated waveforms at the input (top left) and output (top right) of a "Linear" re-driver, depicting the equalized waveform, and the associated eye diagrams (bottom). Note the clear analog relationship maintained between input and output, and the lack of square-wave-like regeneration.

Figure 15 shows the transfer function (magnitude of the differential S-parameter SDD_{21}) of the combination of a PCB transmission line and a linear re-driver for various equalization settings of that particular re-driver. The settings range from under-compensation to overcompensation of the transmission line by the re-driver. Figure 16 shows the residual phase deviation from linearity for the same setup of Figure 15. Note how it is not possible to find a setting which makes the transmission line "disappear" fully. This illustrates why re-drivers change the character of a channel into which they are inserted.

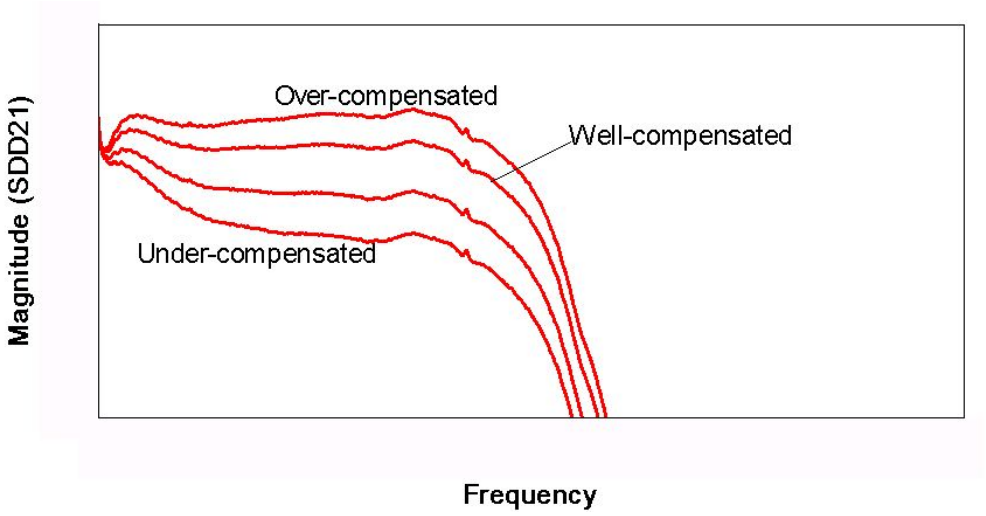

Figure 15 Measured magnitude of SDD_{21} of a re-driver plus a transmission line channel for various re-driver equalization settings, showing the overall transfer function of the combination.

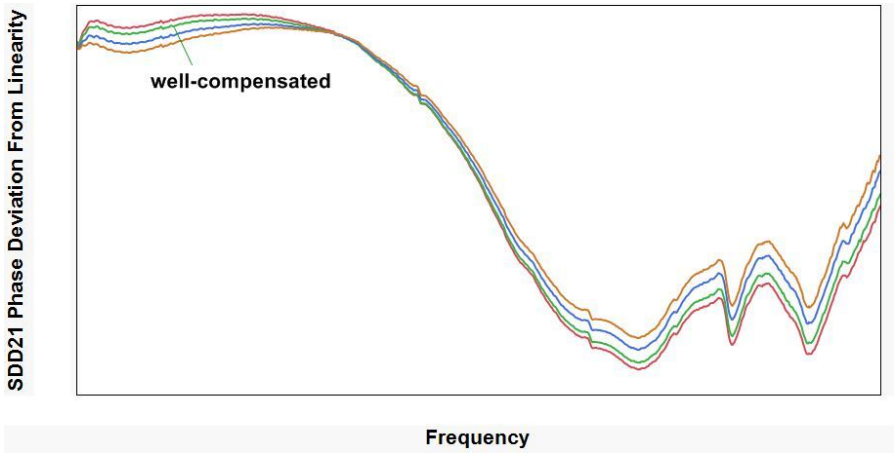

Figure 16 Measured phase deviation from linearity of SDD_{21} of a re-driver plus a transmission line channel for various re-driver equalization settings, showing the overall transfer.

An advantage of a linear re-driver is that its equalization applies toward the full channel (pre-channel plus post-channel), since the re-driver and the channel represent a linear system where --**to first order**-- the location of the transfer functions does not affect the final outcome --Other considerations however, make certain placements more preferable (See section 10.3). Furthermore, if a linear re-driver's CTLE capability is insufficient to compensate well for the pre-channel and/or post-channel, then the final receiver's CTLE and DFE could supplement the overall channel equalization. This is in contrast to a limiting re-driver (section 3.2.1), where under- or over-equalization appears at the output as timing jitter only, with a complete loss of signal *shape* information, thus rendering it un-equalizable downstream.

While a linear re-driver's equalization can apply toward compensating for the pre and/or post-channel, there is a change in the overall channel's transfer, even when the re-driver is programmed most optimally. The residual transfer function expressed by Figure 15 and Figure 16 causes the overall channel transfer function to deviate in its shape from that of a purely passive channel. This causes distortion in the equalization sent by the source Tx, as it appears to the final Rx, and will be discussed in latter sections on PCIe 3.0 and 10G-KR interoperability. This is in contrast to the limiting re-driver case, where the residual (uncompensated) part of the channel expresses itself as just deterministic un-compensable jitter (section 3.2.1).

3.2.3 Other Attributes of Linear and Limiting Re-drivers

3.2.3.1 Jitter Handling

Random jitter (Rj) and non-ISI deterministic jitter (Dj) [14] entering a re-driver are essentially passed on from input to output, albeit with a change in magnitude (spectral content) due to the re-driver's equalization circuits' transfer functions and their limited bandwidth.

Both limiting and linear re-drivers add a certain amount of Rj and Dj due to their own circuits, and their power supply noise. Hence, it is important that the supply (Vdd) of a re-driver meet the manufacturer's noise recommendations. This is made less important, in some case, by virtue of the presence of a Linear Voltage Regulator (or Low-dropout Regulator, LDO) inside a re-driver, which improves its Power Supply Rejection Ratio (PSRR). Added Random jitter (Rj) is caused by thermal noise and popcorn (Burst, or 1/f) noise. Thermal noise is determined by the design, and can only be reduced by consuming more current in the re-driver's internal circuits (which raises device power) and is outside the control of the user. Popcorn noise is determine by the semiconductor process technology and transistor sizes, and is also outside the control of the user.

Since any re-driver's CTLE and TxEQ settings cannot equalize the pre- and/or post-channels perfectly at all frequencies, under or over-equalization of the channel will appear as either additional ISI (in the case of a linear re-driver) or additional Dj (in the case of a limiting re-driver). Neither CTLE nor TxEQ could achieve perfect cancellation of a channel's loss, mainly since passive channel loss starts as a square root function of frequency, and then quickly becomes virtually linear with frequency (section 1.3), whereas CTLE and TxEQ have a frequency response representing a peaking band-pass function, with curved amplitude vs. frequency shape. Figure 15 and Figure 16 showed the residual frequency response of a channel plus re-driver combination, even after the best CTLE settings have been chosen.

Semiconductor process, voltage, and temperature (PVT) variations cause a re-driver's equalization to vary part to part, and system to system. Such variations might range from approximately +/-0.5 to +/-1.5 dB. In addition, PCB material high-volume manufacturing (HVM) variations and temperature/humidity loss dependence, all render a re-driver's fixed settings suboptimal, eventually. All these variations express themselves as added ISI (linear re-driver) or un-compensable Dj (limiting re-driver). To give an example of PCB loss variations, a 15 dB channel extension could suffer from an increase in loss of 1.5 to 3 dB over a 55-C temperature range and humidity range (for a server design), depending on its material properties [15] [16] [17].

To illustrate the effect of temperature alone, Figure 17 shows the input eye to a limiting re-driver located near the middle of a long channel operating at 5 Gb/s, at room temperature. It also shows the re-driver's output eye, and the eye inside a receiver having only a CTLE. Notice a receiver input eye opening of 97 ps. Figure 18 on the other hand, depicts the same waveforms after raising the temperature of only the PCB by 55 degree centigrade, and increasing the total PCB loss by only 10%. Notice the increased jitter at the re-driver's output eye, and the reduced eye opening post-CTLE, all caused by the increase in channel loss, which remains uncompensated for by a statically-programmed re-driver. Note that some PCB materials may have a loss increase approaching 20% for the same temperature rise and humidity increase, and that would cause significant further deterioration. Also note that the re-driver's equalization is not assumed to change as a function of temperature in this example, which is optimistic.

Any system could start up cold, warm up to it maximum temperature rating (or vice versa), and must remain functional at the required BER. In addition moisture content may change as the system warms up. Many receivers perform their initial training upon system start-up, and some of their equalization settings may remain unchanged thereinafter. That is usually referred to as "Start or Train Cold, Run Hot" (TCRH), or "Train Hot, Run Cold" (THRC). Such receivers are usually deigned to be able to accommodate channel loss changes versus environmental conditions assuming a base maximum spec channel. When a channel extension plus a re-driver are added, further accommodation of their environmental variations requires spare link margin, and appropriate validation.

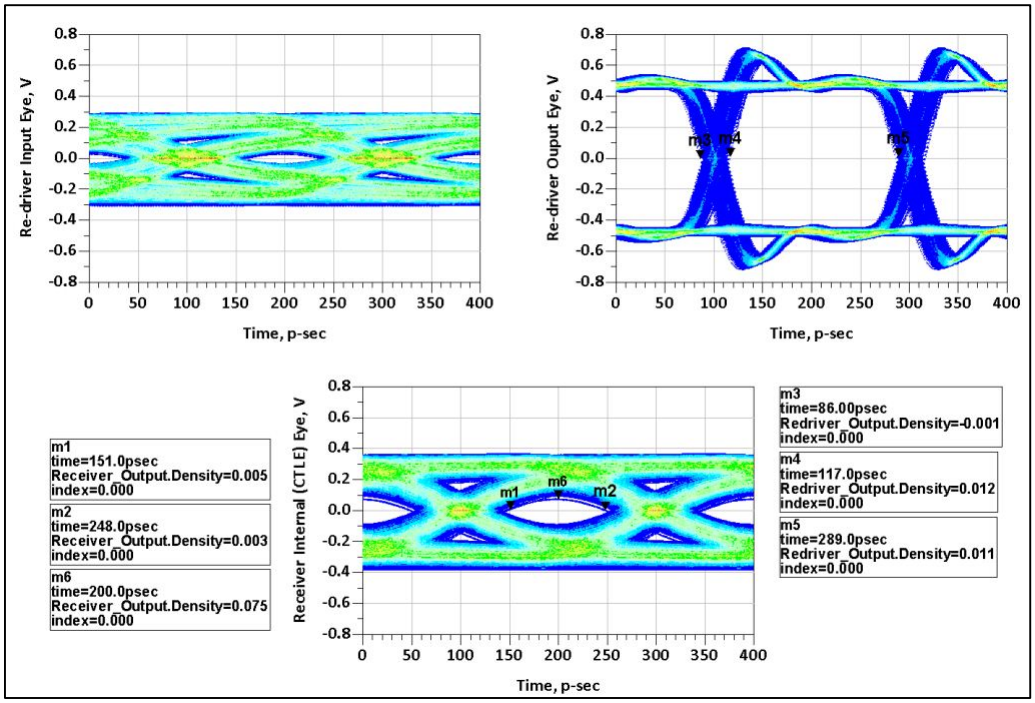

Figure 17 A simulated channel (~29 dB total @2.5 GHz) with a limiting re-driver located off the middle, showing re-driver output eye (Total jitter of 31 ps due to uncompensated ISI) & receiver internal post-CTLE eye (97 ps), at room temp. No non-ISI Dj & no Rj from the Tx, re-driver, or final Rx are included.

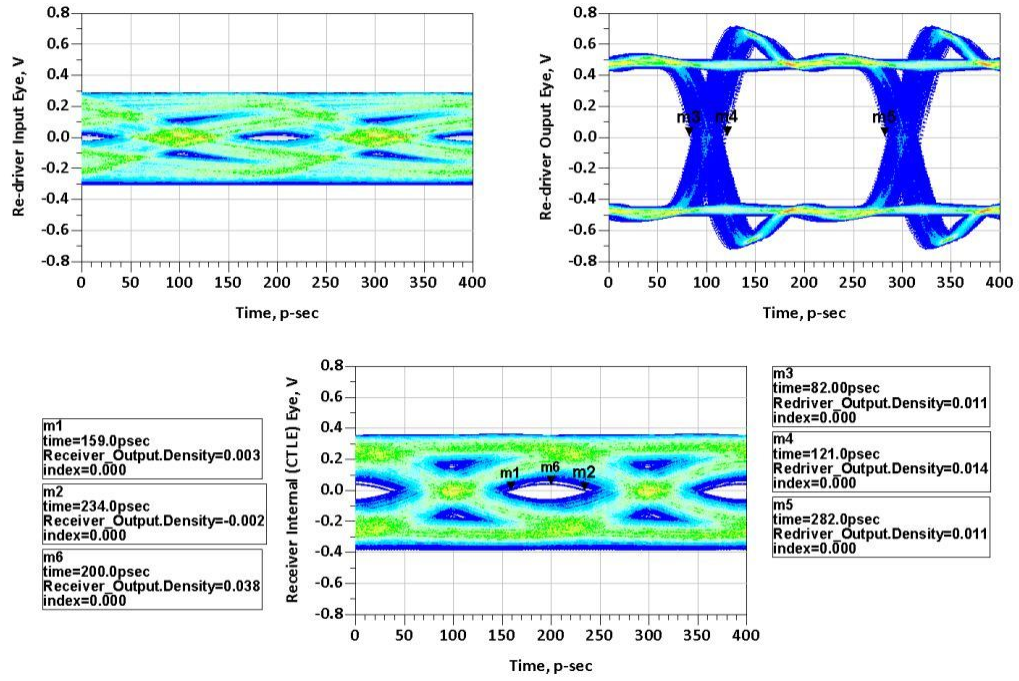

Figure 18 The channel of Figure 17 after raising only the PCB temperature by 55 C, & changing PCB loss by only +10% (Some PCBs may manifest up to +20% for a 55-C rise). Notice the increase in the re-driver's output uncompensable ISI-caused jitter (rising to 39 ps), & the reduced eye inside the final receiver (dropping to 75 ps). Temperature effects on the re-driver itself were not included.

3.2.3.2 Re-drivers and Non-ISI Distortions (e.g. Reflections)

Since a re-driver lacks a CDR or a clock, and has bandwidth-limited equalizing circuits, it can only compensate partially (strictly speaking) for the loss in a channel. In addition, Distortions caused by reflections (discontinuities) and cross talk (from neighboring aggressors), are handled differently by the two types of re-drivers: Linear and non-linear:-

In the case of a linear re-driver reflections and cross talk artifacts are *passed on to the output --albeit with some shape change depending on the frequency domain transfer function and linearity of the device***,** Figure 19**.** Furthermore, a linear re-driver amplifies noise and cross talk, just as it amplifies the main signal, and hence does not improve the Signal to Noise ratio (SNR) significantly. This is why placing a linear re-driver very close to the final Rx (short post-channel) runs the risk of shorter link extension, than placement near mid-channel, due to amplification of accumulated crosstalk.

In the case of a limiting re-driver, *reflections and crosstalk artifacts are largely eliminated from a voltage (amplitude) point of view (assuming that they still occur above the re-driver's switching threshold), but are both converted into deterministic jitter, especially in the case of either artifact occurring on the rising or falling edges of the incoming signal***,** Figure 20**.**

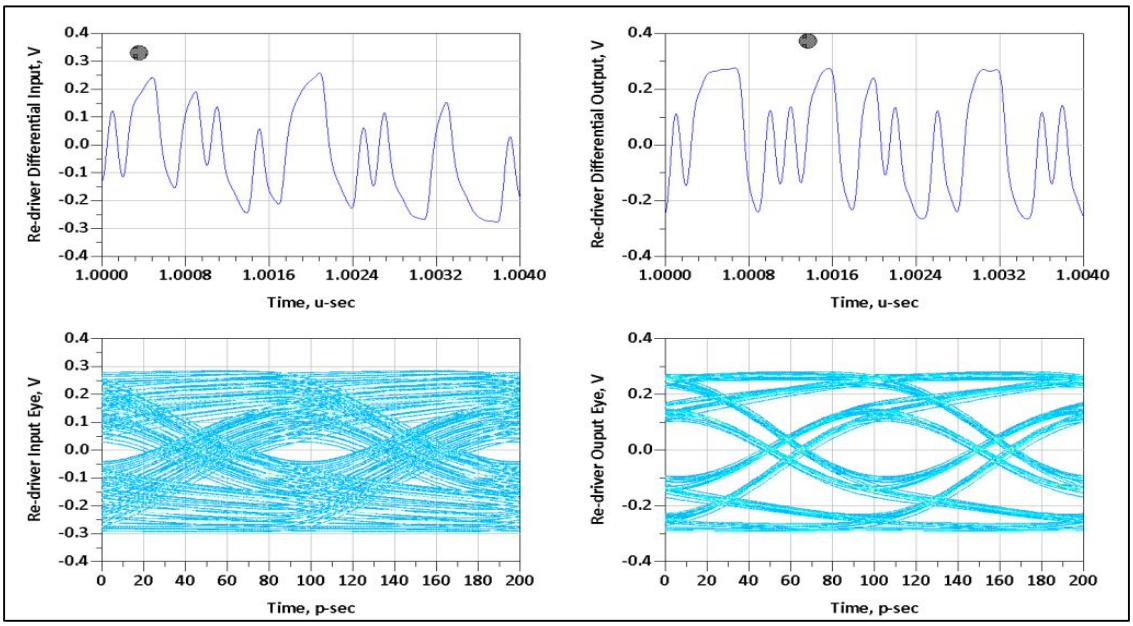

Figure 19 The simulated effect at the output of a "Linear" re-driver of a severely reflective channel discontinuity. Notice the changes in the incoming and outgoing waveforms and the eyes, relative to Figure 14.

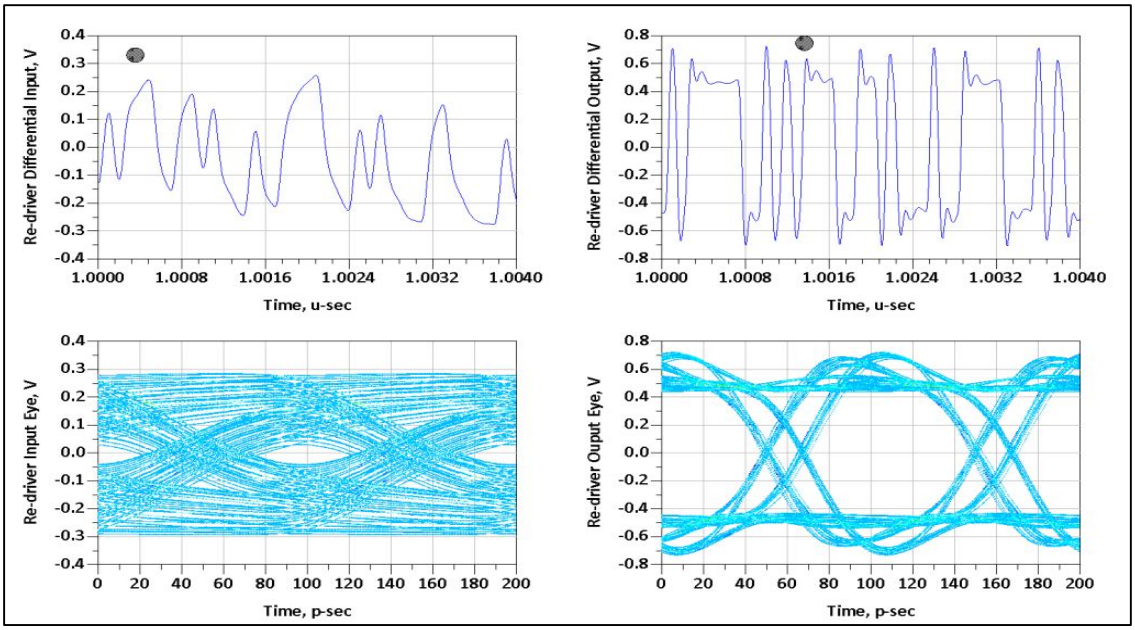

Figure 20 The simulated effect at the output of a "Limiting" re-driver of a severely reflective channel discontinuity. Notice the changes in the incoming waveforms and the eyes, and the increase in output jitter, while maintaining the max eye amplitude, all relative to Figure 19.

3.2.3.3 The Notion of a so-called "Spec-compliant" Re-driver

***It is meaningless to require (or state) that a re-driver be (or is) Spec-compliant if no such device has been explicitly defined in a bus's formal specification*!** The main reason is that Tx and Rx compliance specs are usually designed and stated assuming a passive channel, and do not account for an active element in the channel (with very few exceptions). An active device adds jitter, intra-pair skew, and alters the channel by a transfer function which is not of the same nature as a passive channel's.

Measuring Spec compliance at a re-driver **output**, and requiring it to be identical to that of a standard's Tx, is not meaningful, either. The reason is that a re-driver is not meant to drive a full maximum Spec channel at its output, like a standard Tx is meant and specified to do. A seemingly more relevant location at which to measure Spec-compliance (albeit still outside of the Spec, strictly speaking), is a clearly defined compliance point, such as in the USB3 (after a reference input CTLE), SATA (at a connector), or SFI (off a host compliance board). For these buses, a re-driver (be it limiting or linear) is still non-Spec-compliant, strictly speaking, due to its altering the ratio of compensable ISI to un-compensable jitter. See section 4 for more details on this subtle point.

Further complications arise with standards like PCIe 3.0 and 10G-KR, both of which require the transmitter to adjust Tx Equalization (TxEQ) to any setting requested by a receiver, within a specified capability range. With these types of standards, showing that the reference receiver produces an open eye with one TxEQ value, at a compliance point, is not sufficient to guarantee interoperability in the presence of a re-driver. The reason is that the reference Rx used in the Spec is not a required minimum capability (or design) for practical receivers, but is merely there to allow measuring "an eye" for the purpose of tying different ends of the spec together. This will be discussed in detail in sections 5.5 and 6.

3.2.3.4 Re-driver Usability with Lower Risk of Open Interoperability

While still strictly non-spec compliant, re-drivers are being marketed and used for such busses as USB3 and SATA3. Please refer to section 4 for the details on the reasons for non-spec compliance. Busses where re-drivers are being used, with a lower (but non-zero) risk of interoperability issues, are those which:-

- Have a static (non-adaptive) TxEQ, AND
- Have a post-EQ Rx eye Spec, such as USB3, or,
- An eye spec (mask) at a test point, such as SATA3.

In these cases, it is important to design the placement and programming of the re-driver with sufficient guard-band, in order to ensure that the eye opening meets the Spec's minimum eye mask, under all HVM and environmental (PVT) conditions, and also allow the Rx some room to deal with the altered ratio between increased non-compensable ISI (jitter), and compensable ISI (see section 4). Furthermore, in the Case of USB3, SATA3, and even SFI, the cables or devices connected to a host may vary in length from user to user (e.g. a 3m USB3 cable, versus a 10" cable, or a thumb drive). A host using a re-driver must be designed to accommodate these differing usage modes.

4. Re-drivers for USB3, SATA3, and PCIe 2.0

The USB3 [9], SATA3 [18], and PCIe 2.0 (Gen 2) [10] standards do not specify re-drivers. Hence, re-driver usage is outside those specifications. However, limiting re-drivers are being used in those open eco systems. Except for one subtlety, these buses are more tolerant of re-drivers, mainly due to the fact that they do not require TxEQ adaptation (unlike PSCIe3, and 10G-KR which do). USB3, SATA3, and PCIe 2.0 have a small set of required static TxEQ settings, which a statically-programmed limiting re-driver can provide, and no further TxEQ changes need to pass through the re-driver.

Technically, USB3, SATA, and PCIe 2.0 re-drivers are not Spec-compliant, due to the following subtlety: The aforementioned Specs define receiver compliance as the ability of a device to achieve the required BER, when plugged into a compliance channel presenting it with an eye having a specified minimum height and width. Due to the prescribed maximum Transmitter Dj and Rj, and the maximum length of the compliance channel, there is an implied assumption about the ratio of channel-caused ISI (compensable jitter) to the sum of Random and Deterministic jitters which are un-compensable. When a limiting re-driver is inserted into an extended channel, then the residual (un-compensable) ISI portion of the eye becomes larger (see section 3.2.1) relative to the Rj and Dj profiles. This may not suite the equalization capabilities of a receiver which does pass compliance under normal circumstances. Consider the case of an Rx which has very high internal jitter, but relies on superior equalization to pass Compliance. If such an Rx is presented with a Spec eye which has the required height and width, but which has more inherent un-compensable jitter and less compensable ISI (than during compliance), then it is plausible that such a receiver might fail to open that eye well-enough for its own low BER sampling, since the USB3 input CTLE is informative only (section 6.8.2. of [11]) and no equalizer is specified in PCIe 2.0.

It seems that despite the above cautionary consideration, hosts using a USB3, SATA3, or PCIe 2.0 re-driver, which are capable of producing the pre-scribed eye at a compliance point, are being assumed to

be compliant by some designers. This would be especially incorrect for PCIe 3.0 and 10G-KR, and the reader is referred to section 6 and 9 for a full explanation.

All re-drivers for USB3, SATA3, and PCIe 2.0 on the market are of the limiting type (section 3.2.1), at this time. The ability of a good re-driver to extend the channel is on the order of roughly 50%. If one leaves a guard-band of 10%, then it is on the order of roughly 40%. Thus, a 19.6-dB USB3 Spec channel could be extended by about 8 dB of loss, assuming that the re-diver is placed appropriately, as explained in section 10.3. There are some variations in the capabilities (and programming sophistication) of re-drivers from different vendors.

Most datasheets tend to state channel extensibility in terms of inches of FR4. But, length is not the relevant measure, since low-cost FR4 has a loss range from around 0.7 dB to 1 dB per inch at 4 GHz, depending on the specific materials being used, and the line geometry. A better way of stating channel extension would be in terms of total dB of loss at a specific frequency, of say 2.5 GHz for USB3 & PCIe 2.0, or 3 GHz for SATA3.

5. The PCIe 3.0 Equalization Adaptation Protocol

The PCIe 3.0 Base Specification [10] requires that a compliant Tx be able to produce any of the output equalization (TxEQ) levels shown in Figure 21, whenever requested to do so by the receiving port, during link equalization training. The same specification describes a protocol of TxEQ adaptation, where the host (upstream agent) or the end device (downstream agent) can each request that any of the TxEQ values in Figure 21 be presented to them by the other agent on the link. The intent of adaptation is to allow both agents to adjust the link partner's TxEQ to an optimal value for their receiver, and for the specific channel and operating conditions at hand.

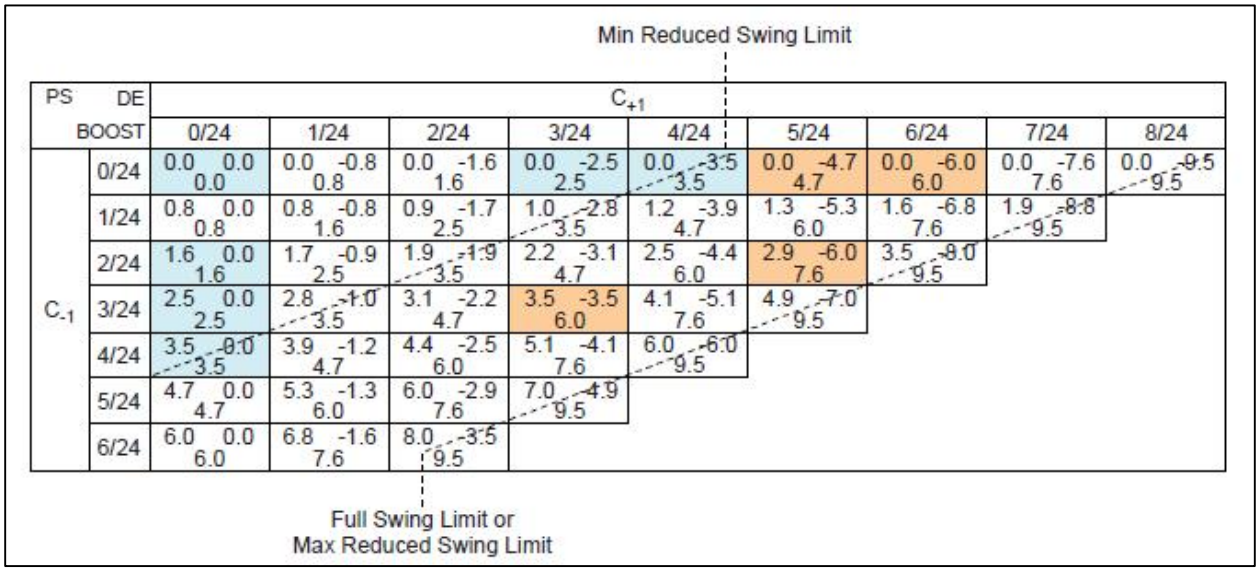

Figure 21 PCIe 3.0 required Tx Linear equalization settings according to the PCIe 3.0 base specification, showing pre- and post-cursor coefficients.

The table in Figure 21 shows the minimum granularity and range of TxEQ for PCIe 3.0. There are two equalization locations, “Pre-shoot” and “Post-shoot”. The Pre-shoot (Pre-cursor) level is indicated by the coefficient C_{-1} and can range from 0 to 0.25. Post-shoot (also called De-emphasis) is indicated by the coefficient C_{+1} which can also range from 0 to 0.25. The cursor is usually called C_0. The sum of the absolute values of all three coefficients is normalized to be equal to 1. Thus:

$$|C_{-1}| + |C_0| + |C_{+1}| = 1, \quad C_{+1} < 0,\ C_{-1} < 0$$

Only C_{-1} and C_{+1} are adjustable directly, whereas C_0 is always derived from the above equation. The output signal amplitude is composed by applying the following FIR filter function:

$$V\_out_n = (V\_in_{n+1} * C_{-1}) + (V\_in_n * C_0) + (V\_in_{n-1} * C_{+1})$$

The above FIR is depicted in Figure 22. Where V_in_n is the current bit (cursor), V_in_{n+1} is the next bit, and V_in_{n-1} is the previous bit (before the cursor). The differential voltage swing of the Tx is dependent on the TxEQ being applied, but is scalable independently of the three coefficients (C_{-1} , C_0 , C_{+1}), and can be in the range of 800-1200 mV while driving a 100-Ohm differential load, when there is zero TxEQ.

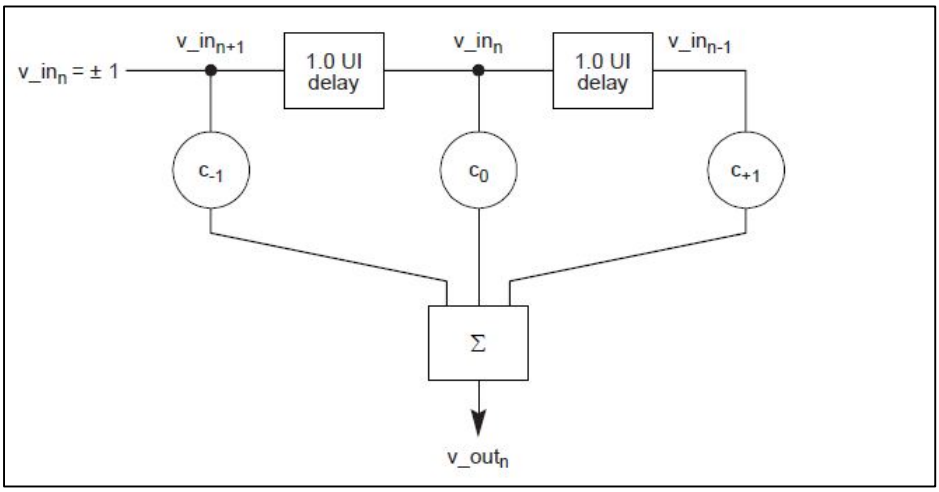

Figure 22 The PCIe 3.0 Tx Equalization FIR filter representation. (Note: The 10G-KR Filter is similar in definition).

This PCIe 3.0 TxEQ adaptation [10] protocol has 4 phases, defined as follows:-

5.1 Phase 0

PCI Express links always start at 2.5 GT/s. Before changing to the higher rate (8 Gb/s), an upstream agent (downstream port) and downstream agent exchange information on the range and granularity of TxEQ values supported by each transmitter at 8 GT/s.

The upstream agent instructs the downstream agent on which TxEQ setting to start transmissions with, once the link transitions to 8 GT/s. The upstream agent picks its initial TxEQ setting to use for its

upstream agent transmitter at 8 GT/s based on system knowledge of the channel lengths on the motherboard.

5.2 Phase 1

The link transitions to 8 GT/s, and both receivers must be able to start receiving with a BER of E-4, or better, with the initial TxEQ settings (decided in Phase 0) at 8 GT/s.

5.3 Phase 2

The downstream agent sends to the upstream agent one or more requests for any TxEQ settings chosen from the table in Figure 21, until it decides on an optimal setting. The implementation details of how an eye (or BER) is evaluated, is left to implementation. Different receivers have a variety of algorithms for both internal CDR convergence, and eye goodness or BER estimation. Since they are implementation-specific, they could range from exhaustive searches, to methods of steepest descent, which could be subject to local extrema traps. In addition, "no changes" in the perceived eye or BER (for example due to a limiting re-driver) might also trip a "Best eye" or "Best BER" search state machine, depending on its specific logical constructs.

5.4 Phase 3

Here, the roles of phase 2 are reversed, and the upstream agent (downstream port) makes TxEQ preset and change requests to the downstream agent. If the training is successful, then at the end of this phase the link is established at a BER of at least 1E-12, both in the downstream and upstream directions.

5.5 PCIe 3.0 Open Interoperability

Open Interoperability is the notion that any pair of specification-compliant devices (or agents) ought to be able to find TxEQ equalization values appropriate for each agent's Rx from the other agent's Tx, and be able to operate the link in both directions at BER of 1E-12, automatically.

The PCI-SIG facilitates standard compliance testing for devices implemented to the Electromechanical (CEM) specification [19]. A compliant Tx passing CEM at 8 GT/s requires that it be able to produce a minimum eye at a BER of 1E-12 when driving the appropriate CEM Reference channel. The eye is measured after the reference equalizer defined in the PCI Express 3.0 base specification – including one of 7 CTLE curves (whichever one works best) and a one tap DFE [10]. A compliant Tx is required to produce the aforementioned minimum eye for at least one TxEQ preset. Furthermore --but using a separate test-- a compliant Tx is also required to produce *all the other* equalizations values tabulated in Figure 21, and this is done by testing it for a selected subset of values (the "Presets"), and requiring that its design be able to produce all the other values in the table. Aside from equalization capabilities, a Tx is also required to meet certain deterministic and random jitter maximum specifications [10].

It is important to understand why the accuracy of the Tx equalization across the required TxEQ space is tested in addition to the eye test. The PCI Express 3.0 base specification defines a receiver "Stressed Eye" test that all compliant receiver silicon must pass. To test Rx for compliance, it is plugged into the CEM test channel, which is driven by a calibrated Tx (say from a JBERT). The Tx calibration process for this receiver test is illustrated in Figure 23. The Tx amplitude calibration is done with the Tx Equalization

fixed first to 0 dB of pre-shoot and 0 dB of de-emphasis. Then the Tx EQ is set to Preset 7 (P7) [10], and noise sources are added and adjusted until the eye --measured at the end of the reference channel (TP2) is 25 mV and 0.3 UI at a BER of E-12 --after applying a reference receiver with a certain equalizer having a CTLE and 1-tap DFE, defined in the specification [10].

After Tx and eye calibration, the Rx under test (such as one on a PCIe card, or a host) in attached to the CEM setup. The Rx is allowed to request adjusting the TxEQ to any *other* value (other than P7) in the space shown in Figure 21, until it achieves a BER of 1E-12. Silicon receivers are NOT required to work as well as the reference receiver at a given TxEQ setting – regardless of channel length. Therefore – it is important that a transmitter be able to reproduce the entire required TxEQ space *faithfully*. Silicon receivers may have a strong preference anywhere in this space, and have performance that degrades rapidly away from the ideal TxEQ region for that receiver. This means that performing a host Tx eye test with the reference receiver, and showing that it passes with a single TxEQ setting, is not a sufficient to guarantee interoperability. This would only be true if all silicon receivers were required to work as well as the reference receiver at all possible TxEQ settings, which is not the case in PCIe 3.0.

One might then ask: Why does the PCIe 3.0 Spec use a reference Receiver? Section **4.3.4.3.5**., of the base specification [10], describes the role of the reference Rx, and how it is *not* a required design implementation:-

"For the longest calibration channel the stressed eye will be closed, making direct measurement of the stressed eye jitter parameters unfeasible. This problem is overcome by employing a behavioral receiver equalizer that implements both a 1st order CTLE and a 1-tap DFE. For the short and medium calibration channels the behavioral Rx equalizer shall implement a 1st order CTLE only.

Rx behavioral CTLE and/or DFE are intended only as a means of obtaining an open eye in the presence of calibration channel ISI plus the other signal impairment terms. The behavioral Rx equalization algorithm is not intended to serve as a guideline for implementing actual receiver equalization." (Emphasis added).

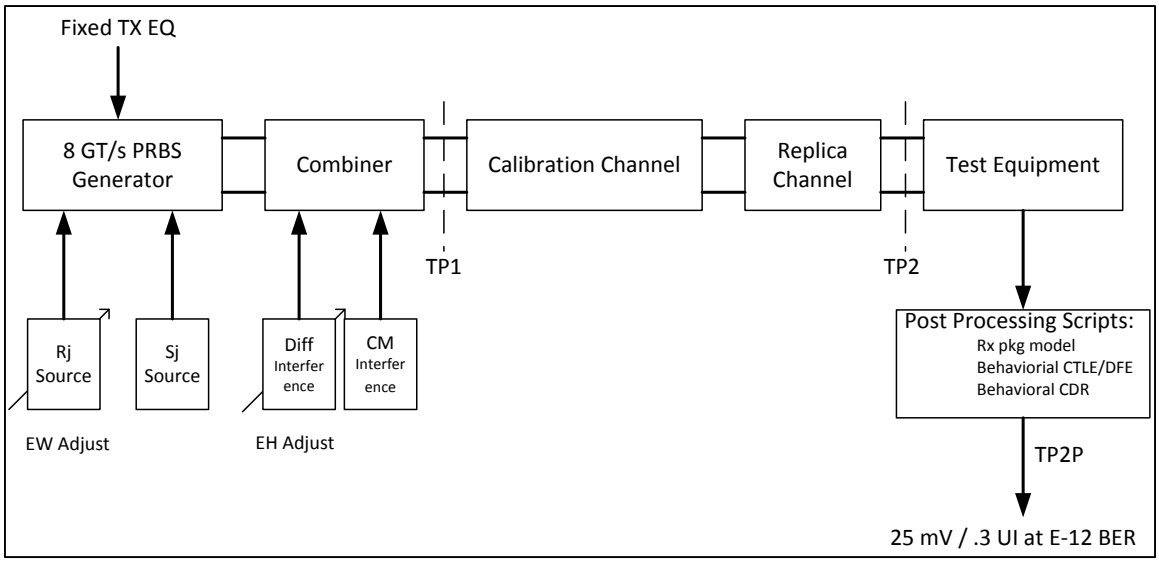

Figure 23 PCI Express 3.0 Rx Stressed Eye Calibration Setup.

The PCIe 3.0 Base Spec [10], & PCIe 3.0 CEM spec [19] guarantee that devices whose Tx & Rx pass compliance testing independently, will interoperate, by requiring (or allowing) all the following:-

1. A Tx must be able to produce any of the pre- and post-shoot values in Figure 21, upon request from an Rx, and also meet specified tolerance requirements (+/-1 dB for pre-shoot, and +/-1 to 1.5 dB for post-shoot). See table 4-16 of [10].

2. A Tx must pass a compliance eye mask test at a minimum of one (or more) of the TxEQ values of Figure 21.

3. A Tx compliance eye must be obtained using an informative spec Rx, which is provided for measurement purposes, and also for simulating a channel using a minimally-compliant spec Rx.

4. An Rx is allowed to request any of the TxEQ values of Figure 21, and receive them from the link Tx.

5. An Rx achieves a BER of 1E-12 for one (or more) of those TxEQ values --any of its choosing. A spec-compliant Rx is not required to match the performance of the reference Rx at any specific TxEQ setting – but must match, or exceed, the reference Rx for at least one TxEQ setting.

6. The TxEQ selection algorithm used by the Rx to achieve a BER of 1E-12 is implementation-specific.

7. The architecture of the Rx (its equalization and CDR) is implementation-specific.

8. The TxEQ at which a Tx passes compliance may differ from that required by different receivers to pass compliance.

9. The channel between Tx and Rx --the CEM channel described in [19] is passive. That channel has 2 connectors and 3 boards, and transports the TxEQ space of Figure 21 in a deterministic manner. Due to this attribute, the CEM channel is suitable for both Tx and Rx compliance testing. **The CEM Channel's passivity assumption underlies open interoperability.** An Rx design would rely on such a channel during design simulations, and later also during physical compliance testing.

The presence of re-drivers (even the linear type), as will be shown in section 7.2, distorts the TxEQ space of Figure 21 in a manner different than the passive CEM spec reference channel alone does. Thus, re-drivers present a compliant Rx with a link where it might never be able to find a TxEQ which appears to it in the same fashion it did during its compliance testing. The distortion is worse than that of added vias, or other passive components, as can be shown in simulations within a relevant range of frequencies.

Stated at a high level, an active component, which is not specified in the Base Specification, has no room in a compliant link, since the Spec does not budget for such a device's transfer function and tolerances. This determination was also made by the PCI SIG. Hence, the SIG recommends using compliant re-timers [20].

6. Re-driver issues in PCIe 3.0

The PCIe 3.0 Base Specification [10] is completely silent on extension devices (including re-drivers). A repeater ECN --which specifies re-timers-- has been approved, and is available through the PCI-SIG's website [20]. When an active device (other than a re-timer) is placed in a PCIe 3.0 link, the above

scheme of open-interoperability (section 5.5) based on TxEQ Figure 21, passive channels, and the Phase 2/3 adaptation protocol, is no longer valid, strictly. More detailed reasons are given in the next two subsections for each type of re-driver.

6.1 Limiting Re-drivers versus PCIe 3.0 TxEQ Requirements and Adaptation Protocol

Referring to section 3.2.1, a limiting re-driver re-shapes the incoming signal at its output, turning it into a pseudo-square wave (with finite rise and fall times), with a new statically-programmed TxEQ. Thus, the TxEQ changes requested by either host or end agent in phases 2 and 3 (sections 5.3 and 5.4), are blocked from passing through the re-driver. Hence, the port requesting the TxEQ change would not be able to see its effects, since the expected signal shape change is blocked by the limiting re-driver.

Furthermore, since the Base Spec [10] is silent on the exact implementation of the TxEQ adaptation algorithm in a receiver, then it is entirely possible that some algorithms might fail (time-out) completely. Consider a case where an agent requests a TxEQ change, and expects to see a different eye, hence a different BER, and checking that against a pass/fail criterion. If the check comes back with the same result every time, (since the limiting re-driver is blocking TxEQ changes from passing), then depending on the specific algorithm used, the Rx might time out.

Hypothetically, a system designer might be able to select a specific limiting re-driver, and program it statically in such a way that it could operate with one, or a small and well-characterized (tested) set of link partners while effectively disabling the TxEQ adaptation protocols on both ends of the link. This implies that neither agent in the link should even attempt to optimize the link any further. A PCIe 3.0 host can disable Phases 2 and 3, and the end agent would default to the values sent to it by the host in phase 0. Such a system is referred to here as a "**Closed System**" or a "**Closed-slot System**" which does not support open interoperability (Section 10.1). The market might not offer a wide range of limiting re-driver with a high enough frequency, and a wide enough TxEQ space for this to be possible. Furthermore, disabling TxEQ training would entail intervention into the firmware of a system. Hypothetically, such a usage model should ensure that firmware could disable coefficient update requests during training.

6.2 Linear Re-drivers vs. PCIe 3.0 TxEQ Requirements and Adaptation Protocol

Referring to section 3.2.2, a linear re-driver does allow incoming signal shape *changes* to appear at its output. TxEQ changes requested by either the host, or the end agent, in phases 2 and 3, could pass through the re-driver, and the requesting agent can detect a change in the eye opening, or the BER.

There is an issue which arises when a "Linear" active device is introduced into a channel. As explained in section 3.2.2, linearity only implies that the relationship between the incoming and outgoing signal *amplitudes* is that of direct proportionality (a straight line), irrespective of their shapes. But, the active device would alter the incoming signal shape according to its frequency domain transfer function *H(s)*, which is a function of its CTLE, TxEQ (if any), and amplifier frequency response. In principle, it is not possible to design a compact equalizer (using a finite number of poles and zeros), to compensate perfectly for the loss-vs-frequency function of a PCB channel (section 1.3), especially with current &

foreseeable gain-bandwidth process capabilities. There are extraneous *amplitude and phase effects* which render the compensation imperfect.

Furthermore, PVT variations of the re-driver and the extension PCB would cause deviations in the compensation, and add tolerances to apparent TxEQ which are unaccounted for by the pre- and post-shoot tolerances allowed in table 4-16 of the base PCIe 3.0 specification [10]. In addition, re-drivers have a finite number of programmable settings for their CTLE and TxEQ. The discrete nature of the settings means that, even if the re-driver has the perfect frequency response which cancels the channel extension, there would most likely be either over- or under-equalization, both of which alter the pulse response of the channel. Figure 15 and Figure 16 are an example of such deviation.

When a channel extension is imperfectly compensated for by a linear equalizer (re-driver), then there is no guarantee that the equalization space tabulated in Figure 21, would appear to the Rx in the same way it does, when the channel is only a passive channel, within the limits of the specification.

Receiver design is implementation-specific (not even having to meet any specific CTLE or DFE equalization requirements, as was stated in section 5.5), and some receivers seem to prefer certain regions of the equalization table in Figure 21, e.g. a specific preference for pre-emphasis and/or de-emphasis. If such a region appears distorted due to the placement of a re-driver, then the receiver might not find an operating point at a BER of 1E-12. There are other factors also which might affect link health, such as noise (deterministic or random jitter) which might be added (or amplified) by a re-driver.

The notion that if a linear re-driver is set to over-compensate an extension (hence compensate some of the base Spec channel also), then CEM could be met, is also flawed. Actual testing has shown this notion to be inaccurate, since over-compensation causes TxEQ distortion just as under-compensation would (See section 7.2). Please also note that since re-drivers are not constrained by any specification, it is not even possible to re-define a new "Re-driver-friendly" CEM eye --which might guarantee open interoperability-- if presumably met by an extended channel which is compensated (or over-compensated) by a re-driver!

Practically, TxEQ adaptation might succeed in finding an operating point for several host/end device combinations. However, there is no longer *an implied Spec guarantee* that this would hold true always, for any two devices that pass CEM compliance independently, in the open market. Despite this, some designers are using re-drivers in systems that are meant to be openly interoperable. **With the information provided in this document, a designer should be better-prepared to understand the implications, and make his or her own risk-taking judgment.**

In addition, a host designer might be able to select a specific linear re-driver, and program it in such a way that it could operate with a well-characterized (tested) set of end agents. Such a system is still referred to here as a "**Closed-slot System**", which does not support "unconditional" open interoperability (see 10.1).

7. Comparative Evaluation of Linear PCIe 3.0 Re-drivers

While PCIe 3.0 and 10G-KR re-drivers can never be openly interoperable, they are sought by many designers as a convenient method to extend those channels. Pressure is mounting to provide designers with a method to select the best re-driver offering. In support, the authors developed, and tested, such a method. The method provides an objective approach for comparing (and choosing) the best device for "Closed-slot" design, and also demonstrates the degree of departure from open interoperability, thus giving the designer a measure of such risk. The procedure is based on the following principles:-

A. The CEM channel is the most relevant baseline for comparing re-drivers as they lengthen that same channel (using a PCB extension), with both extended and un-extended channels tested under identical spec stress levels.

B. A better re-driver is one which is able to convey the TxEQ space of Figure 21 with more fidelity than another.

7.1 A CEM-based Method for Evaluating Linear PCIe 3 Re-drivers

Referring to Figure 24, which shows a CEM channel and a CEM channel plus extension, the test procedure is as follows:-

1. Calibrate the BERT for the standard PCI-SIG CEM Rx test procedure (for the specific BERT being used), as described by the CEM test procedures, including V-swing, RJ, Sj, and DMI. (See [19]).

2. Measure the CEM eye using the CEM channel alone (Figure 24 A), at room temperature (e.g. 27 C) for a wide range of TxEQ values, using the standard presets (P0 through P9), plus additional points as indicated by the green ellipses in Figure 25. These points almost fully enclose the TxEQ space prescribed by the PCIe 3.0 Spec [10]. Repeat the measurement at least 5 times, and store each result, in order to overcome random effects in signal acquisition and associated post-processing artifacts by SigTest.

3. Connect the extension channel to the re-driver alone (Figure 26), and configure the re-driver's settings such that it produces P7 most faithfully when driven by the BERT (no jitter or DMI enabled), with an extension channel. Alternately – follow the re-driver vendor recommendations for how to select the optimal re-driver settings for the desired extension channel.

4. Connect the re-driver and the extension after the CEM channel (Figure 24 B), then re-measure the CEM eye for the same set of TxEQ values used in step #2 above. Repeat the measurement at least 5 times, and store each result, in order to overcome random effects in signal acquisition, and the associated post-processing by SigTest.

5. Subtract the Eye Width (EW) and Height (EH) obtained in step #2 from those of step #4, then plot the results versus post-shoot and pre-shoot, for each one of the 5 acquisitions. Positive deltas (extended channel is better than CEM), or the smallest negative ones, are desirable.

6. Change the re-driver settings in search of better ones (e.g. sweep its CTLE peaking), and repeat steps #4, and #5 only.

7. Repeat steps #2 while heating the base CEM channel in a heat chamber to 80C. Repeat step #4 at 80C also, while heating the CEM channel, the re-driver, and the extension channel. Perform the subtraction of step #5 using the 80C results.

8. Repeat steps #3 to #7 with different extension channel lengths of interest.

A better re-driver produces a positive difference, or the smallest negative difference, in steps #5, #7, and #8, at more of the marked TxEQ values of Figure 25.

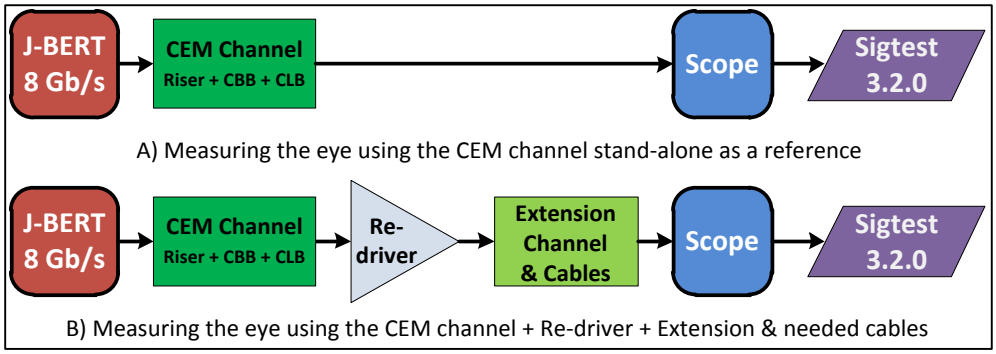

Figure 24 The two measurement setups required by the CEM-based methodology for evaluating PCIe 3.0 re-drivers, each labelled individually (A & B).

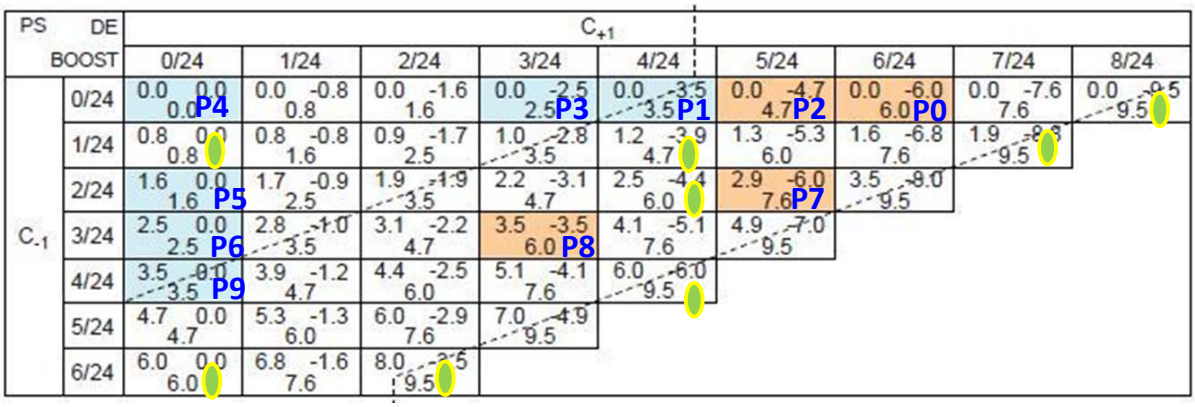

Figure 25 The TxEQ Pre- & Post-shoot space (showing C_{+1} and C_{-1} coefficient values) required by the PCIe 3.0 Base Spec, with the Presets & additional (green) ellipses suggested for bounding the space.

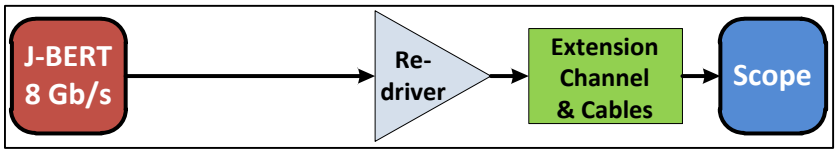

Figure 26 The Measurement setup used to set, or verify, the re-driver equalization & any of its other configurations.

7.2 Some Test Results

Figure 27 illustrates the difference in EW and EH between the extended CEM (using a re-driver) and the CEM channel only (Step #5, and #6 in section 7.1), at room temperature, for one linear re-driver offering. Figure 28 presents the result for a second re-driver. The first re-driver yields the smallest delta, in EW and EH, between the extended CEM and the base CEM. However, none of its settings show positive (or zero) deltas at *all* TxEQ values, thus demonstrating that the re-driver distort the TxEQ space, and invalidates open interoperability. Notice that the second device (Figure 28) distorts the TxEQ space so much, that for several TxEQ values, the delta is highly negative, and that only the middle range of TxEQ settings is produced somewhat faithfully. Both devices are almost equally linear, but the better device of Figure 27 has a frequency domain transfer function which emulates the inverse of PCB loss (compensates) more faithfully than the device of Figure 28, and does so up to about twice the frequency range. Early Heat Chamber results suggest further degradation in the ability to reproduce TxEQ, even for the better device, as expected. Heat degrades a re-driver's ability to equalize, while the extension PCB's loss increases at higher temperature also. These results clearly demonstrate which device is a better choice for Closed-slot design, and also provide a basis for judging the degree of violation of Open Interoperability. This comprehensive methodology –based on the most relevant requirements of the Spec (CEM)-- is recommended for evaluating which is a better re-driver offering.

It is worthwhile noting the "humps" (or re-closing of the eye) in the responses in Figure 27, as the equalization is set beyond the optimal value for any particular transmitter TxEQ. This shows that over-equalizing a channel using a re-driver can distort the TxEQ space just as harmfully as under-equalization. It also demonstrates why the notion, that one could use a re-driver to make an extended link look even better than the maximum Spec channel, is flawed. Early test results (not shown here), involving an actual silicon receiver (different than the reference receiver), also indicate that there is an optimal re-driver setting for obtaining the best sampling eye, and that over-equalization also causes reduction of eye width, and reduction of timing margins to the left and right of the sampling point.

While a good receiver might have a wide-enough range to *work-around* the presence of a re-driver, for instance by asking for alternate TxEQ settings from the Tx, some receivers are known for requesting only one TxEQ value, for a long channel, and are unable to search for better ones.

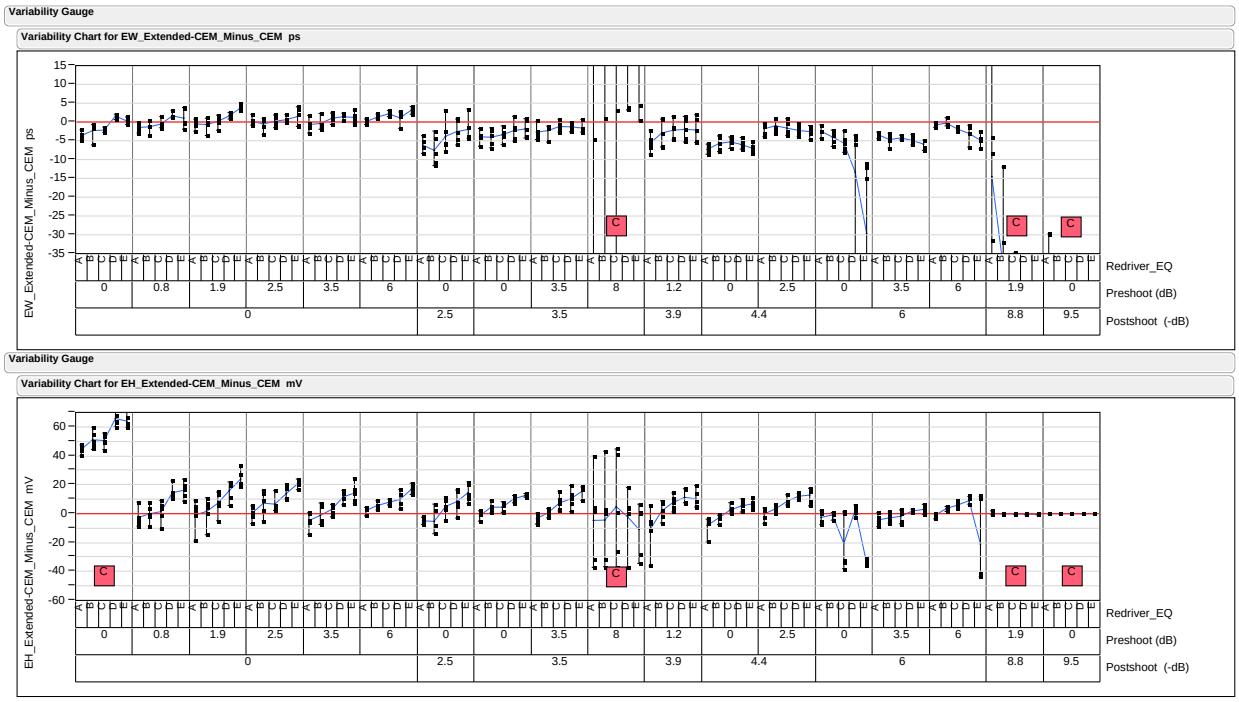

Figure 27 The difference between EW (top) & EH (bot) of the "CEM+10dB (@4GHz) extension" and the "Ref. CEM channel" using one linear re-driver offering, versus the range of TxEQs selected in Figure 21 and a number of progressively higher re-driver equalization settings. Note how no one re-driver setting produces a positive (or zero) delta at all TxEQs. Also note how there seem to be an optimal re-driver settings which are different for different TxEQ values. The red boxes indicate closed Ref. CEM eyes using the Reference receiver.

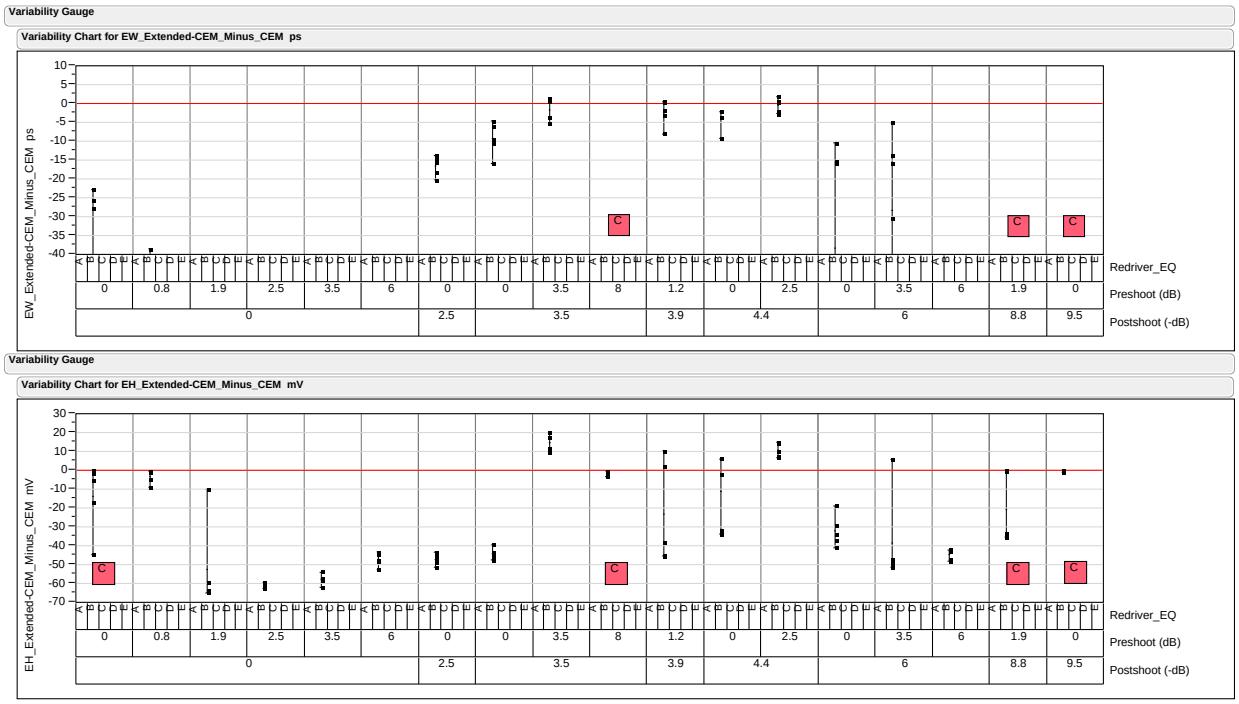

Figure 28 The difference between the EW of the CEM+8dB (@4GHz) of total extension and the "Ref. CEM channel" alone, using a second re-driver offering, vs. the range of TxEQ values selected in Figure 21. Only the most relevant re-driver setting is shown here. Note how no one re-driver setting produces a positive (or zero) delta at all TxEQs. The red boxes indicate closed Ref. CEM eyes using the Ref. receiver.

8. The 10GBASE-KR Standard

10GBASE-KR utilizes 64b/66b encoding (at 10.3125 G-baud) and is part of the IEEE 802.3 Ethernet specification [21]. It is intended to operate using PCB traces up to 1 m long, from Tx to Rx, including two connectors which connect two boards that plug into a backplane. The recommended trace impedance (Zo) is 100 Ohms +/- 10%. There is an N to P side (differential) skew requirement which must be less than the minimum transition time.

KR interconnect is defined between test points TP1 and TP4 as shown in Figure 29. The transmitter and receiver blocks include any external AC-coupling capacitors, and also packages. "Informative" characteristics and methods of calculation for the insertion loss, insertion loss deviation, return loss, crosstalk, and the ratio of insertion loss to crosstalk between TP1 and TP4 are defined in 69B.4.3, 69B.4.4, 69B.4.5, 69B.4.6, and 69B.4.6.4, respectively [21]. These characteristics may be applied to the full path, including transmitter and receiver packaging and other components, such as AC caps.

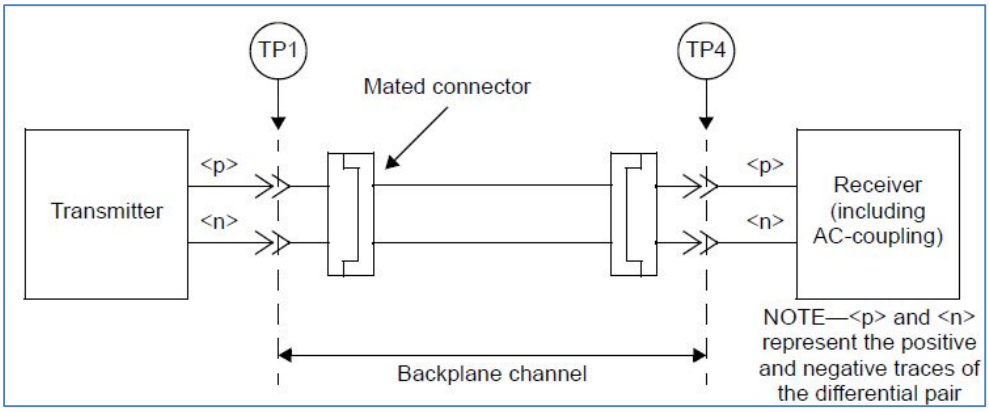

Figure 29 The KR Interconnect reference model defining the reference test points TP1 and TP4 (Fig. 69B-1 of Annex 69B, [21]).

This standard's channel compliance is specified in a way different than PCIe 3.0's. In PCIe 3.0, the channel is considered acceptable if *it is passive*, and if when driven by a Spec Tx plus noise injection, it produces a certain eye opening defined inside a Reference Receiver after application of a Reference Equalization comprising a certain CTLE transfer function and a 1-tap DFE [10]. This is also referred to usually as the "CEM3" eye. For a PCIe 3.0 host with an open slot, a Compliance Load Board (CLB) is plugged into the slot, and a CEM test procedure is followed, aiming to obtain an eye equal to or better than the CEM3 eye [19], after embedding an Rx package model, and applying the aforementioned reference equalization. Conversely, for a PCIe 3.0 card, the card is plugged into the Channel Base Board (CBB), and a CEM eye (or larger) is sought on the other end of the channel after Rx package embedding, and application of the reference receiver.

In KR, however, the channel is specified as a standalone entity, using an informative specification (IEEE 802.3 Annex 69B, in [21]). The 69B specification defines maximum Insertion Loss (IL) bounds, and IL deviation bounds (ILD) at every frequency, as shown in Figure 30. Other aspects of the full channel, such as return loss, and insertion loss to crosstalk ratio, are also specified in Annex 69B.

This approach implies that if such a channel is driven by a Spec-compliant Tx, and signaling is received by an Rx which passes the Rx interference tolerance test [Annex 69A, [21]], then there is a high-confidence (albeit not guaranteed) probability that such a channel would allow such compliant agents to inter-operate successfully. Although the specification is informative, industry treats it as normative (i.e. required).

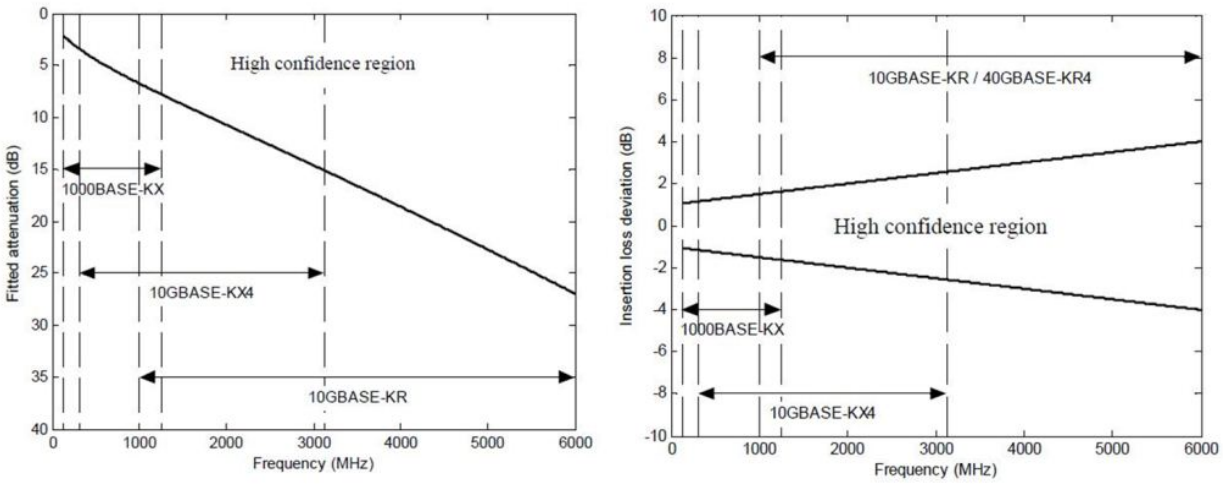

Figure 30 Annex 69B [21] fitted pad to pad channel Insertion Loss (IL), and allowed maximum IL deviation (ILD) from the fit.

8.1 The 10GBASE KR Equalization Standard

8.1.1 10GBASE-KR Equalization Definitions

In 10G-KR, the transmitter is required to be able to produce a wide range of equalizations, slightly larger de-emphasis (post shoot) than that of PCIe 3.0, but less pre-emphasis (pre-shoot). To understand 10G-KR transmitter equalization, refer to Figure 31 which defines the voltage levels of a standard equalized waveform when the Tx is driving a standard load of 50 Ohms per side, through 100-nF decoupling caps.

10G-KR defines so-called Equalization ratios R_{pre} and R_{pst}, which are defined as follows (referring to the voltage levels in Figure 31):-

$$R_{pst} = v_1 / v_2 \ , \quad R_{pre} = v_3 / v_2$$

Table 1 shows the min to max ranges of the TxEQ ratios defined above, along with the permissible values of the DC voltage, when driving the reference load described above. Note that R_{pst}, in dB, can range from +0.45 to -12 dB (including tolerance), while R_{pre} can range from +0.45 to -3.75 dB.

The waveform of Figure 31 is obtained at the Tx output by using an FIR filter which is identical to the one used for PCIe 3.0, and shown previously in Figure 22. Hence, 10G-KR also implies use of the coefficients C_{-1}, C_{+1} and C_0, where one could derive equations which define these coefficients in terms of the three voltage levels v_1 , v_2 , and v_3 defined in Figure 31 . Implementation of coefficient values greater than zero or less than the minimum values defined by R_{pre} (min) and R_{pst} (min) in Table 1, is optional.

Note that unlike PCIe 3.0, the 10G-KR equalization space does not require that the sum of the absolute values of the three coefficients ($|C_{-1}|+|C_{+1}|+|C_0|$) be normalized to 1. Instead, the range of v_2 defined in Table 1, along with the range of the equalization ratios (R_{pre} and R_{pst}) are all that constrains the values of those coefficients.

Table 1 The required ranges of the 10G-KR pre- and post-shoot equalization ratios R_{pst} and R_{pre}, along with the range of the DC level v_2.

Coefficient status			Requirements		
$c(1)$	$c(0)$	$c(-1)$	R_{pre}	R_{pst}	v_2 (mV)
disabled	minimum	disabled	0.90 to 1.10	0.90 to 1.10	220 to 330
disabled	maximum	disabled	0.95 to 1.05	0.95 to 1.05	400 to 600
minimum	minimum	disabled	—	4.00 (min)	—
disabled	minimum	minimum	1.54 (min)	—	—

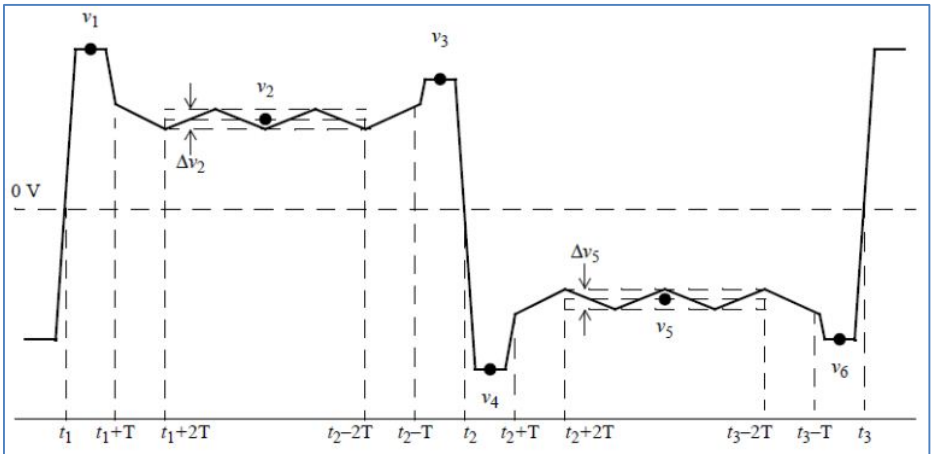

Figure 72–12—Transmitter output waveform

T	=	symbol period
t_1	=	zero-crossing point of the first rising edge of the AC-coupled signal
t_2	=	zero-crossing point of the falling edge of the AC-coupled signal
t_3	=	zero-crossing point of the second rising edge of the AC-coupled signal
v_1	=	maximum voltage measured in the interval t_1 to $t_1 + T$
v_2	=	positive steady-state voltage measured as the average voltage in the interval $t_1 + 2T$ to $t_2 - 2T$
v_3	=	maximum voltage measured in the interval $t_2 - T$ to t_2
v_4	=	minimum voltage measured in the interval t_2 to $t_2 + T$
v_5	=	negative steady-state voltage measured as the average voltage in the interval $t_2 + 2T$ to $t_3 - 2T$
v_6	=	minimum voltage measured in the interval $t_3 - T$ to t_3
Δv_2	=	positive voltage ripple measured as the peak-to-peak value of the difference between the voltage in the range $t_1 + 2T$ to $t_2 - 2T$ and v_2
Δv_5	=	negative voltage ripple measured as the peak-to-peak value of the difference between the voltage in the range $t_2 + 2T$ to $t_3 - 2T$ and v_5

Figure 31 The 10G-KR Transmitter output waveform equalization parameter definitions (Clause 72, [21]).

8.1.2 Summary of the 10G-KR Equalization Training Protocol

When two Ethernet devices are connected, then upon Rx detection in both directions, the link starts Auto Negotiation (AN) (section 69.2.4, [21]). AN is the process by which the two devices communicate

their interface capabilities (at low speed, using Manchester encoding), and settle upon the highest common denominator of link type, and speed. After AN, the link starts the TxEQ training process by which both receivers try to find the best TxEQ setting of the other port's Tx, in order to ensure operation at a BER of 1E-12. The protocol allows a total of only 500+/-5 ms for both link partners to finish their training.

Training is conducted through use of a Training Frame which has four parts: A Frame marker to denote the beginning of the frame, a coefficient update section where the Rx sends to the other agent's Tx its request for a new coefficient setting, a "Status Report" where the Rx reports back to the other agent whether it has been able to tune its Rx using the coefficients that it has requested, and finally a training pattern which consists of 4094 bits from the output of an 11-bit pseudo-random bit stream (PRBS-11).

Both ports (end agents) enter the training state concurrently, and they send each other the above Frame, making coefficient change requests, and training their Rx, until both finally find the coefficients that they prefer, in order to achieve "Receiver Ready" status. Training ends after both agents report back to the other agent that their receiver is trained and ready to bring up the link.

The protocol begins Training (via use of a Training Frame at low speed) where each end point requires the other Tx to set its TxEQ to the "INITIALIZE" values corresponding to R_{pre} = 1.29 (2.2 dB) +/-10%, and R_{pst} = 2.57 (8.2 dB) +/-10%, and that C_0 shall be set such that the peak-to-peak differential output voltage is equal to, or exceeds, 800 mV for a 1010… pattern. From that point on, an Rx is allowed to request from the other agent's Tx to increment, decrement, or hold the coefficient values. However, only one coefficient is allowed to be changed at a time, in each direction. Table 2 defines the required change magnitudes of the three voltage values shown in Figure 31, every time there is a coefficient change request. Note that changing any of the individual coefficients C_{-1}, C_{+1} or C_0, results in a change required for all the three voltage values v_1, v_2, and v_3.

After changing a coefficient, a Tx sends the training pattern using its new equalization settings. The receiving Rx tries to detect the incoming pattern as it adapts its receiver's CTLE (and DFE, or any other resources). When the receiver has completed training, and has determined that it is ready to bring up the link (the implementation of this decision process is left to the PHY designer) it then sets the "Receiver Ready" bit in its Status Report which is sent back to the other agent, in order to inform it of tuning success. The specification for the Rx interference tolerance test [Annex 69A [21]] requires a BER of 1e-12 to pass, but this is not explicitly required for achieving "Receiver Ready". That being said, most users expect, or require, a BER of 1e-12 or better. Training ends if both receivers achieve "Receiver Ready" within 500 ms +/-1%.

Table 2 Transmitter output waveform voltage level requirements related to 10G-KR coefficient updates.

Coefficient update[a]			Requirements[b]		
$c(1)$	$c(0)$	$c(-1)$	$v_1(k)-v_1(k-1)$ (mV)	$v_2(k)-v_2(k-1)$ (mV)	$v_3(k)-v_3(k-1)$ (mV)
increment	hold	hold	–20 to –5	5 to 20	5 to 20
decrement	hold	hold	5 to 20	–20 to –5	–20 to –5
hold	increment	hold	5 to 20	5 to 20	5 to 20
hold	decrement	hold	–20 to –5	–20 to –5	–20 to –5
hold	hold	increment	5 to 20	5 to 20	–20 to –5
hold	hold	decrement	–20 to –5	–20 to –5	5 to 20

[a]Step size requirements for the tap under test apply regardless of the current value of the other taps.
[b]This difference is measured relative to the voltage prior to the assertion coefficient update k equal to hold.

9. Re-driver issues in 10G-KR

The 10G-KR base specification (Clause 72, [21]) is silent on extension devices (including re-drivers). When an active device (other than a Spec-compliant re-timer) is placed in a KR link, the above scheme of open-interoperability (sections 8.1.1 and 8.1.2) based on passive channels meeting the 69B Spec, Tx training protocol, and Rx tolerance testing is no longer valid strictly, for the reasons detailed in the next two subsections on re-driver issues.

9.1 Limiting Re-driver and KR's TxEQ Training Protocol

A limiting re-driver blocks the signal equalization (coefficient) changes (which are requested of a Tx) from reaching the requesting Rx during training. In this case, the issues with such a re-driver are similar to those described in section 6.1, for PCIe 3.0. Please refer to that section for more details. If this is the desired implementation of the system, then the devices are no longer compliant with the IEEE specification for KR, and it is therefore recommended to forego both AN and Training, and simply use an engineered link that is forced to 10G.

9.2 Linear Re-driver and KR's TxEQ Training Protocol

Since the KR Specification has a set of TxEQ range requirements as described in section 8.1.1, and a channel specification (Annex 69B, [21]) which is treated as normative by the industry, then a channel using a linear re-driver cannot be qualified, strictly, as being compliant, and openly interoperable. The 69B specification implies a passive channel. If a linear re-driver is introduced, and its transfer function is convolved with the channel's transfer function, then even if one could show that the resulting SDD_{21} amplitude of the channel is still within the bounds of Figure 30, there remain two issues: The first is that 69B has no phase shift specification against which to check the overall channel response, and the second is that the shape of SDD_{21} will have characteristics that are not normal for a passive channel, due to the presence of the active device.

Similar to the PCIe 3.0 case (section 6.2): There is an issue which arises when a linear active device is introduced into a channel. As explained in section 3.2.2, linearity only implies that the relationship between the incoming and outgoing signal *amplitudes* is that of direct proportionality (a straight line), irrespective of their respective shapes. The active device would alter the incoming signal shape according to its frequency domain transfer function *H(s)*, which is a function of its CTLE, TxEQ, and amplifier frequency response. In principle, it is very difficult to design an equalizer (using a finite number of poles and zeros, and today's gain-bandwidth product process capabilities), to compensate perfectly for the near-linear loss function of a PCB channel (section 1.3). There are extraneous amplitude and phase effects which render the compensation imperfect. In addition, re-drivers have a finite number of programmable settings for their CTLE and TxEQ. The discrete nature of the settings means that there would most likely be either over- or under-equalization, both of which alter the pulse response of the channel. When a channel is extended, and the extension is imperfectly compensated by using a linear equalizer, there is no guarantee that equalization space described in Table 1, would appear to the Rx in the same way it does when the channel is only a passive within the limits of the specification (Figure 30).

Practically, TxEQ adaptation might succeed in finding an operating point for several host/link partner combinations. However, there is no longer *an implied Spec guarantee* that this would hold true always, for any two devices that pass compliance independently, in the open market. Despite this, some designers are using re-drivers in systems that are meant to be openly interoperable. **With the information provided in this document, a designer should be better-prepared to understand the implications, and make his or her own risk-taking judgment.**

Linear re-drivers are, however, amenable to use in a "**Closed System**" (Section 10.1). Such a system could operate with one, or a small and well-characterized (tested) set of link partners, while allowing the TxEQ adaptation protocols on both ends of the link to proceed as defined in the specification. Just as described for PCIe 3.0 (section 6.2), such a system should undergo careful design and validation, in order to guarantee HVM yield, and tolerance to PVT conditions.

10. Designing with Re-drivers

Once it has been decided that a re-driver is appropriate for use in a link, then one would have to address other practical considerations, as described I n the following sections.

10.1 Closed System Using a Re-driver

This is a system using a re-driver, and which is foregone as non-openly interoperable. It is limited to a host, and one or a few devices (such as a few PCIe 3.0 add-in cards, or KR link partners). As mentioned in prior sections, in the case of PCIe 3.0, such a system might deploy a limiting re-driver (if Phases 2 and 3 of TxEQ adaptation are disabled), or a linear re-driver with adaptation enabled. Programming the re-driver settings during validation is the only available degree of freedom in such a system.

A main consideration with closed systems is that they require careful HW validation, in order to ensure that the Host and end device Silicon skews, PCB, temperature, humidity, and voltage variations do not cause failures in the field (HVM). The dilemma is that the system itself is usually not available for such extensive testing, until it has been designed and prototypes delivered. To mitigate such a dilemma, re-

driver evaluation cards could be used to gain a very good level of confidence as to whether such a system would be reliable. This still requires that the Host and end device (or devices) be available for such testing, in addition to a platform with which to perform the evaluation. This latter requirement can cause un-acceptable time to market delays. Erring on the side of shorter link extension is the best way to enhance the chances of a successful design.

10.2 Choice of Re-driver

Choosing the right device for a design always starts at the datasheet. However, there are usually unaddressed properties in published literature. For USB3, SATA, and PCIe 2.0, one of the most important characteristics, of limiting re-drivers, is the amount of available pre-channel equalization (CTLE) at the input of the device. Many data sheets state inches of FR4, but as discussed in section 4, FR4 can differ widely in its loss. In addition, the shape of the equalization curve, as a function of frequency, is relevant, and determines the amount of residual un-compensable ISI. One method to address this is to procure a re-driver evaluation KIT, and measure the eye opening using a JBERT and various pre and post-channel loss values. Despite this, one has to remember that even for those busses, which have simple TxEQ schemes, using re-drivers is not strictly Spec-compliant (Section 4).

For Closed PCIe 3.0 or 10G-KR systems, linear re-drivers offerings differ in their linearity, and frequency response. Linearity has been improving recently, but re-drivers still remain widely different in their frequency response (transfer function, or SDD_{21}), which distorts the TxEq space required by Specs. Section 7 details the recommended method for evaluating a re-driver's TxEQ distortion.

For all re-drivers, the authors have noticed that high temperature affects their behavior differently. Hence high-temperature testing using Evaluation KITs is highly-recommended.

10.3 Placement

Placement of a re-driver is usually dictated by the system architecture and its usage model, in addition to Signal Integrity considerations. A re-driver might be needed on one large coplanar board, where both host and device (or output connector) reside, such as with USB3, where the designer has more placement freedom. In many cases, there might be at least one connector, and in many cases a second connector such as in the case of a backplane. In other cases, host and device might reside on different boards, and the topology might comprise a mezzanine connector, in addition to a slot connector, or a USB connector, etc. More complex multi-connector topologies have also been requested. Backplanes or mid-planes may not be appropriate for repeater placement, due to layout, power supply availability, or usage model restrictions such as re-usability across multiple generations of agents (or link partners).

From a usage model perspective, an open system (where add-in cards or blades have to be swapped) imposes interoperability requirements that cannot be met in the case of PCIe 3.0, and also 10G-KR. In the case of a closed PCIe 3.0 system (where both host and end agent are predetermined and known), the designer might still have some flexibility in exactly where to place a re-driver.

Figure 32 shows the different Signal Integrity considerations, which arise, depending on the placement of a limiting re-driver (e.g. in the case of USB3 and SATA3). Figure 33, on the other hand, shows the

considerations for a system using a linear re-driver (e.g. in the case of Closed-slot PCIe 3.0). These considerations might have to be traded away, sometimes, for more overriding system design requirements.

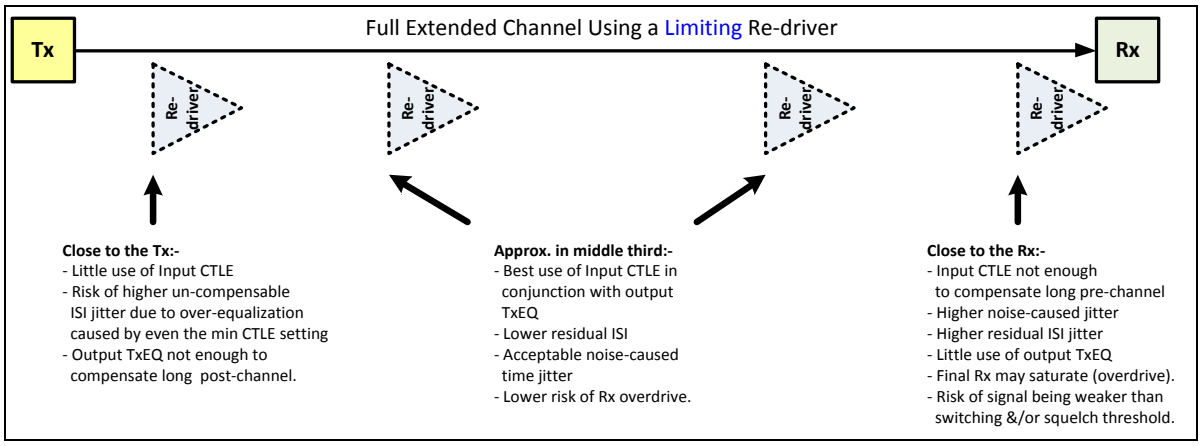

Figure 32 Signal Integrity considerations and tradeoffs guiding the placement of a single limiting re-driver in an extended full channel.

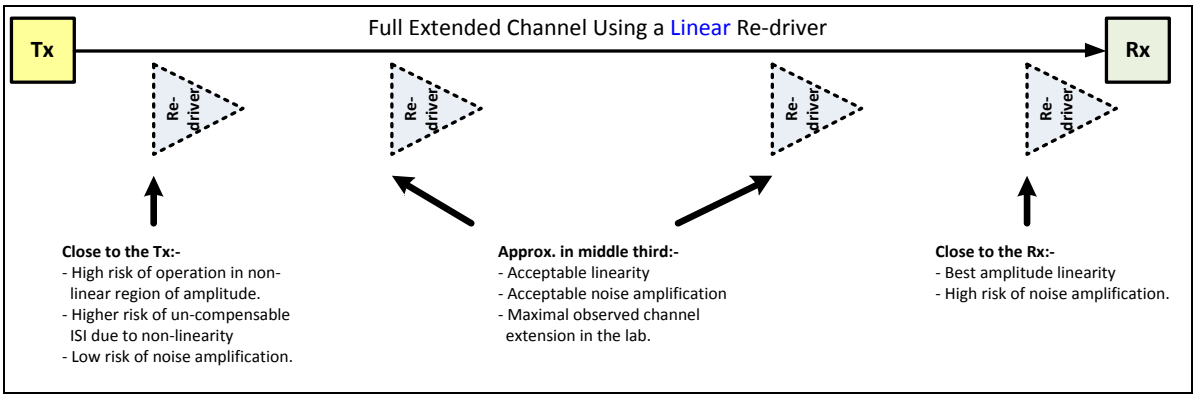

Figure 33 Signal Integrity considerations and tradeoffs guiding the placement of a single linear re-driver in an extended full channel.

10.4 Power Supply Noise

Many re-drivers obtain their power from either 3.3V, 2.5V, or in some cases even 1.5V supplies. Some re-drivers contain internal voltage regulators to control circuit behavior, and also reduce supply noise. Attention should be paid to system supply noise, by following the vendor's bypassing guidelines, in order to reduce deterministic jitter.

10.5 PCB Material Reported in Re-driver Data sheets, and Evaluation Cards

The great majority of re-driver datasheets list channel extension in inches of FR4. There is a wide range of FR4 loss in dB/in (from about 1 dB/in down to about 0.7 dB/in at 4GHz), and temperature affects its

loss by anywhere from 0.2%-0.4% per Degree C, in addition to moisture content. Therefore, it is important to inquire from the vendor as to what the loss of the PCB is in their reported channel extension tables. It is more meaningful to represent channel extension in dB at a bus's Nyquist frequency, than in inches.

Most vendors provide free Evaluation KIT (or board) samples. They come in Edge-connector, SMA, or SMP form factors. Some USB3 re-driver boards also offer Std-A or Std-B connectors.

10.6 Power Consumption

During active mode, re-drivers consume a significant amount of power, ranging from 100 to 200 mW per channel, roughly, depending on the specific device. Some re-drivers offer low-power or sleep modes (e.g. when the load is disconnected), where standing current is reduced, and power consumption drops to 20-50 mW per channel. For a single lane (two channels) the numbers are doubled. Furthermore, for PCIe 3.0, which usually has the highest number of lanes (x4 and up to x32), the total active power can be significant, and should be accounted for when architecting and designing a platform. If a bank of re-drivers can be placed away from the hot spots of a platform, then that might be a good choice, assuming that Signal integrity requirements are also fulfilled.

11. Simulating Channels with Re-drivers

11.1 The Two Modes of Signal Integrity Simulations

Most Signal Integrity simulators are behavioral (i.e. not at the transistor level). They allow a designer to examine channels over HVM variations, including PCB and device PVT skews. They facilitate using a Design Of Experiments (DOE) followed by a Response Surface Model (RSM) fit, which allows prediction of failure rates. These simulators run much faster than transistor-level Spice circuit simulators. Their shortcoming is the need to build (or obtain) a behavioral model which emulates the Tx and Rx circuit behaviors, faithfully. These simulators usually require either a pulse response of the channel (from pad to pad), a netlist, or an S-parameter file (e.g. a Touchstone file) representing the channel. Behavioral simulators usually have two modes: A "Statistical mode", and a true Time Domain (transient), or "Bit-by-Bit" mode.

11.1.1 Statistical (a.k.a. Analytical) Simulation Mode

In the statistical (or analytical) mode, the simulator assumes perfect linearity of all components. It characterizes the full channel from Tx output die pad to Rx input die pad. It does so by either producing (or requiring) a full-channel pulse or step response. It then applies a probability distribution convolution of the bit stream (say a long LFSR random stream), the characterized channel, Tx equalization, and Rx equalization. The resulting final statistical eye is then analyzed mathematically for width and height at a specific BER. Rx equalization settings might also be adapted, in search for the best solution.

A Statistical simulator is suitable for analyzing full channels that comprise a linear re-driver, assuming that the re-driver is receiving signals whose amplitude is small enough to guarantee that it is indeed operating in the linear region, and not suffering from saturation (or compression, 3.2.2).

A statistical simulator is not fit for analyzing full links with limiting (non-linear) re-drivers. One might think that such links could be analyzed in the statistical mode, if the full channel is broken into two completely separate simulations: One from the source Tx, through the pre-channel, to the re-driver input pads, and a second from the re-driver output Tx, the post-channel, to the final Rx post-equalization. The problem with such an approach is that a statistical simulator assumes perfect linearity, and cannot handle what happens inside the re-driver, since the re-driver's limiting amplifier (post EQ) reshapes the signal non-linearly. Even if one assumes that the output wave shape could be assumed to be an ideal pulse train with equalization and proper edge rates, the user would have to gather all the jitter information at the output of the re-driver's equalizer followed by its non-linear amplifiers, and use such jitter as input for the post-channel simulation. A very tedious task, especially when performing DOE simulations, where the pre-channel might have tens of variants, each one of which producing a different set of jitter numbers, each one of which, in turn, having to be combined with the tens of post-channel simulations.

11.1.2 Bit-by-bit (Transient, or "Empirical") Simulation Mode

In this mode, a simulator performs a true time domain transient simulation (picosecond by picosecond), using 1E4 to 1E7 bits, and accumulates the eye inside the Rx, for later width and height analyses. These simulators are slower than the statistical ones. Due to their longer run times, they are used to accumulate an eye over at most 1E6-1E7 bits (at the time of this writing). They build a bath tub diagram, then they project the eye width and height to higher bit counts (lower BER) mathematically. The bit-by-bit mode of a simulator is suitable for analyzing both linear and non-linear (limiting) re-drivers.

11.2 General Simulation Guidelines for Re-drivers

The advent of IBIS-AMI [22], and the ability of simulators to read IBIS-AMI models make it possible to simulate channels with re-drivers, behaviorally. However, there are still some issues to be considered. When acquiring models from a re-driver vendor, it is important to inquire about the following:-

1. **The Silicon, Voltage, and Temperature (PVT) corners that the model covers**

Silicon circuit designers use several (sometimes tens) of corners for gauging the performance of their circuits. There are many circuit variables; such as N-type versus P-type transistor cross skews, resistors and capacitors which are unrelated to transistors, interconnect metal resistance and capacitance, plus voltage and temperature corners. Thus, if a vendor gives only 3 model corners, then there might be pre-determined statistical assumptions, such as: N and P transistors are both slow or both fast together (i.e. no cross-skews), and that the back end of the process (resistors, caps, and metal) might also be assumed to be slow or fast at the same time that the transistors are. This builds pessimism into the model, which would yield pessimistic (albeit safer) simulation results, yielding shorter channel extension capability. Model corners which predict temperature dependence (along with Si fast, typical, and slow skews) are very important, since many circuits usually lose gain and bandwidth at high temperatures.

2. **The number of Standard Deviations (Sigmas) of variation at model corners**

A model which represents 6-Sigma corners will --when convolved with the channel variation corners-- yield very pessimistic statistical results.

3. **How well the model and simulator represent Rj and supply noise (Dj)**

The deterministic non-ISI jitter of a re-driver is a function of its supply noise (which is determined to some degree by the system power supplies), and also the variation in its output switching rise and fall times (Duty Cycle Distortion). Furthermore, Random jitter (Rj), and supply-caused jitter are also functions of the chosen output voltage swing. Model creators might not be accounting for such dependencies, especially if they are not the original designers of the device.

4. **How well the model represents the input SDD_{11}, and the output SDD_{22}**

One must ascertain whether the model represents the input and output terminations correctly, and if a package model and die input capacitance are included.

5. **Whether the model has the same programmability of the VOD, CTLE, TxEQ, and other settings as the physical part**

6. **How well the model reflects the phase response (of SDD_{21}) in the case of a linear re-driver**

7. **How well the model represents the onset of non-linearity of a linear device**

A model which assumes perfect linearity, irrespective of the input voltage amplitude, and provides only the equivalent of a frequency domain transfer function, can misrepresent the true signal shapes at the output of a re-driver. For instance, S-parameter models of a linear re-driver are sometimes offered. While S-parameter models allow very fast simulations, they force the assumption that the device is perfectly linear in the relevant region of operation, which might not be the case always.

The reader is advised to **exercise the model stand-alone**, e.g. with a simple resistive load, in order to explore its properties, before advancing to full-channel simulations. One should understand what the model corners cover, and how they should be used. If the model lacks corner settings, or if the corners do not cover PVT far enough, then one has to keep additional design margin. Conversely, statistically-unreasonable corners could add pessimism.

12. Ascertaining Link Robustness of Closed PCIe 3.0 and 10G-KR Systems using Re-drivers

Due to TxEQ adaptation (or training) in both protocols, re-drivers present similar issues to both buses, in addition to what was discussed in sections 6 and 9. When a system designer chooses to use a re-driver, a determination has to be made as to which vendor's device to choose, and how to select its settings for maximum production yield. The selection unlikely to be based on accurate simulations, due to the lack of accurate corner models of all of the host, re-driver, and add-in card (AIC) or link partner end agents, all simultaneously. But simulations might give an initial approximate method to evaluate options.

Usually, an initial a determination is made for the placement of the re-driver (e.g. on a riser card, a backplane, or somewhere between the host and a PCIe 3.0 slot on the mother board, etc.). This is based on initial approximations, using the device datasheets, the loss of the overall channel, the availability of power supplies, and ensuing pre and post channel lengths. See section 10.3 for some placement considerations.

The final selection of the re-driver's DC gain, amplitude, and peaking equalization settings have to be made during system testing and validation. Those settings then have to be appropriate for the life of the system over a number for years, where several parameters could vary.

During high-volume production, the items described in the following sub-sections could be changing over time. They would all affect whether the receivers on both ends of a PCIe 3.0 or 10G-KR link could find BER 1E-12 operating points or not. For PCIe 3.0 this applies to both the case of disabled Phase 2 & 3, and the case where a linear re-driver is used while phases 2 & 3 are enabled. Similarly for 10G-KR's Training.

12.1 Silicon Skews of the Host and End-agent (AIC or Link Partner)

While Tx and Rx variations are present in spec-compliant un-extended links, their significance increases with the presence of a re-driver. The reasons are that the distorted EQ space might be requiring the receivers to work harder (go to an extreme AGC, CTLE, DFE, etc.), in order to support a re-driven extension, and the added variations of the re-driver and link extension.

Transmitted jitter can be a strong function of process corners (the cold and slow process corner in CMOS devices usually produces the most jitter). The Rx on the other hand is more sensitive due to its input equalization's sensitivity to process corners. An Rx's CTLE equalizer gain, peaking, overall bandwidth, and to some degree the amplitudes of the DFE coefficients, are process-sensitive and could shift, especially if the device is not designed to compensate, at least partially, for process variations.

It is not always easy to obtain skewed devices from manufacturers for validating a system, due to the logistics of fabrication plants. In some cases, even design teams do not have access to all possible skews, unless the Fab runs skew lots, or cherry-picks dies, occasionally. Hence, link margining using whatever parts are available, plus temperature and voltage skewing, could be used as a minimal alternative in order to ascertain link health.

12.2 Silicon Skews of the Re-driver

Since a re-driver is an added element to a channel, unaccounted for by Specs, then its variation is also not accounted for. Hence, re-driver variation has an added effect on a link.

When designing a system which uses re-drivers (e.g. USB3, SATA3), or a closed PCIe 3.0 or 10G-KR system which uses re-drivers, it is important to study the effects of the re-driver's semiconductor process skews on link health. Gain-bandwidth is a function of the process corners, whose number can be numerous. Vendors may, or may not, be able to supply parts (or evaluation boards) that represent significant skew corners, due to logistical limitations in coordination with a fab, especially when dealing with small volumes. Hence, some vendors rely on simulations, or cherry-picked devices, to ascertain if their device remains within its datasheet specifications. Some semiconductor designs attempt to stabilize their part -- versus skew corners-- by using post-Silicon trim (or fusing), in order to "trim" variation. It is important to understand, from the vendor, the expected range of variation, which is not always stated in a datasheet.

12.3 PCB Material Variations

The transmission line impedance and loss are also functions of HVM and PVT, even from one vendor. While it is realistic that one vendor's impedance might remain under control to within +/- 5% to +/-7%, and loss to be within +/-5%, a 5% loss change in 10 dB of channel extension amounts to 0.5 dB shift in required equalization.

12.4 Supply Voltage Effects

Deterministic uncorrelated jitter added by a re-driver is a function of noise in both its Tx's and Rx's power supplies. Good repeater designs tend to have internal voltage regulators which provide an accurate internal DC rail, with reasonable noise rejection. Nevertheless, some attention must be paid to designing the local external supply, with minimal high-frequency AC noise.

A careful reading of serial bus specifications would reveal that jitter characterization usually requires activation of all lanes in a port, not just one bit. Similar margining should be performed, when using a re-driver, in order to understand the effect of its internal (and external) supply noise during heavy switching, and also internal circuit (and package) coupling effects between its channels.

12.5 Temperature (and Humidity) Effects

Temperature has the most significant effect on a system, more so than well-regulated supplies. Temperature affects not only host and end agent Tx and Rx analog circuit bandwidths, gain, and jitter, but also the loss of the connecting PCB, and the re-driver itself. Compared to an unrepeated link, the added pieces are only the re-driver and the additional PCB (or perhaps cable) extension. Hence, to a first order, only the effect of temperature on the re-driver and the added PCB need special attention.

PCB loss variations from room temperature (27 C) to about 80 C (a 53-C temperature shift) and humidity can range anywhere from 10-20%, or 1-2 dB of extra loss for 10-dB of extension [16] [17]. Furthermore, the re-driver's jitter and equalization are functions of temperature. The host and device Tx and Rx jitter and equalization are also functions of temperature, but those are present in links even before extension. It is only when an Rx has to exercise more of its resources to service a link with a re-driver that temperature variation of that receiver might become a limiter.

Furthermore, in many cases, a link is trained when the system is still cold (at least the PCB and host are cold), then has to operate when the chassis' internal temperature rises to its allowed maximum air temperature (e.g. 75-80 C). That is sometimes referred to as "Train Cold Run Hot" (TCRH), and the opposite is also possible (THRC). When one examines all the components that could change, then unless the receivers on both ends of the lane have enough dynamic equalization range, a system with a re-driver might become marginal. It has been demonstrated in the lab that this usually requires reducing the amount of extension, until proper BER operation is still achieved in TCRH cases.

One might argue that a linear re-driver's EQ should be set to be higher than needed, in order to compensate for the channel extension, and handle high temperature. In a validation environment at lower temperatures, test results would actually appear to reduce the amount of possible extension as demonstrated in section 7.2.

In the case of limiting re-drivers (for USB3, SATA, or PCIe 2.0), over or under-compensation is transformed into un-compensable data-dependent pure jitter, which would also reduce link margins. A possible alternative would be to select the re-driver's optimal settings at a mid-range temperature, as a way to deal with TCRH (or THRC), thus minimizing the amount of un-compensable jitter created at either extreme.

12.6 Example of Adding Overall Sources of Variation

First, we have to assume that the temperature variations affecting the Tx and Rx, on both ends of the un-extended maximum Spec link, could be absorbed by those two devices, as they are supposed to be Spec-complaint at all of their PVT corners. What remains, then, is the variation in the PCB extension, the re-driver, the distortion of equalization space caused by the introduction of the re-driver into the channel, and how that might tax the receivers on both ends of a link.

Consider a PCIe 3.0 system using a re-driver to extend the channel by 10dB, with phases 2/3 enabled. It trains cold, and then it has to operate at 80C. It is evaluated by using nominal devices (corner skews unknown) from host and AIC vendors, and the re-driver settings are selected at a mid-range temperature of 55 C. Over HVM, temperature, and humidity, the following variations might occur in the link:-

- PCB loss of the channel extension might change by +/-0.5 dB due to manufacturing and impedance variations.
- PCB loss of the channel extension changes by +/-0.5-1 dB due to a 25 C temperature rise or drop around 55 C.
- The re-driver Silicon skew is different and its equalization shifts by +/- 1 to 2 dB.

The above scenario shows that a re-driven channel can experience a total change on the order of up to +/- 3.5 dB. This represents 15% of the PCIe 3.0 maximum channel loss Spec of 23.5 dB. Therefore, it is advised that closed systems using re-drivers be reasonably stressed during validation. Hence, it is advisable to try to exercise the following combinations or tests during validation:-

- Obtaining impedance and loss corner PCB boards of all main pieces in the system.
- Asking the vendors of all the 3 active devices in the link (host, re-driver, end agent) for skew corners, but keeping in mind the low likelihood that such a request could be met.
- Optimizing the re-driver gain, amplitude and equalization settings at a temperature mid-way between the coldest and the hottest in the operating range.
- If re-driver's skew devices are unobtainable, then a qualitative indicator of link robustness could be had by adjusting its equalizations up and down by +/- 1 to 2 dB around the optimal setting. If the system continues to function properly, over the full temperature range, then that increases confidence.
- Agents differ in their ability to indicate link health, but if possible; evaluation of the eye margins (or BER), on both ends of the link, using all the above combinations, is advisable.

13. Re-timers

Re-timers are mixed-signal analog and digital devices. A Re-timer has a Clock Data Recovery circuit (CDR, [5] [6]) similar to a SerDes PHY. A re-timer converts an incoming analog bit stream into purely digital bits that are stored (staged) internally; it then re-transmits the digital data anew. Thus, a re-timer breaks a link into two distinct sub-links, which are completely independent from each other from a Signal Integrity (analog amplitude and timing) perspective. Since a Spec-compliant re-timer re-establishes signal shape and transmitted jitter just like a compliant Tx would, it could double channel reach. Hence, re-timer allows longer reach than a re-driver, since it does not suffer from jitter accumulation. Re-timers render pre- and post-channel analyses simpler than the case of re-drivers, due to the complete separation of the two. They also tend to provide debug capabilities, such as eye margining, and Compliance patterns. Re-timers are substantially more complex than re-drivers, larger in die and package size, are more expensive, and are offered by fewer vendors than re-drivers.

13.1 Re-timer Types

As described in the introductory section (2.2), re-timers are of two types: **Bit Re-timers (Protocol-un-aware, or TxEQ Training-incapable)**, and **Intelligent** or **Protocol-aware Re-timers (TxEQ Training-capable)**. The following subsections describe the main features of each type.

13.1.1 Bit Re-timer (Protocol-unaware or TxEQ training-incapable) Micro-architecture

Such a device usually comprises only a CDR [5] [6] to convert the incoming data stream into a purely digital one, then re-transmitting it anew, with fixed equalization, and new jitter. Figure 34 shows the conceptual micro-architecture of a Bit Re-timer. A bit re-timer is usually programmed by the designer for a specific output TxEQ, and cannot participate in TxEQ training such as for PCIe 3.0 [10] or 10G-KR [23]. These re-timers are fit for busses which do not require such training, such as SFI [24]. Hence, in the case of SFI, such a re-timer is not really “Protocol Un-aware” since Equalization training is not part of the protocol, and such an appellation would be non-descriptive.

There are several CDR architectures, as described in [6], which is highly-recommended reading. Here we chose one of the robust ones, for the sake of illustration only. Although a re-timer could recover the clock from the data itself alone, a Reference Clock (RefClk) is usually provided, in order both to speed-up locking onto the correct data rate, and also to avoid false locking. The input RefClk is used to synthesize a nominal baud-rate frequency using a Voltage-controlled Oscillator (VCO) composed of a number of delay stages (e.g. 5 stages). The output of each VCO stage is fed to a Phase Interpolator (PI) which is used to create the final clock used to sample the data (eventually becoming the recovered clock at the true incoming data rate).

Since the incoming data rate is determined by the RefClk of the sending Tx, there will be a slight frequency difference between that clock and the RefClk shown in Figure 34. The difference is allowed by Specs usually (in PCIe 3.0, a RefClk can be within +/-300 ppm of 100 MHz). To compensate for the small frequency delta, the PI uses the VCO phases fed in from the PLL, to generate the “Recovered Clock”, by interpolating between an adjacent pair of those phases. The PI keeps “sliding” its selection of what pair to use, and the ratio between them, such that it adjusts its output phase continuously. The PI performs its function based on feedback from the CDR. In the meantime, the CDR tries to position the sampling clock close to the center of the incoming data eye, using its own internal phase detector and control

decision loop. The desired location of the sampling clock is communicated via controls sent back to the PI, guiding it in sliding and adjusting its Recovered Clock phase.

In addition to the CDR's internal control, a Finite State Machine (FSM) is used to adjust the Input CTLE, and Gain (AGC) in order to provide a more open eye. The algorithm used to achieve this is usually proprietary. In addition, the recovered bits are (or can be) used to feed a DFE generator, which adds equalization to the output of the Gain stage, in order to open the eye further (as explained later in 13.2.6). DFE control is usually also under FSM control, and is part of the proprietary algorithm used by the Receiver. At the end of this process, the receiver is expected to be achieving the desired BER (e.g. 1E-12 for PCIe 3.0). A re-timer usually comprises other functional blocks, which will be described later in section 13.2.

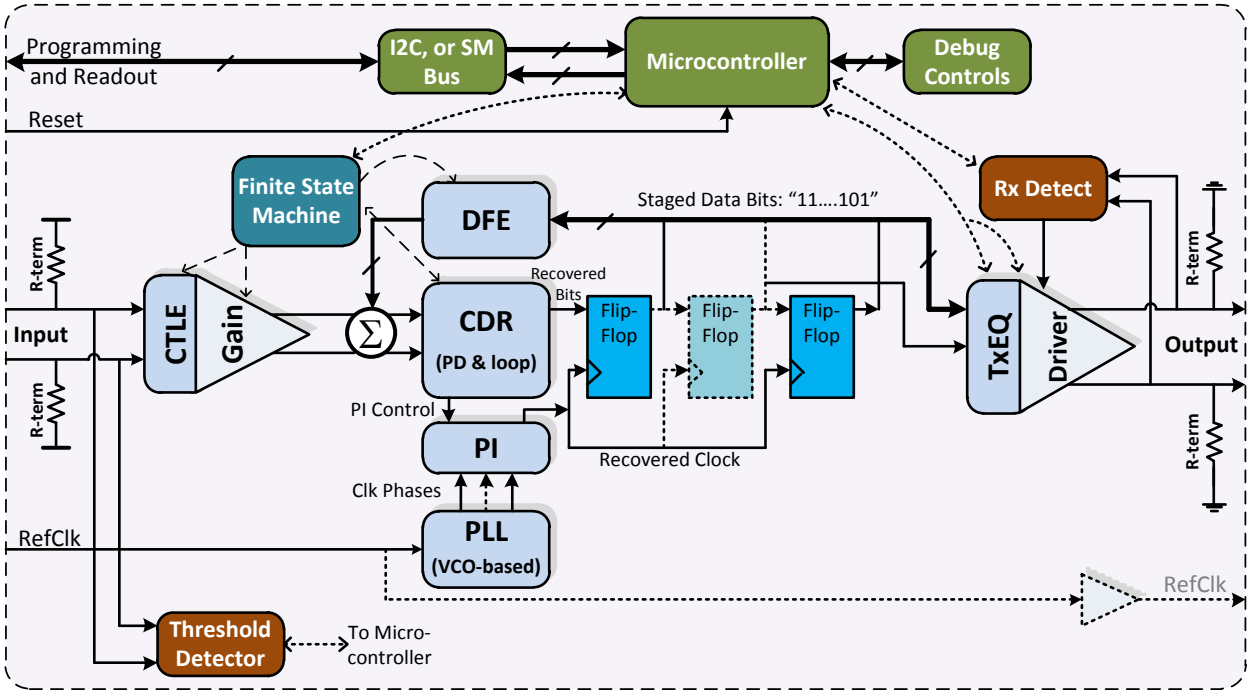

Figure 34 Conceptual micro-architecture of a "Bit Re-timer", showing input CTLE, AGC, DFE, a CDR, and a data path connected directly to an output driver with TxEQ.

13.1.2 Protocol-aware (Or TxEQ training-capable) Re-timer Micro-architecture

Such a device comprises all the blocks of the Bit Re-timer described in section 13.1.1, and in addition contains a more sophisticated microcontroller which allows it to participate in TxEQ training transactions (Figure 35). Training must occur with the two other link partners at the re-timer's input and output (upstream and downstream, respectively), including for example Phase 0 through Phase 3 of TxEQ Adaptation in PCIe 3.0 [10] (section 5), or TxEQ Training in Ethernet Fabric's KR [23] (section 8.1). Protocol-aware re-timers are the only kind of SerDes repeater which is capable of being truly fully Spec-compliant with PCI3 [10] and 10G-KR [23]. They are the most-expensive, and are also offered by a small number of vendors (at the time of this writing) due to their complexity, their need for both analog and digital design expertise, and protocol IP ingredients, simultaneously with access to a high-gain-bandwidth analog-digital semiconductor process.

Assuming that such a re-timer is truly both protocol-compliant and also electrically-compliant, then its use is the most straightforward from a signal integrity point of view --but not necessarily from a programming, or debugging, standpoint. Its use in PCIe 3.0/4.0 or 10G-KR should support open-interoperability, as described in sections 5.5 and 8.1. More details on the PCIe 3.0 and 10G-KR versions of re-timers are given in sections 14 and 15.

In PCIe, there are 3 clocking architectures: Common Ref Clock with or without Spread Spectrum Clocking (SSC), Separate Ref Clock with Independent SSC (SRIS), and Separate Ref Clock No SSC (SRNS) [10]. The SRIS and SRNS architectures cause mismatch in the data rates between sender and receiver, which must be compensated for in PCIe 3.0 by insertion or removal of extra bits using Skip Ordered Sets (SOSs). The SRIS mode is more stressful than SRNS, and causes the highest mismatch in data rates, due to the added independent clock modulation on both ends of the link. It is the mode which sets the maximum requirement on buffer depth, and also the maximum number of re-timers in a link.

A protocol-aware re-timer has two main modes of operation: Forwarding, and Training (or Execution). In training mode, the re-timer tries to set itself and its link partners for the appropriate bit rate, port widths, etc., and complete TxEQ adaptation. After training, the re-timer moves into forwarding mode (also called data mode), where it behaves like a bit re-timer, with some exceptions, such as insertion (or removal) of SOSs, in order to match the baud rate throughout the link. Note that a re-timer could start in forwarding mode upon system reset, in order to allow slow back channel communication, before moving to training, and then the eventual return to forwarding as the final high-speed operational mode.

The Physical Coding Sublayer (PCS), shown in Figure 35, performs bit stream encoding and decoding as required by the different bit rates (PCIe 1.0, 2.0, or 3.0), and it also compensates for the different data rates by removal or insertion of SOSs. This latter function depends on the clocking architecture of the system, as explained above. It also depends on the direction of data. In a system where the re-timer shares its RefClk with one of the agents, then traffic directed toward the other agent does not require data rate compensation, whereas traffic in the opposite direction would invoke SOS manipulation. More details are given, in the following section, about the additional blocks contained in both Bit and Protocol-aware re-timers.

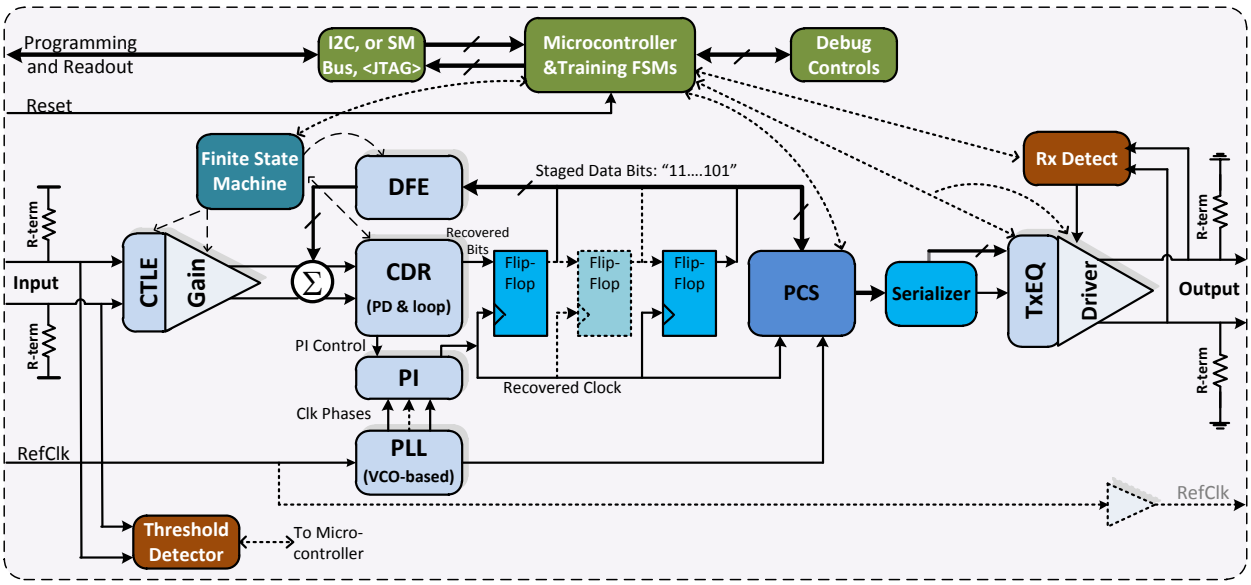

Figure 35 Conceptual micro-architecture of a "Protocol-aware Re-timer", showing input equalization, a CDR, and a data path connected to a PCS, Serializer, and output TxEQ driver, with a Micro-controller.

13.2 Additional Re-timer Micro-architectural Blocks

Figure 34 and Figure 35 showed the basic micro-architecture of re-timers, but specific implementations differ. Re-timers usually come as an identical pair (or pairs) in one package (similar to re-drivers, as depicted in Figure 4), in order to accommodate both the sending and the receiving directions of one or more lanes (or links). Though the basic mechanics of re-timer operation and some of their basics building blocks where described in sections 13.1.1 and 13.1.2, more detailed descriptions of those blocks, and others, are given next.

13.2.1 Input Terminations

These are similar to the input terminations of a re-driver (section 3.1.1).

13.2.2 CTLE and Gain (or AGC) Stage

These circuits are similar to those in a re-driver (section 3.1.2). The main difference is that in a re-timer, input equalization and gain are usually adaptive, under the control of the FSM which controls the algorithm used to guide the EQ and the CDR into the best sampling point (using the most-open eye), and consequently the lowest BER.

13.2.3 Clock and Data Recovery (CDR) Circuit

A CDR is a circuit which can extract both clock and the data from an incoming (specially-encoded) data stream [5] [6]. A CDR depends on the requirement that the incoming data stream has a known ratio of zeros to ones (e.g. 1:1, which also maintains DC level balance), enforced in SerDes specifications via encoding in the transmitter's PCS. A CDR employs a Phase Detector (PD) and a decision circuit used to position the sampling clock, in the best eye location, leading to detecting zeros and ones properly. In addition, a counter usually accumulates the number of detected zeros and ones, over a moderate period of time, and produces an error signal which is a function of the imbalance between the detected zeros and ones. The CDR's control loop uses the fed-back error to try to improve the exact sampling location.

Furthermore, to achieve proper detection of zeros and ones, the CDR requires an eye which has been opened by receiver's equalization resources (such as Gain, CTLE, and DFE). In advanced receivers, a Finite State Machine (FSM) runs a proprietary algorithm which steers the equalization for the best open eye, to ensure success in CDR locking at the target BER (or better) for that bus.

13.2.4 Reference Clock (RefClk)

RefClk is usually a low-frequency clock (e.g. 100 MHz) which feeds a PLL. The PLL produces a high-frequency internal clock running at the bus's bitrate. Some receiver CDRs can operate without the need for a RefClk, but such designs have limitations, such as low frequency jitter accumulation, and potential false clocking [6]. Most CDRs require an input RefClk, as explained in section 13.1.2. The allowable frequency delta between independent clock sources is bounded a bus's specification's allowable RefClk deviation from nominal (e.g. +/-300 ppm in PCIe 3.0, and +/-100 pm in 10G-KR).

13.2.5 Data Staging Registers (Flip-flops)

A CDR's main outputs are the extracted clock and data bits. The CDR's recovered clock is used to capture and store the recovered bits into a shift register (consisting of serially-connected flip-flops). The bits are pure digital data lacking any analog content from the pre-channel. In addition to being sent out to the PCS layer, the staged bits are also fed, in parallel, to the DFE generator (see section 13.2.6).

13.2.6 Discrete Feedback Equalization (DFE)

DFE [2] is a non-linear equalization used to cancel the residual pulse response tails (and ripples) in the incoming analog signal at the output of the CTLE. It helps produce a narrower recovered single pulse response, thus reducing ISI. The idea is to add, or subtract, small amounts of Unit Interval-wide (UI-wide) voltage (or current) at progressive bit intervals around the main cursor, in a fashion which counters the tails and ripples in the channel's pulse response. The amplitude of each DFE tap is controlled by the receiver's FSM, which also controls other resources such as CTLE and AGC. Eventually this also drives toward a better balance of the received zeros and ones, implied by the CDR's convergence loop. In addition to channel pulse response tails, DFE also cancels some of the reflection artifacts caused by discontinuities in the channel. The number of DFE taps ranges from 1 post-cursor tap to several, in addition to pre-cursor taps in some designs.

13.2.7 Finite State Machine (FSM) Controller

This digital state machine usually has a proprietary algorithm which manages adjusting the input Gain (AGC), CTLE, DFE settings, and other parameters, in such a way as to open the eye as widely as possible, before it enters the phase detector (PD) inside the CDR. The FSM also helps locate the best position to sample the eye. It is usually programmable through firmware (inaccessible to the user), in order to optimize the CDR's performance. The FSM may choose to train equalization upon startup then freeze it, or continue to train throughout operation, depending on the sophistication of the design.

13.2.8 Physical Coding/Decoding Sublayer (PCS)

This digital block exists in protocol-aware re-timers, and performs decoding and encoding of data into balanced zero and one streams (e.g. using 128b/130b encoding for PCIe 3.0, or 64b/66b encoding in 10G-KR). The PCS also manages insertion or removal of Skip Ordered Sets (SOSs), as described in section 13.1.2, in order to compensate for differences between the received bit rate and the RefClk-implied bit

rate. The parallel data output of the PCS is fed to the Serializer. In the case of a bit-re-timer (a.k.a. protocol-unaware re-timer), a PCS is not needed usually.

13.2.9 Serializer

A Serializer converts the parallel digital data (at a lower frequency) to a stream of serial bits at the link's high bitrate. The serializer pre-drives the output transmitter. The simplest serializer is a synchronous parallel-load serial shift register.

13.2.10 Transmitter Output Driver

The driver is the final power amplification stage which drives the output load. It is usually merged with the TxEQ function described next. The Output Differential Voltage swing (VOD) is usually adjustable from about 0.4V to 1.2V, for PCIe and KR. The driver can usually be turned off by the micro-controller during low-power modes (e.g. squelch, or no output load).

13.2.11 Output TxEQ

This circuit is usually merged with the output driver, and is used to add de-emphasis and usually pre-emphasis also to the output. It is a true FIR filter, as described in section 3.1.5. TxEQ is usually under micro-controller control, since its settings have to be programmable by the re-timer itself, in order to allow the device to participate in the PCIe 3.0 or KR TxEQ adaptation or training.

13.2.12 Output Terminations

These are similar to the output terminations of a re-driver; please refer to section 3.1.6.

13.2.13 Microcontroller

The controller manages communication with the I2C (or SM) bus, and acts as the intermediary between those I/O bus protocols, and the internal architecture of a re-timer. It allows receiving any external presets (or user configurations), in addition to communicating factory settings to the correct registers inside the re-timer. The controller configures the receiver FSM which manages input EQ adaptation, and CDR convergence. That functionality is indicated by the dotted arrows in **Error! Reference source not found.**. The most important function of the micro-controller in a protocol-aware re-timer, is communicating with the PCS (which extracts and inserts information from/into the training sequences) and managing all the phases of TxEQ adaptation from beginning to end. Hence, the controller is responsible for transitioning the re-timer from Forwarding, to Training (or Execution) mode, and back to Forwarding. In addition, the controller communicates with the input Signal Level (Threshold) Detector, and the Rx Output Load Detector, in order to manage the power states transitions of the re-timer. It also manages any available debugging features, such as eye margining, and status readouts which can be accessed by the user, through I2C, SM, or even JTAG.

13.2.14 Input Idle Threshold (Squelch) Detector

This function is similar to that in a re-driver, as explained in section 3.1.7.

13.2.15 Receiver Detection (Rx Detect)

This function is identical to the one described in section 3.1.9 for re-drivers. The main difference is that the output of the Rx detector is usually sent to the micro-controller, in order to manage power states, and input/output termination settings.

13.2.16 I2C, or SM Programming Bus

Re-timers require initial configuration by the user and status readout, both of which are usually achieved by using an I2C or an SM bus [12] [13]. Data sheets give the user extensive register tables indicating what parameters can be set or read out.

13.2.17 Debugging Logic

Some re-timers allow the user to margin the post-EQ receiver eye width and height, in both directions, to assess link health, or also allow detecting error rates (via communicating with a sophisticated PCS). Since a re-timer is transparent in the link (after training --if applicable), it is difficult to determine if bit errors originate downstream or upstream of a re-timer – however eye margining tools can quickly locate the poorly performing channel. While no eye margining block was shown in Figure 34 or Figure 35 (for simplicity), some re-timers do offer such a capability. Debug features can also allow tracing the progression of the re-timer's state machines during link initialization and operation. Debugging can be achieved through a set of read and write registers connected to the micro-controller, and accessible via the I2C or SM bus.

14. PCIe 3.0 Re-timer

An Engineering Change Notice (ECN) to the PCI-SIG 3.0 specification defining re-timers [20] has been approved. That ECN guarantees interoperability with specification-compliant PCI Express 3.0 devices. The PCI Express re-timer must be protocol aware. The PCI Express re-timer runs the adaptive phases, (phases 2 and 3, described in section 5) of the link equalization protocol, in each direction. This effectively splits the link, on each side of the re-timer, into two completely separate and independent "Link Segments" at the analog level. Therefore, the channel on either side of the re-timer can be up to the full worst-case channel allowed by the PCI Express specification, while still guaranteeing interoperability. This also means that the electrical requirements for re-timer receivers and transmitters are the same as those for standard PCI Express devices, as defined in the PCI Express specifications [10]. A PCI Express re-timer does not need to have a receiver that exceeds the requirements defined for standard devices in the PCI Express specification, and implementations can be based on standard PCI Express physical layer IP.

There can be at most 2 re-timers in the link between a PCI-Express upstream port and a downstream port. The major reasons for the 2 re-timer limitation are timeouts and Reference Clock parts per million (ppm) tolerances, which will be explained in subsequent sections. Figure 36 shows the possible PCI Express link scenarios with one or two re-timers, along with the terminology used in the PCI Express Extension Device ECN [20]. The portion of the link on each side of the re-timer is called a "Link Segment". The re-timer's own upstream and downstream ports are defined as the "Upstream Pseudo Port" and "Downstream Pseudo Port", respectively. Unlike traditional PCI Express devices and upstream and downstream ports – the re-timer is not directly visible to software through the PCI Express configuration space. The PCI Express re-timer is software-transparent.

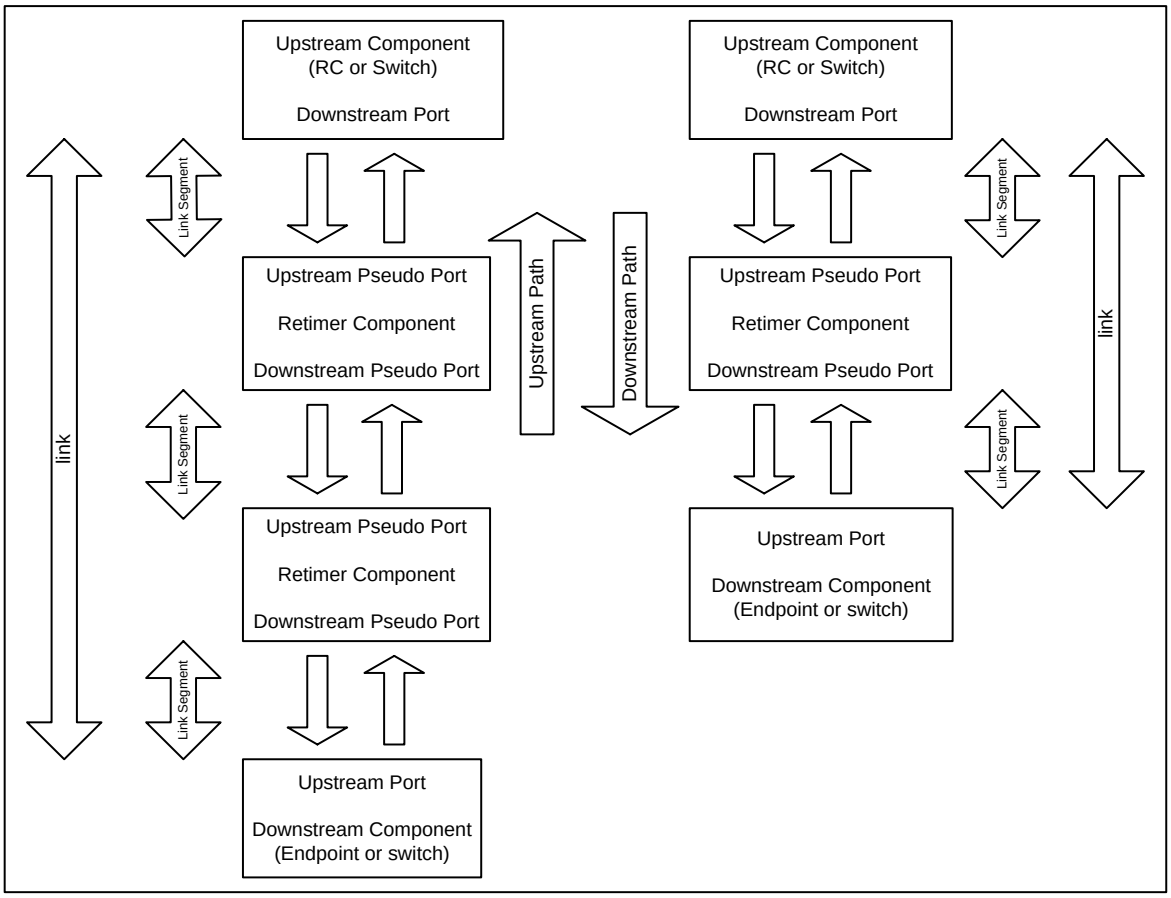

Figure 36 PCI Express links with one or two re-timers.

14.1 PCI Express Re-timer Link Training and Status State Machine Overview

The PCI Express re-timer does not have the same link training and status state machine as do standard PCI Express upstream or downstream ports. At a high level - the PCI Express re-timer operates in one of two functional modes, **Forwarding** mode and **Execution** mode. The re-timer starts --and spends most of its time-- in the Forwarding mode, where incoming traffic is mostly retransmitted through the re-timer without changes. Technically, it could start in the Forwarding mode, even at 8 GT/s, although the standard protocol does not allow that, as a matter of course. The re-timer does have to modify some fields in training ordered sets – but never modifies or interacts with transaction layer and link layer packets.

For a small portion of the time – the re-timer operates in Execution mode, where it must initiate traffic directly, and act like a standard PCI Express upstream port on the re-timer Upstream Pseudo Port, and like a standard PCI Express downstream port on the re-timer Downstream Pseudo Port. The main case, where the re-timer operates in Execution mode, is that when the adaptive phases (Phases 2 and 3) of the PCI Express link equalization protocol take place.

14.2 Rx Detection, Re-timer Orientation, and Hot Plug

The PCI Express protocol requires that silicon attempt to detect receiver terminations, repetitively. Once receiver terminations have been detected, silicon must attempt to train, and if there is no response to training, then silicon must go into a "transmit compliance" mode, where a specified compliance pattern is transmitted repetitively. This test mode is designed to make analog transmitter compliance testing possible, by connecting any compliant re-timer directly to a 50-ohm terminated real-time oscilloscope.

The re-timer does not present 50-Ohm terminations, on power up. The reason is that if it does, then even if nothing is connected to the far re-timer port --or something is connected but not active– then the device on the other side of the link will end up stuck in transmit compliance mode. A device stuck in transmit compliance mode consumes unnecessary power, and potentially interferes with hot-plug. Since re-timers power on without their receiver 50-ohm terminations applied, they perform Rx Detection repetitively, in the same way as a standard PCI Express device on both Pseudo Ports (section 3.1.9). Once Rx detection is successful on a Pseudo Port, the re-timer applies its own 50-ohm receiver terminations on the opposite Pseudo Port. In addition, the re-timer has error detection rules which will trigger it to re-attempt Rx Detection when a hot-pluggable device is disconnected.

Re-timers also detect which direction is upstream and which is downstream, dynamically. This is obtained from information in the training sets – thus allowing a standard re-timer to be usable in a reversible cable (where either end can be plugged into the upstream or downstream component).

14.3 Speed Detection and Speed Support

Re-timers must support all PCI Express speeds up to the maximum one that they are designed for. For example – a PCI Express re-timer that supports 8.0 GT/s must also support 5 GT/s, and 2.5 GT/s. Re-timers modify information in the training sets so that an unsupported speed is never advertised to the upstream or downstream components. For example – a re-timer that only supports 2.5 GT/s and 5.0 GT/s would modify training set information and remove any advertisement of support for the 8 GT/s link speed. Re-timers must properly detect the analog signaling rate on the link, whenever a re-timer receiver detects exit from electrical idle, and begins to receive analog signaling. The re-timer must ensure that it has the rate detected properly, before it starts forwarding received information to its other Pseudo Port. This detection must be performed within small time windows defined in the PCI Express ECN [20], in order to ensure that the delay in forwarding does not impact the PCI Express training protocol.

14.4 Phase 2 and Phase 3 Adaptive Tx Equalization Training at 8 GT/s

Whenever a PCI Express re-timer detects an exit from electrical idle, properly detects the signaling rate, and starts forwarding information, then it is in the Forwarding mode by default (Figure 37). When the protocol reaches the beginning of the adaptive TxEQ phases (Phases 2 and 3), the re-timer switches into the Execution mode, and starts to execute (actually initiates) the training protocol itself (independently) on both the upstream and the downstream Pseudo Ports, simultaneously.

To begin with – the re-timer runs Phase 2 on its Upstream Pseudo Port, in exactly the same way a standard PCI Express Upstream Port would behave (as was described in section 5.3), and runs Phase 2

on its Downstream Pseudo Port in exactly the same way a standard PCI Express Downstream Port would behave (Figure 37).

Once the Upstream Pseudo Port has completed optimizing the TxEQ for its own receiver in Phase 2, it keeps the link in Phase 2 by not updating the appropriate status fields in the training sets, until the Downstream Pseudo Port has also completed optimizing the TxEQ for its receiver.

Once the downstream Pseudo Port has completed optimizing the TxEQ for its receiver in Phase 3 (section 5.4), then the upstream Pseudo port indicates the completion of Phase 2, and the link segment moves to Phase 3, where it behaves as a standard PCI Express upstream port.

Though the Downstream port has finished Phase 3 (Figure 38), it stays in Phase 3 by continuing to request, repetitively, the final TxEQ which was selected for its receiver, while the upstream Pseudo Port completes Phase 3. Once the Upstream Pseudo Port completes Phase 3, then both Pseudo Ports return to the forwarding mode, at the same time.

The re-timer keeps states synchronized, in order to prevent introducing new error conditions due to its presence. This helps prevent such cases where, for instance, one Link Segment completes Phase 2 and Phase 3, and moves on to the next phase of the protocol, while the other Link Segment encounters errors and exits TxEQ negotiation, before successfully completing Phase 3.

The re-timer must be able to complete Phase 2 and Phase 3 equalization for its own receivers in 2.5 ms, so that the standard Upstream and Downstream Port equalization timeouts are never violated.

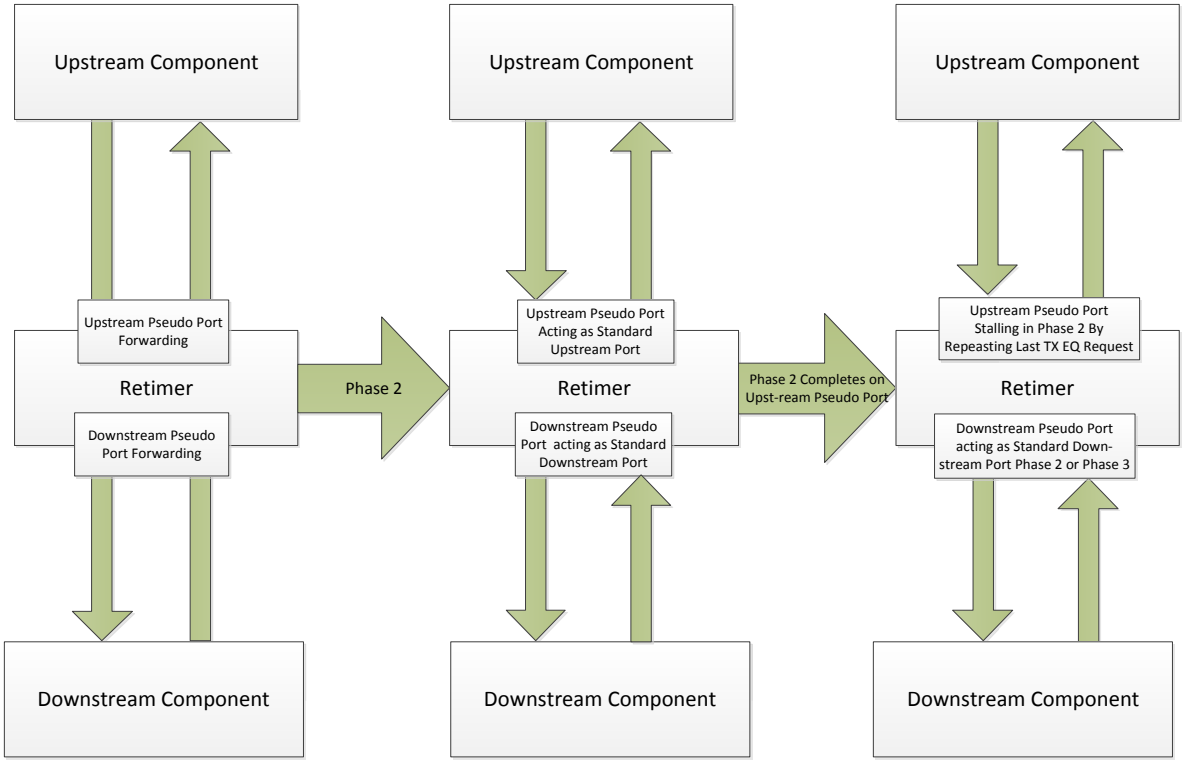

Figure 37 Phase 2 and Phase 3 Sequence with a re-timer --Part 1.

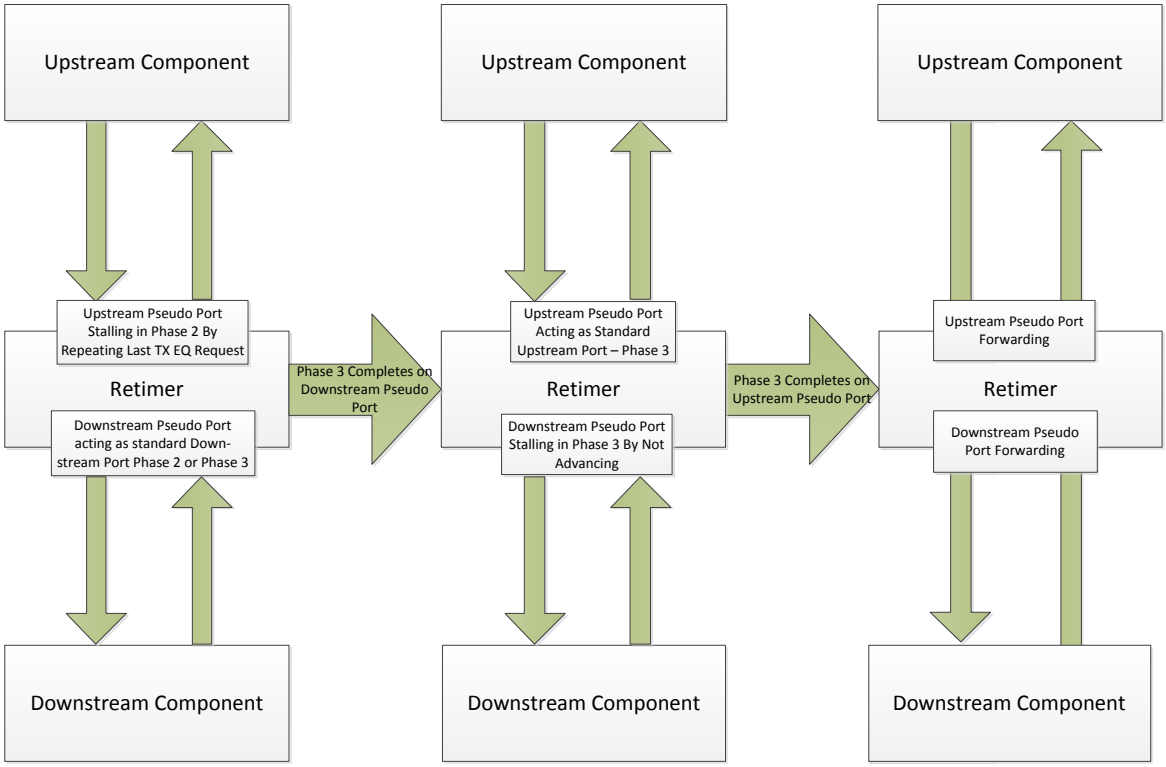

Figure 38 Phase 2 and Phase 3 sequence with a re-timer --Part 2.

14.5 Clocking Architecture Support

The PCI Express specification [10] supports several different clocking architectures. The architectures supported in the latest revision of the specification are a common clock architecture (where a common reference clock is distributed to both the upstream and downstream components), and an independent reference clock architecture, where the upstream and downstream components have independent reference clocks. Both clocking architectures allow the use of "Spread Spectrum Clocking" (SSC), which is defined by PCI Express as 30-33 KHz clock modulation of up to 0.5% below the nominal reference clock frequency of 100 MHz. The PCI Express ECN covering re-timers [20] allows support of either clocking mode. A PCI Express re-timer available on the market may or may not support both architectures. Re-timer implementations must have significantly larger elasticity buffers to support the independent reference clock mode and have increased jitter tolerance at low frequency.

The PCI Express protocol requires a transmitter to send SKIP Ordered Sets (SOSs) with a minimum and maximum average rate that varies with the clocking architecture. The average rate of SOS transmission is about 10 times higher for the independent reference clock architecture with SSC (SRIS) than with the common clock architecture. The SOSs are designed to be a set that a receiver can either expand, or reduce, in order to keep internal buffers from underflowing or overflowing due to ppm variations between the transmit and receive clocks. A re-timer is required to modify the SOSs to keep its receiver data buffers from ever underflowing or overflowing when forwarding data from one re-timer Pseudo Port to the opposite Pseudo Port. Furthermore, the PCI Express protocol requires that changes to the SOSs are done consistently on all lanes at the same time, and failure to do so may break existing PCI Express receivers. The PCI Express transmitter requirements provide enough SOS content, such that a maximum of two re-timers may modify SOSs between the Upstream and Downstream Component. If additional re-timers are present, then there is no longer a guarantee that the final receiver will have enough remaining SOS content to function – this is true for both clocking modes.

14.6 Debugging and Electrical Testing

All PCI Express transmitters are required to support a Compliance mode where they automatically start transmitting a compliance pattern [10] if 50-ohm terminations are detected, but the other side of the link does not attempt to train. This allows transmitter testing to occur by connecting devices directly to a 50-ohm-terminated oscilloscope. Re-timers are also required to support this mechanism such that standard electrical testing can still occur for a re-timer by itself, or for a system which includes a re-timer. In addition, re-timers are required to interoperate with the standard loopback modes defined by the PCI Express specification such that loopback would still work between the Upstream and Downstream components when one or two re-timers are in the path. There is also an optional "Local loopback" definition in the re-timer ECN which allows loopback to be run directly to and from a re-timer which supports this optional feature. The optional "Local loopback" helps to isolate the location of errors that occur using the standard loopback modes between the Upstream and Downstream component in a system.

15. 10G-KR Re-timer

There is a variety of ways that a 10GBASE-KR re-timer may be implemented. One fundamental rule applies regardless of the link up flow methodology: The re-timer must be protocol-aware and capable of performing KR Training, as defined by IEEE 802.3 Clause 72 [18]. By performing this training, the re-timer is effectively breaking the channel in two from an electrical stand-point, and therefore creating two separate KR Channels. Each side of the channel must conform to the IEEE 802.3 Annex 69B channel parameter requirements. For additional details on this training process, please refer to the specification and section 8.1 of this document. In addition to performing training with each end point, the re-timer must either perform, or at least not interfere with, the other phases of bringing up a 10G-KR link, specifically “Auto Negotiation” and “PCS Block Lock”.

15.1 Auto-Negotiation

The IEEE 802.3 standard’s Clause 73 “Auto Negotiation” (AN) is the first step required during the process of establishing a 10G-KR link. At the beginning of the link up process, the two Ethernet ports attempt to establish a connection using AN “Base Pages” (lower speed signaling) to communicate between each other, thus allowing the devices to determine the highest common link speed. They can also configure Forward Error Correction and perform additional functions through “Next Pages” such as enabling Energy Efficient Ethernet (EEE) and other proprietary link modes. AN is used for link speeds from 1G through 100G, and will continue to be used for future IEEE backplane protocols. If the two devices are both capable of 10G-KR, and are not both capable of a higher speed, then AN will resolve to establish a 10G-KR link. When link is resolved to 10G-KR, the two devices have 500 ms to complete the AN process, which includes KR Training and PCS alignment. Once the AN Base Page communication has been completed, the devices transition to the KR Training phase of the link up process. The re-timer’s role in this process is either to complete AN with both devices independently, or to pass the signal through itself, and allow AN to complete directly between the end points. See section 15.4 for more info on the full link up flow.

15.2 KR Training

IEEE 802.3 Clause 72 defines the PMD control for 10GBASE-KR, which includes KR Training. This is when the equalization for the 10G signaling is adapted to the present channel (see section 8.1.2). In a system using a re-timer, the channel is divided into two completely separate electrical domains, where side of the link must conform to the Annex 69B channel specification, and training must occur on each side of the re-timer. As stated above, a KR re-timer must be capable of performing this training. Once the KR Training phase of the link up process is entered, the re-timer should close off the path between the two devices and train each side of the link individually. Link training gives each receiver time to converge on the incoming signal, which includes adjusting the TxEQ settings of the connected transmitter, so at the end of training, all four transmitters and receivers are adapted to the connected channels.

15.3 PCS Alignment

The final phase of establishing a KR link is transitioning to data mode, and aligning the PCS layer of the PHY to the 64B/66B encoded data (IEEE clause 49, [21]). In the case of a re-timer with a PCS, a

connection to each end point can be established, independently, because the re-timer can supply the encoded data needed to maintain link. The re-timer would go through AN and KR Training with the device, then move into data mode with the re-timer supplying the 10G data until all channels are trained. While link is only up on one side, the re-timer should send the end point "Remote Fault Ordered Sets" (IEEE 46.3.4, [21]) to indicate that the far end device is not ready. When the re-timer has finished training both sides of the link, then it can begin passing through the encoded data, and a full link can be established through the re-timer. If the re-timer does not have a PCS, then the PCS alignment will occur between the two end points with the re-timer simply passing the data along. In this case, the main concern is enabling a clean and seamless transition into data mode (no loss of signal), as well as making sure that all data paths are running on a congruent clock.

15.4 Link Up Flow

In the case of using a re-timer to extend a 10G-KR channel, there are two basic options for bringing link up, depending on whether or not the re-timer implements a 64b/66b PCS. If the re-timer has a PCS, then it can establish an independent link with each end point, and enable an extended channel once both sides have been trained. The other option is only to implement the KR Training phase of the link up flow within the re-timer, and allow the rest of the AN process to occur between the two end points, passing *through* the re-timer. In this case, the re-timer must be able to recognize when the devices transition from AN "Base Page" exchanges to KR "Training Frames" (section 8.1). At that time, the re-timer should close off the channel, then train both channels independently. Once both sides have been trained, then the re-timer can indicate to the end points that it is ready to transition to data mode. At that time, the re-timer would go back into a pass-thru mode, exchanging information between the end points using the equalization that was converged on during training. This transition into data mode should be seamless, so that the end points do not lose CDR lock, and can lock to the incoming PCS signal as quickly as possible.

There are advantages and disadvantages to both link up methods. A PCS-enabled re-timer would be more complex and expensive, since it would need an AN layer as well as a PCS layer. Whereas a re-timer which passes-through the AN signaling between end points, and does not have a PCS, may be more prone to timing and interoperability issues, but is a simpler device. The most important consideration with a "Training-only" re-timer is developing a robust link up flow. The recommended method is to make sure that the re-timer is not indicating (to either end point) that it is ready to move to data mode, until all receivers are ready to move to data mode. The way to accomplish that is by controlling the "Receiver Ready" indication on all of the channels. Receiver Ready is a bit within the KR "Control Channel" (which is a portion of the low-speed KR Training Frame) used to indicate to the link partner that the local receiver is trained, and is ready to move to data mode. A Training-only re-timer should only propagate the Receiver Ready indication when both of its ports are trained and are ready to move to data mode. In other words, a re-timer should indicate "Receiver Ready" to an end point, only when both re-timer ports, are trained AND when "Receiver Ready" is seen from the other end point. An example of this type of link up flow is shown in Table 3 and Figure 40 with its associated legend in Figure 39.

Table 3 KR re-timer "Link Up Flow" Description.

Link Up Stage Description	Note
1. Ports enabled and initialized	Re-timer passes through End Point's AN signal. End Point's complete AN process with each other
2. Independent link training	KR Training occurs independently on either side of the link
3. End Point #1 achieves Receiver Ready	End Point #1 local receiver is trained (RX2 and TX2), RX Ready indicated to re-timer. RX Ready not propagated through re-timer because re-timer ports are not converged
4. Either port of the re-timer achieves Receiver Ready	Does not matter which re-timer port converges first (4.1 or 4.2), RX Ready not indicated/propagated to End Point #2 until both re-timer ports are converged
5. Both Re-timer ports achieve Receiver Ready	Begin propagating RX Ready indication from End Point #1 to End Point #2 when BOTH re-timer ports are trained
6. All ports achieve Receiver Ready	All ports are trained and ready to move to data mode. Re-timer propagates RX Ready indication from both End Points
7. Transition to Data Mode	Re-timer propagates received data using equalization determined during training. Transition into data pass through mode (SEND_DATA) must be seamless

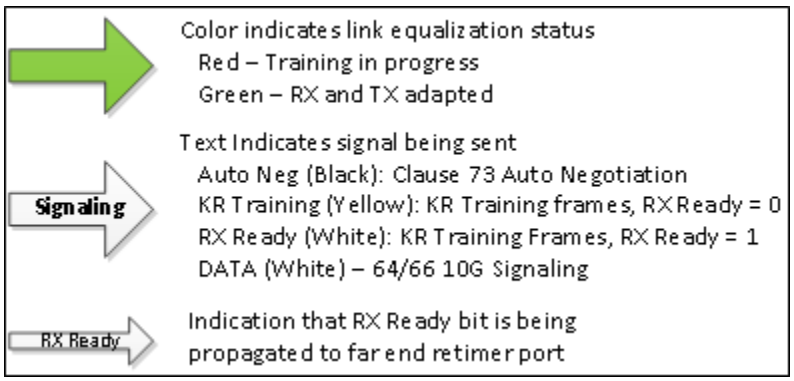

Figure 39 Legend for the KR re-timer "Link Up Flow" Block Diagram in Figure 40, shown below.

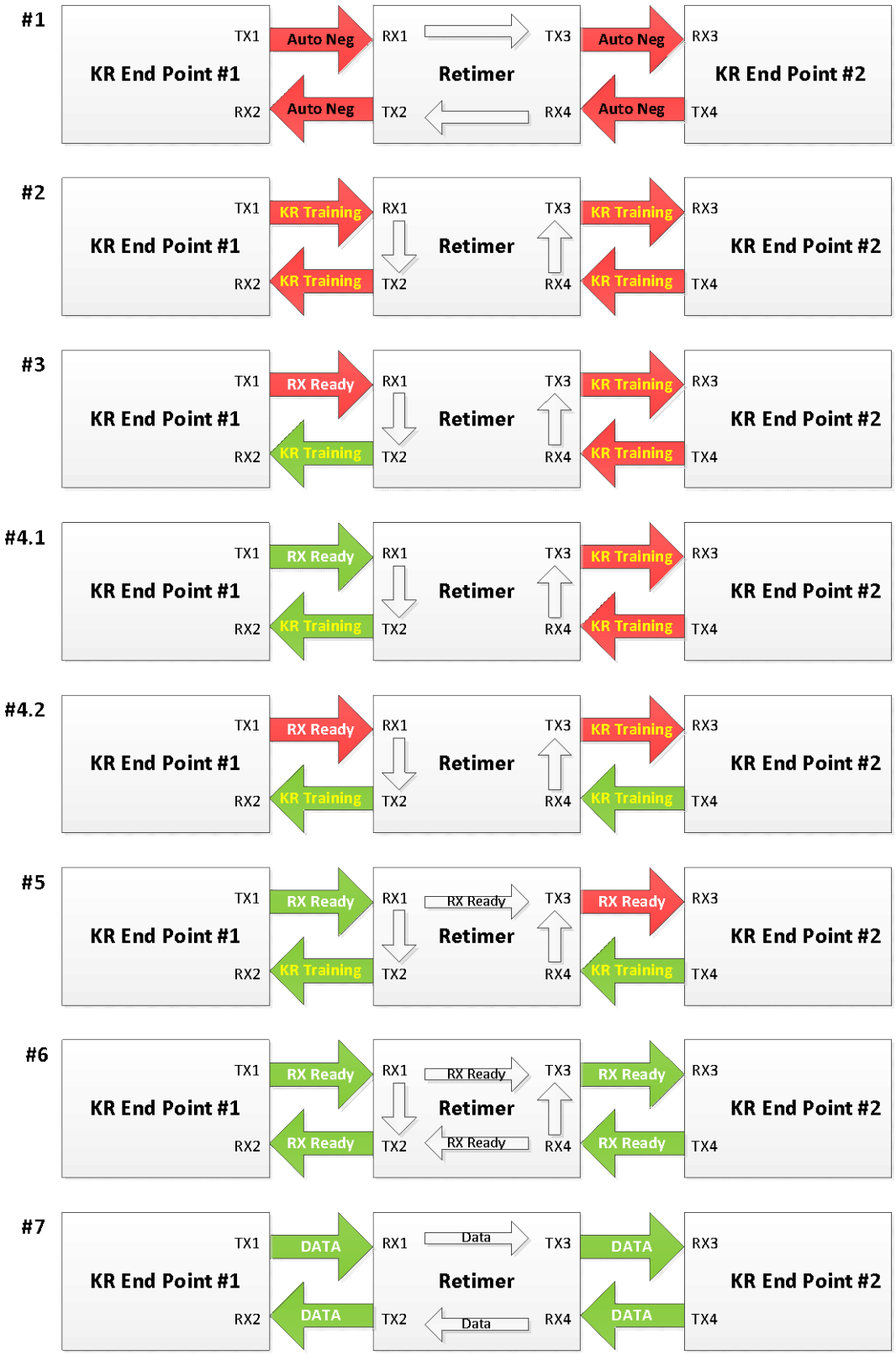

Figure 40 KR re-timer "Link Up Flow" Block Diagram.

15.5 Other Considerations For 10G-KR Re-timers

15.5.1 Energy Efficient Ethernet (EEE)

IEEE Clause 78, which addresses EEE [21], can cause issues with a protocol-aware re-timer if the re-timer device is unaware that EEE is enabled. EEE enables the devices to turn off their transmitters intermittently --without losing the clock recovery or channel equalization-- which could be seen by a re-timer as link going down. EEE could only be enabled with a re-timer if the re-timer is specifically EEE-capable, and completes AN with the end points, which is when the EEE configuration is enabled. In the case that the re-timer is only doing training and not AN, EEE should be disabled.

15.5.2 Cascaded Configuration

In the event that a KR channel is longer than what can be supported by a single re-timer, it may be desired to use multiple KR re-timers in a cascaded configuration. This can potentially create additional link establishment concerns, but if the re-timer implements either one of the two "Link up Flows" recommended in section 15.4, then the re-timer should be capable of establishing link up in such a cascaded configuration, without introducing new issues.

PCS- and AN-capable devices should be able to support cascading easily, since they can establish links independently with each end point, and/or a connected re-timer. As the other PHYs are enabled, and once link has been adapted and brought up on each channel, the full data path will be enabled and ready to pass traffic.

If the re-timer is only capable of supporting KR Training, then it is critical that the re-timer implement the method of only propagating "Receiver Ready" after *both* of its ports have been trained, and it is receiving "Receiver Ready" from the far-end link partner. The following example (Figure 41) of a KR Channel, with multiple cascaded re-timers in the path, highlights the expected behavior of a "Training-only" re-timer. There are four different channels to be trained, labeled #1 through #4, and each re-timer is in a different phase of the link up process.

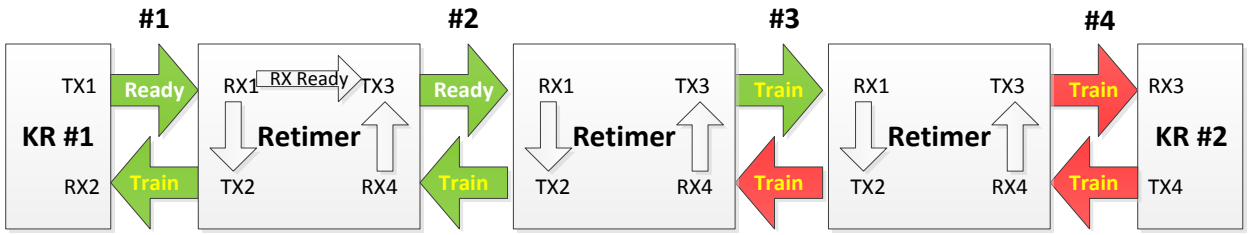

Figure 41 KR re-timer Cascaded "Link Up Flow" Example.

In this example, channel #1 is fully trained by both KR End Point #1 as well as the re-timer. Since the End Point is trained, it is signaling "Receiver Ready" to the re-timer, and because both re-timer ports are trained, the re-timer is propagating that "Receiver Ready" indication on channel #2. Since the re-timer is not receiving "Receiver Ready" on channel #2, it does not indicate "Receiver Ready" back to End Point #1. Also notice that even though channel #2 is fully trained, and the middle re-timer is receiving

“Receiver Ready”, it does not propagate it onto channel #3 because it does not yet have both of its local receivers trained yet. All of this shows that even though half of the channels are fully trained, the link will only be brought up when all channels are fully trained, and the “Receiver Ready” indication is generated by both End Points and propagated through each re-timer.

15.5.3 KR Re-timer Link Debugging

KR Interoperability can be problematic between two PHYs from different industry vendors, and adding an active device to the path would only make it more complicated. It is important for a KR PHY to have certain debug features in order to allow investigating interoperability issues, and a KR Re-timer should have similar features. If link is not getting established, then it will take a strong knowledge of both the IEEE KR specification [21], as well as the end point and re-timer configuration to determine the cause of the problem. In order to enable this, at a minimum the re-timer device should be able to indicate the following to the user:-

- Link status of all ports
- KR Training State of all ports (IEEE Figure 72-5, [21])
- Frame Lock status of all ports (IEEE Figure 72-4, [21])
- Receiver convergence status of all ports
- Initial and Final Transmitter equalization coefficients
- Link partner “Receiver Ready” status
- Received eye health

The following features are desired, and may help facilitate debug, but they are not required:-

- Loopback modes at various levels (or layers) of the device
- Rx/Tx Parts Per Million (PPM) clock differences. Knowing the various clock rates --or at least the differences between them-- can help debug a variety of receiver issues, such as elastic buffer overflow problems
- Time-to-Link measurement
- Tx Feed-forward Equalization (Tx FFE, or TxEQ) adaptation requests made by re-timer to a link partner
- Applicable re-timer receiver parameters (VGA, DFE, etc.)

Following are additional requirements if the re-timer contains a PCS and an AN layer:-

- Auto Negotiation State of all ports (IEEE Figure 73-11, [21])
- PCS Block alignment status of all ports
- Remote and Local fault status of all ports

15.5.3.1 Interoperability Testing

Assuming that all of the end points and re-timers in a link are compliant with the Tx and Rx electrical requirements of the IEEE specification [21], then the most critical part of ensuring that a system which utilizes a KR re-timer will behave as expected, is testing the functional interoperability of the system. This interop testing should happen as early on in the design process as possible, so that any issues that

are discovered can be addressed prior to building systems. Testing should, at a minimum, include analyzing the full re-timer and End Point configuration for the following:

- Analyzing “Time to Link”
- Testing link up reliability when resetting any device
- Passing traffic reliably across comparable channels

Testing interoperability between the End Points, without a re-timer in the path, as well as testing a single End Point with the re-timer in a loopback mode, can help facilitate this testing, as well as help isolate any issues that may exist.

15.5.3.2 Transition Timing issues

One very common source of issues in KR Interoperability, that is made more complicated by the presence of a re-timer, is transition timing. This is the amount of time that it takes the devices to transition from AN signaling to KR “Training Frames”. Every device behaves differently, and devices can be sensitive to long timeouts, or to the presence of unexpected data during this transition. The re-timer should be capable of tolerating long transition times from end points, but should minimize the amount of time that it adds to this transition, while it is closing off the channel to train each side. As mentioned above, the transition from KR Training Frames to the passing through of KR data, should be seamless.

15.5.3.3 CDR Time domain issues

An additional consideration, which must be taken into account with re-timers, is knowing which clocking (data rate) domain drives each independent channel section. The CDR of a re-timer’s receiving port should use the recovered clock of the first (pre-re-timer) section of the full link to drive the re-timer’s output transmission path to the other end point (the post-re-timer link). This ensures that there are no buffer overflow or data rate mismatches. This can have an impact on the link up flow of the device, and is up to the re-timer designers to verify that systems, in real world applications, will not have issues regarding the various clock domains. Again, interoperability testing is critical to uncover potential issues such as this.

16. About the Authors

Samie Samaan is a principal engineer at Intel Corporation, where he joined in '94. He is currently in the CPU design team in Oregon, working on the power modeling of a next generation of CPUs, including power-performance optimization to manage process variations. Recently, he spent 4 years at the Data Center Group (DCG), leading all platform-level modeling of I/O channels (PCB, connectors, sockets, etc.) employing behavioral models of PHYs, for various high-speed SerDes and single-ended busses. He also oversaw the creation of customer design guides addressing all I/Os of that upcoming platform. He pioneered and led investigations of analog SerDes Re-driver characteristics, and their effects on high-speed SerDes bus behavior and specifications. In that role, he created several reports, test methods, and helped set internal guidelines. Prior to DCG, he spent 15 years in CPU design, working on high-performance custom VLSI circuit and logic design. He documented Intel's GTL+ bus electrical Specs, then designed multi-GHz high-performance self-timed dynamic logic adder circuits for CPUs. He developed expertise in process variation statistics, and subsequently invented a ubiquitous intra-die process variations monitor, which became a staple of process monitoring. He co-designed a Soft Error Rate (SER) monitor IC, and performed numerous process variation analyses on CPUs. Later, he designed several Analog blocks in an industry-first integrated voltage regulator for high-performance CPUs, including a control loop compensator, a DAC, and a real-time very high-speed intra-die voltage monitor.

Prior to Intel, Samie was VP of Engineering at Zeelan Technology, which specialized in Behavioral I/O buffer simulation models for high-speed board design. He was briefly at Epson working on a Pentium chipset project. He has also worked at the Grass Valley Group, in CA, on an ASIC design for Video Compression.

Samie spent 8 years at Tektronix, working on multi-GHz Si & GaAs large-signal linear ICs. At Tektronix, he also developed a numerical code to compute electron deflection in electrodynamic fields, used in GHz-CRT studies.

Samie obtained his Master's in Electromagnetics & Microwaves from the University of Mississippi. He had also obtained a post-graduate "Advanced Studies Diploma" in Medical Electronics from Nancy-1 University in France. His later education includes several graduate-level courses in Semiconductor Physics.

Dan Froelich is a principal engineer in the I/O Technology and Standards group. His expertise covers a broad span in I/O including electrical, protocols, software, form-factor, and compliance. Dan chairs three WGs (Electrical (acting), CEM, and compliance) in PCI-SIG, and is among the few people responsible for the broad adoption of PCI Express in the industry. Dan works closely with the internal teams, as well as the ecosystem, to ensure that PCI Express continues to work first and best with IA through its evolution. Dan has defined and evolved PIPE, the core PHY IP SoC interface for PCI Express, USB, and SATA, which has been universally adopted inside Intel as well as across the industry at large.

Sam Johnson is a Network Hardware Engineer in Intel's Networking Division, supporting server-class Ethernet controllers and high-speed communication technologies. Sam started at Intel in 2010, and quickly became the primary debugger of high-speed serial Ethernet protocols for Intel's Networking

Division. He is an expert in IEEE and industry standards for backplane and cabled Ethernet technologies, including involvement with new and future protocols. He has also worked extensively with a wide variety of external PHY devices; including enabling development of the first KR to KR re-timers and conducting industry-leading cross vendor evaluations of re-drivers for 10G-KR applications. Sam works closely with both customers and design teams to connect future device capabilities to customer needs, and ensure that new devices will support the latest debug tools. Sam currently conducts regular debug training classes, educating and mentoring engineers, from around the world, in Ethernet technologies and debug methodologies.

References

[1] W. C. Johnson, "Transmission lines and networks," McGraw-Hill, Technology & Engineering, Jan 1, 1950.

[2] C. Belfiore and J. J. Park, "Decision feedback equalization," *Proceedings of the IEEE,* vol. 67, no. 8, pp. 1143 - 1156, Aug 1979.

[3] I. Gerst and J. Diamond, "The Elimination of Intersymbol Interference by Input Signal Shaping," *Proceedings of the IRE,* pp. 49 , Issue: 7, 1961.

[4] W. Dally and J. Poulton, "Transmitter equalization for 4-Gbps signaling," *Micro, IEEE,* p. 48 – 56, Volume: 49 , Issue: 7 1997.

[5] B. R. Saltzberg, "Timing recovery for synchronous binary data transmission," *The Bell System Technical Journal,* vol. 46, no. 3, pp. Page(s): 593 - 622, 1967.

[6] M.-t. Hsieh and G. Sobelman, "Architectures for multi-gigabit wire-linked clock and data recovery," *Circuits and Systems Magazine, IEEE,* vol. 8, no. 4, pp. 45 - 57, 2008.

[7] R. E. Best, Phase-Locked Loops, McGraw Hill Professional, Jun 20, 2003.

[8] C. Lee, M. Mustaffa and K. Chan, "Comparison of receiver equalization using first-order and second-order Continuous-Time Linear Equalizer in 45 nm process technology," in *4th International Conference on Intelligent and Advanced Systems (ICIAS), Volume: 2, Page(s): 795 – 800.*, 2012.

[9] "Universal Serial Bus 3.0 Specification," June 6, 2011.

[10] "PCI Express® Base Specification, Revision 3.0," PCI-SIG, November 10, 2010.

[11] Universal Serial Bus 3.0 Specification Revision 1.0, USB-IF, June 6, 2011.

[12] "I2C-bus Specification and User Manual, Rev. 6 — 4," NXP, April 2014.

[13] "System Management Bus (SMBus) Specification, Version 2," SBS Implementers Forum, Aug. 3, 2000.

[14] H. L. H. Stephen H. Hall, Advanced Signal Integrity for High-Speed Digital Designs, Wiley, 2009.

[15] ASHRAE Technical Committee, "2011 Thermal Guidelines for Data Processing Environments – Expanded Data Center Classes and Usage Guidance," http://ecoinfo.cnrs.fr/IMG/pdf/ashrae_2011_thermal_guidelines_data_center.pdf, 2011.

[16] J. K. R. a. B. G. Loyer, "Humidity and Temperature Effects on PCB Insertion Loss," in *DesignCon*,

2013.

[17] P. B. G. Hamilton and G. a. S. J. Barnes Jr., "Humidity-Dependent Loss IN PCB Substrates," *Printed Circuit Design & Manufacture,* vol. 24, no. 6, p. 30, 2007.

[18] "Serial ATA Revision 3.2 (Gold Revision)," Serial ATA International Organization, 7 August 2013.

[19] "PCI Express® Card Electromechanical Specification, Revision 3.0, ver. 1.0," PCI SIG, March 6, 201.

[20] PCI-SIG, Extension Devices ECN, Oct. 6, 2014.

[21] I. S. 802.3, "Standard for Ethernet," IEEE, 2012.

[22] IBIS Open Forum, http://www.vhdl.org/ibis/.

[23] "Standard 802.3, clause 72, “Physical Medium Dependent Sublayer and Baseband Medium, Type 10GBASE-KR," IEEE, 2008.

[24] "Specification for SFP+," SFF Committee , Rev 4.1 July 6, 2009, Rev 4.1 Addendum September 15, 2013.