Trusted Debugging with Compliant PHYs

Testing and research conducted by Total Phase Engineering using industry-standard benchmarks and practices.

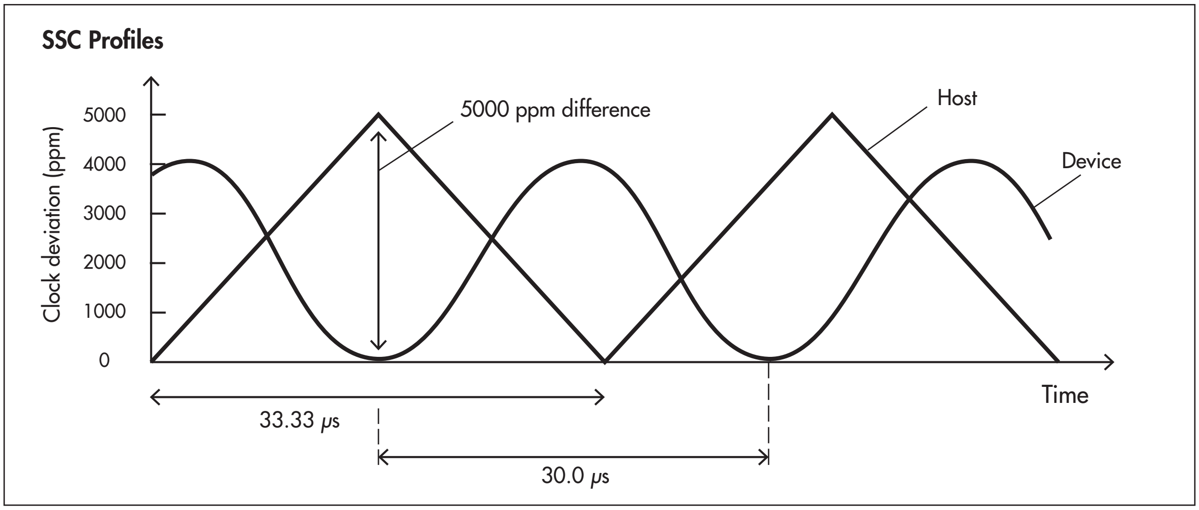
INTRODUCTION

PHYs are a crucial part of all SuperSpeed USB analyzers as they perform the critical role of decoding and deserializing the analog signals. These solutions typically fall into two categories: USB 3.0 compliant PIPE-based solutions, or non-compliant, generic deserializers.

COMPLIANCE

There are numerous benefits to using a compliant USB 3.0 PHY in an analyzer. First and foremost is the guarantee that the analyzer will be able to properly receive all SuperSpeed traffic on the bus. By using a test tool with a compliant PHY, the developer can capture data without any concern for loss of data. Spread-spectrum clocking profiles, jitter tolerances, receiver thresholds, 8b/10b encoding, and link recovery are all verified to work within the requirements of the USB 3.0 specification. Non- certified or non-compliant PHYs, as commonly found in an FPGA's general purpose deserializer, may be able to partially accommodate these tolerances, but there is no guarantee that all requirements of the specification have been met.

The market can be quite confusing in this regard. While compliant PHYs clearly state their adherence to the USB 3.0 standard, non-compliant devices can sometimes inaccurately give an impression of conformance. For example, FPGAs typically have generic deserializers which are capable of transmitting and receiving at the necessary bit rates for SuperSpeed USB. Although some of these deserializers may even be certified for SATA or PCI-e, this is still not enough to meet the rigid standards of USB 3.0. In fact, some reference designs for SuperSpeed device controllers on FPGAs use USB 3.0 compliant external PHYs, such as the Texas Instruments TUSB1310A.



Shown above are two USB 3.0 spread-spectrum profiles. Although both compliant, they share few similarities; they have different modulation rates, different frequency deviations, and completely different shapes. Triangular profiles may be the norm, but they are not guaranteed by the specification. These kinds of differences in SSC profiles can occur between a transmitter and receiver, and the receiver must be robust to these differences. Only USB 3.0 compliant PHYs can guarantee proper reception of these kinds of signals.

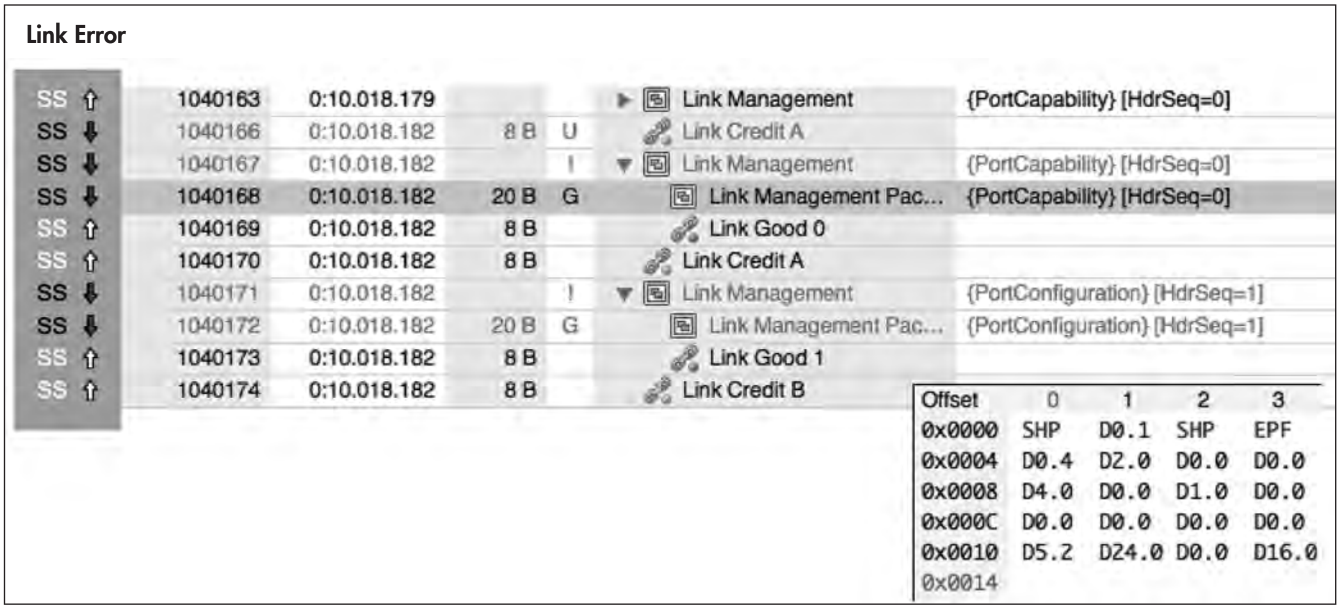
As a simple example of the differences between PCI-e and USB 3.0, consider how the reference clock is handled in the two protocols, especially in the presence of Spread-Spectrum Clocking (SSC). Both protocols share similar requirements on clock frequency accuracy, SSC modulation rate, and SSC modulation amplitude, but differ in how the clock is distributed. In PCI-e the reference clock is ordinarily forwarded from the Root Complex to the various PCI-e peripherals so as to share the same SSC profile and thereby satisfy the requirement that each link partner transmit within a bit rate of +/- 300 ppm of each other. In USB 3.0, the host and device have their own, unique, reference clocks, and completely independent SSC profiles. It is left to the receiver to track and recover the clock as part of its data recovery circuitry-a clock which can differ by as much as 5000 ppm due to SSC differences alone. Compliant USB 3.0 PHYs guarantee that they can recover data from all allowable SSC profiles, even without a forwarded reference clock from the transmitter. Analyzers which use non-compliant PHYs can make no such guarantee, and developers may find those analyzers incapable of capturing data from certain hosts or devices.

The differentiation between compliant and non-compliant PHYs is perhaps most noticeable when trying to analyze devices that are coming out of a lower power state. USB 3.0 devices make heavy use of low power states to conserve battery on portable devices, and are constantly moving in and out of these states very quickly. To allow for these quick transitions, the USB 3.0 specification only requires a very short training sequence handshake which enables each receiver to lock to the respective transmitter’s signal. Analyzers using non-compliant PHY technology may not be able lock as quickly as the link participants (i.e., the host and device), and can often fall by the wayside during these transitions, making further analysis after a low power state troublesome. These analyzers can even make it seem as if the bus is not operating properly, when in fact it is the analyzer which is malfunctioning. While an analyzer using compliant PHY technology may have the occasional issue with obtaining lock, it will usually fare better than a generic deserializer.

PIPE

As of this writing, the only commercially available fully-compliant USB 3.0 PHY for generic use is the TUSB1310A from Texas Instruments. This device provides the industry-standard PIPE interface for link layer communication. Although the industry agrees that PIPE interfaces are fully capable of reporting errors in the data payload, there can be some confusion as to its capabilities at handling lower level issues, such as the ability to interpret the 10b symbol, packet framing, decode, and disparity errors.

It would be easy to assume that because the PIPE interface does not transmit the raw 10b code to the link controller, that one would not be able to capture the actual 8b/10b symbol that was transmitted on the bus. This misconception is clarified when developers understand that despite the name “8b/10b”, the PIPE link layer interface is actually using 9 bits so that it can specify whether a D or K symbol is transmitted or received. Analyzers using a PIPE interface can easily, and accurately, indicate which symbol was transmitted, even if the raw 10b code is not available. While it is true that the disparity information per symbol is discarded, any decode and disparity errors are clearly indicated by the PIPE interface through a secondary channel, known as RxStatus. Once the presence of disparity errors is identified by the protocol analyzer, the root cause can fall into two classic categories: (1) random disparity errors that are indicative of signal integrity issues, for which a high-end oscilloscope would be the next stage of debugging; or (2) repeatable disparity errors that would be tackled via hardware simulation of the design.



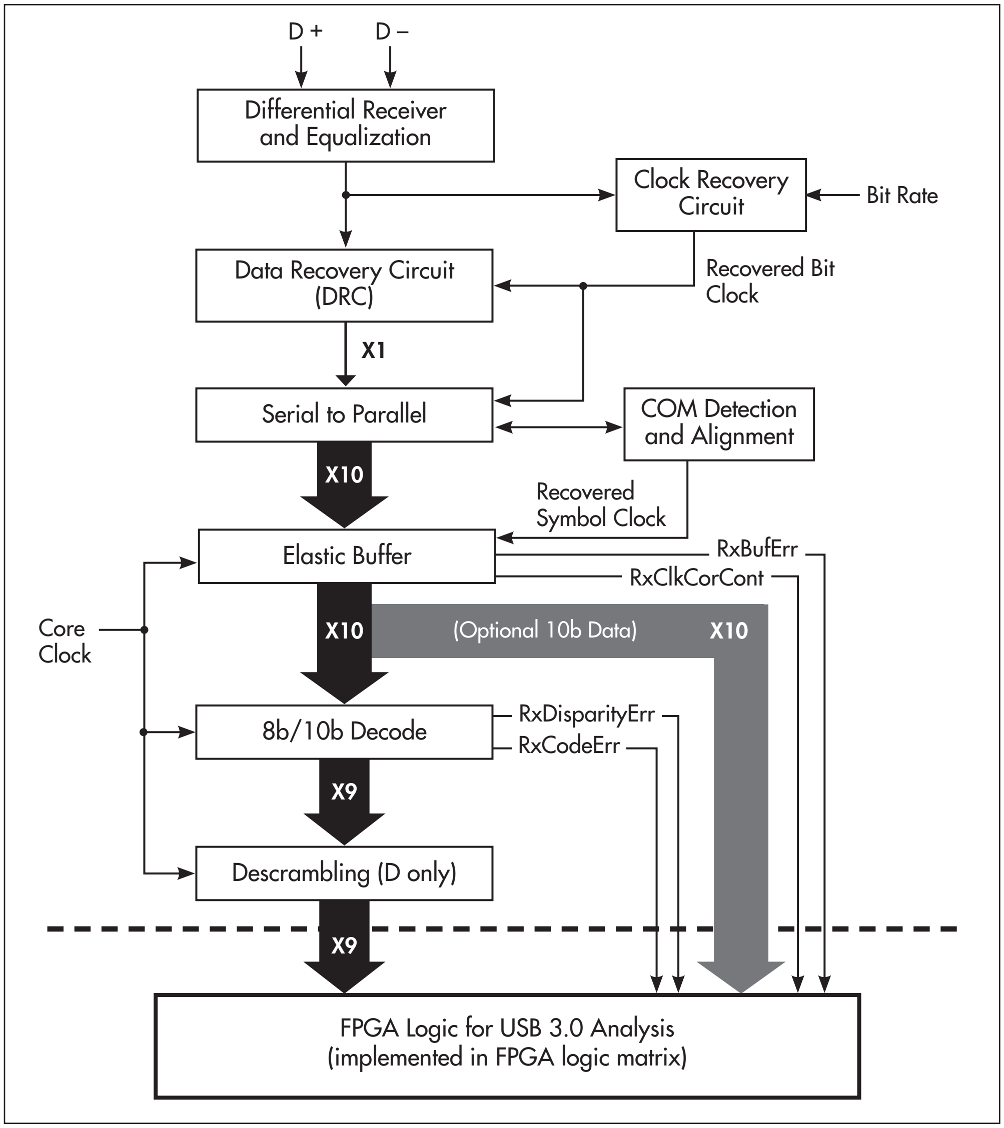
Link Error

```
SS ⇧  1040163  0:10.018.179              ▶ Link Management           {PortCapability} [HdrSeq=0]
SS ⇩  1040166  0:10.018.182   8 B   U      Link Credit A
SS ⇩  1040167  0:10.018.182         !    ▼ Link Management           {PortCapability} [HdrSeq=0]
SS ⇩  1040168  0:10.018.182  20 B   G        Link Management Pac...  {PortCapability} [HdrSeq=0]
SS ⇧  1040169  0:10.018.182   8 B            Link Good 0
SS ⇧  1040170  0:10.018.182   8 B          Link Credit A
SS ⇩  1040171  0:10.018.182         !    ▼ Link Management           {PortConfiguration} [HdrSeq=1]
SS ⇩  1040172  0:10.018.182  20 B   G        Link Management Pac...  {PortConfiguration} [HdrSeq=1]
SS ⇧  1040173  0:10.018.182   8 B            Link Good 1
SS ⇧  1040174  0:10.018.182   8 B          Link Credit B
```

```
Offset   0     1     2     3
0x0000   SHP   D0.1  SHP   EPF
0x0004   D0.4  D2.0  D0.0  D0.0
0x0008   D4.0  D0.0  D1.0  D0.0
0x000C   D0.0  D0.0  D0.0  D0.0
0x0010   D5.2  D24.0 D0.0  D16.0
0x0014
```

The view above is from a capture of the USB-IF Link Layer 7.5 test case taken with an analyzer using the TUSB1310A PIPE-based PHY. The 7.5 test intentionally corrupts the header packet framing to test the receivers robustness to these situations. In this particular test, a 10b symbol of D0.1 was transmitted instead of the correct SHP symbol (K27.7). As shown, the analyzer is fully capable of clearly marking the error that occurred as well as indicating the actual symbol that was transmitted on the bus.

Whether the solution is a discrete PHY or a generic deserializer (FPGA), the same general architecture is employed.



The timing of symbols, the ability to detect errors, and the number of SKP symbols reported by the analyzer are consistent across solutions. Additionally, a discrete PHY preserves the 8b/10b symbol; while simultaneously offering significant performance advantages such as superior spread-spectrum clocking profiles, lower jitter, and better link recovery.

CONCLUSION

USB 3.0 compliant PHYs are the only PHYs verified to properly work across all SuperSpeed signaling patterns-an especially important trait in a protocol analyzer. The TUSB1310A is currently the market's only option for such a tool, and maintains low-level debugging features.

Total Phase's award-winning Beagle™ USB 5000 SuperSpeed Analyzer uses the fully-compliant Texas Instruments TUSB1310A as its PHY solution, thus guaranteeing developers conformance to the SuperSpeed specification. This Analyzer is capable of handling the full range of USB 3.0 transceiver requirements while maintaining the ability to indicate low-level errors on the bus. When it comes to analyzing SuperSpeed issues of any sort, developers can put their trust in the Beagle USB 5000 analyzer's dependable and reliable technology.