

Trailblazing SuperSpeed USB Design and Verification

USB's eagerly awaited 3.0 revision offers ten times the speed of USB 2.0, while maintaining backwards compatibility with USB 2.0 and USB 1.1 devices. Early adopters ramping up their USB 3.0 developments are testing the next-generation peripheral interconnect (known as SuperSpeed USB) that offers 5 Gbit data rates over copper interconnect. Developers are grappling with sophisticated physical layer features, including high-speed signaling, dynamic equalization, and power management, that add complexity to both the development and the test effort.

USB 3.0, like other high-speed serial protocols, introduces challenges at both the physical and protocol layers. The protocol component takes the greater part of the specification and adds rigid power management states that are mandatory for SuperSpeed certification. No longer is it just a case of whizzing an electron down a cable at the right time. The precise way the electron is sent and received must be negotiated before any useful work can be done. The option of relying on good fortune in the development of high-speed serial bus is decreasing as quickly as data-rates and protocol complexity are increasing.

SuperSpeed USB shares many physical layer characteristics with PCI Express®

2.0, including 5 Gbps signaling, 8b/10b encoding carrying an embedded clock and support for active power states. However, unlike PCI-Express 2.0, USB 3.0 is targeted at external device interconnects.

Mechanically, the USB 3.0 connector has been designed with backward compatibility to the USB 2.0/USB 1.1 connector. However, attenuation on the 5 GHz serial lines is considerable and receivers operate on very small margins. Support for wired cable lengths up to 3 m present additional challenges at these speeds that require techniques that differentiate it electrically from PCI-Express.

Similarities can also be drawn between USB 3.0 and Serial Attached SCSI 6G. These interfaces offer comparable data rates and cable lengths. While the point-to-point link handling is substantially different, SuperSpeed USB has been designed with an emphasis on storage applications. Planned enhancements to the USB 3.0 mass storage driver stack will add multiplexed data streams to boost throughput.

A State of Independence

USB 3.0, like its predecessors, is designed to work asynchronously over a differential pair. While higher layer transfers remain primarily host driven events, both host and device are dependent on the use of state machines to keep track of link status. Beyond the simple request acknowledgement sequence, USB 3.0 maintains logical

state machines for everything from power management, through stream protocols, hub management to error recovery. The power management state machine even gets its own appendix to further detail its operation. In total, the SuperSpeed spec adds around 20 new state machines to the USB control interface. USB 3.0 devices must track entry and exit from all logical states at the link layer while simultaneously managing flow control buffers and packet framing. Pre-silicon simulation provides one method for getting early test coverage for complex IP designs. Yet post silicon testing using real world devices remains an essential step on the road to production release.

Debugging state machines can be simplified with the addition of debug code or stubs that can expose the current state in some manner. Developers electing to work with third party IP libraries may not have this option.

One solution is use a protocol analyzer between host and device that can unobtrusively monitor and track the state changes of all the state machines as messages pass from one end to the other. This requires no debug stubs or modification of the base IP at either end of the link. Monitoring state changes in this manner adds no delays as event timings will be the same as the production device.

Timing is everything

As communication speeds increase a number of time related challenges present themselves. This may be intuitive, but one of the major advantages of migration to asynchronous serial from synchronous parallel is the time of flight for an edge is not as important. Therefore, the time it takes a signal to propagate down the wire shouldn't matter. After all, the clock is embedded in the data, so it will arrive at the other end together. Therefore, instead of using a massively redundant overhead to protect against occasional errors (USB 3.0 expects bit error rate to be less than 10^{12} bits) a timeout is usually employed to cover the cases where a critical or double failure cannot be completely avoided. Consequently, the developer needs to have a method to identify the root cause of an unexpected state change (eg timeout causing a recovery).

Protocol analyzers are designed to capture and display an exact copy of the two-way communication between a host and device. By presenting complex data exchanges in an easy-to-digest format, analyzers can be used to verify all 200+ timeout references on each transaction considerably quicker than a human could with an oscilloscope or logic analyzer trace.

Link Synchronization

Establishing a link between a SuperSpeed host and device operating at 5 GHz requires both receivers on the

DUTs to extract clock and phase timing by locking to the electrical transitions as quickly as possible. USB 3.0 devices achieve this by repeating a series of special link training symbols that enable PHY synchronization. The USB 3.0 specification allows for variations in the receiver's abilities as no two PHYs will lock at exactly the same rate.

USB 3.0 developers face a fundamental challenge when introducing protocol analyzers into these high-speed link synchronization sequences. To effectively debug link up issues, the analysis system should capture and show precise timing information for each of the link training state transitions. This requires that the analyzer serializer / deserializer (SERDES) detects the electrical idle state and then achieve bit lock as quickly as possible after the DUTs enter the RX_EQ state. Any delay in lock on the signal can cause the analyzer to miss the synchronization "window". If the analyzer is unable to synchronize with the TX / RX pair, the system will erroneously show "garbage" in the trace.

PHYs: Chicken or Egg Dilemma

It's become quite common for new serial technologies to leverage existing technologies to reduce development time and risk. Similarities between USB 3.0 and PCI Express 2.0 have allowed some early developers to use PCI-Express PHYs to begin prototype

testing. Although the data rate and symbol coding scheme is similar, the USB 3.0 specification has some specific enhancements for external device applications and special out of band signaling methods to maximize power savings.

In some cases, vendors will develop test chips that can be used on development platforms prior to the vendor integrating the analog block and digital block into a single ASIC. These early stage discrete PHYs allow physical layer teams to begin characterization well ahead of the production SOC availability. Using early test PHYs involves some risk that these prototypes contain bugs or may be missing the full functionality of a production SOC.

Still another alternative is to use programmable SERDES, configured for PCI Express 2.0 physical layer characteristics. While this approach enables testing between early device prototypes, subtle differences in implementation can require careful tuning to the front end. It's also possible that devices from different vendors will not reliably operate together. This method is often used to allow division of labor in a design team such that the digital group can begin internal testing without waiting for 3.0 PHYs to become available.

Development of analyzers and testers for new technologies like SuperSpeed

USB is also aggravated by the scarcity of PHYs. The developers of protocol analyzers are under considerable pressure to provide test equipment as soon as the first chip vendors reach power-on. The analyzers themselves frequently incorporate actual PHYs in their design. Yet there are limited PHY options available during early development. Production silicon can lag the test market by as much as 12-18 months. Once vendors are confident enough in their design to begin sampling components,

In some cases, test PHYs will be incorporated in the design of the analysis equipment. This approach also runs some risk that early stage development PHYs may not be functionally complete. Interoperability problems in the analyzer may surface that manifest themselves as link synchronization issues when testing with production silicon. Fortunately, for the USB 3.0 community, alternate analog front-end (AFE) probing schemes for 5 Gbps signaling have been developed that reduce the reliance on prototype PHYs. Teledyne LeCroy's expertise using programmable SERDES for 5 Gbps PCI Express allowed it to provide reliable test tools well ahead of first USB 3.0 silicon availability. Fine grained controls are provided for these programmable PHYs allowing the Teledyne LeCroy analyzers to adapt to a variety of test setups.

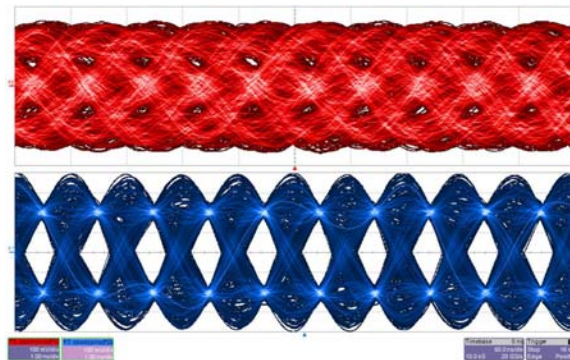


Figure 1: Before and after Receiver equalization

Emphasis on Equalization

It's anticipated that most USB 3.0 devices will use dynamic receiver equalization to overcome signal loss common when operating at 5 GHz frequencies. For SuperSpeed devices, the equalization is adaptive allowing devices to calibrate the receiver for different cable lengths. PHYs will accomplish this dynamic equalization by cycling through special "spectrally rich data patterns" during link training. Circuitry on the SuperSpeed device can adjust the receiver eye to minimize the effects of dielectric loss and cross-talk (Figure 1). The SERDES on the analyzer must also provide some capability to equalize 5Gbps signaling to ensure link synchronization. Analysis tools based on programmable SERDES have an advantage here as they provide options for tuning pre-emphasis and differential voltage - among other settings - to ensure signal fidelity that ideally should exceed that found on the system under test.

Power Management

The USB 3.0 specification defines aggressive power management strategies to extend battery life, reduce power consumption and provide responsive devices. When a SuperSpeed device “wakes up” and turns its transmitters on to exit electrical idle (U1 transition to U0), synchronization must be re-established. If the upstream port does not know that a device is going to re-connect it is unlikely to successfully meet the USB 3.0 timing constraints.

Therefore an additional mechanism is provided for waking up the port from the quiescent state. The device issues a Low Frequency Periodic Signaling (LFPS) handshake to alert the upstream port and then moves through the required Recovery and Link Training states. The USB 3.0 specification currently defines a rigid exit latency of ~1us moving from U1 to U0 (operational) power states. As with the initial link synchronization, the analyzer front-end must also detect and capture each state during these frequent *recovery* sequences that are required when transitioning from power save mode. Any delay in lock when exiting electrical idle outside this window will again cause the non optimized analyzer to be unable to re-synchronize with the link-under-test.

FPGA-based prototyping

Higher levels of integration and complexity in chip design has led developers to first prototype complex

designs using software emulators or FPGA-based development platforms. While most USB devices will ultimately be implemented as an SOC, these prototyping environments allow large portions of the digital logic or IP to be developed and tested at reduced speed. For bleeding edge designs that outpace what can be implemented in a modern FPGA, there is value in prototyping the design at reduced speed to verify functionality (half rate, quarter rate, etc...). The ability to capture and analyze SuperSpeed packets transmitted at user defined clock frequencies allows designers to verify MAC layer logic prior to committing a design to silicon.

Validating error detection and recovery

USB 2.0 used CRC5 and CRC16 checksum algorithms to verify data integrity at the packet layer. USB 3.0 adds a third 32-bit CRC because of the larger supported data payloads. However the polynomial used for the CRC-5 is the same between USB 2.0 and USB 3.0 the polynomials used for CRC-16 and CRC-32 checksums are new.

This means that cyclic redundancy check algorithms and circuits used in USB 2.0 cannot be directly re-employed for USB3.0. As with any new technology, there is a risk of mis-reading the specification or not reaching the same consensus as the wider community. A common problem in USB 1.1 was the failure of some developers

to acknowledge that the CRC did not follow the LSB rule defined near the start of the specification. As a result, the CRC was sent in reverse order to the requirement defined later in the specification, causing a situation in which devices could interoperate between themselves but failed when connected to devices from different vendors.

Protocol aware tools provide a third party interpretation of the specification to validate device behavior. Not only will they expose real bit errors, but also can reveal systemic mis-interpretations of the specification.

Of course, the CRC allows the device to detect and retry frames that contain bit errors. But verifying whether the IP correctly recovers from CRC and other errors at the link layer is difficult to simulate using real devices. While an arbitrary waveform generator could be programmed to transmit some errors, the complexity of generating logical state machine errors make this a prohibitive task.

It has become standard practice to use protocol-aware exercisers for injecting errors and validating link recovery behavior. An exerciser system will interface directly to the DUT over standard cabling and emulate real host or target behavior. To be effective, these exercisers must have the ability to establish link synchronization and direct the device to a specific logical state prior to injecting the error.

Users can author controlled test scenarios using a script based higher-level language. But with USB 3.0, a nearly continuous stream of link layer handshakes, (Idles, Skips, Header ACKs, etc...) makes it painstakingly difficult to construct device emulation behaviors using packet level scripting. Fortunately, a new generation of protocol-aware exercisers now automatically handles the low latency handshakes to dramatically simplify test script development. These exercisers feature a complete link layer implementation that intelligently responds to logical state changes. These systems allow early adopters to start bring-up testing with USB 3.0 chips well before commercial hosts are available.

The more common application for exercisers includes simulating simple bit errors by corrupting the CRC. With



intelligent exercisers that can progress through multiple logical states, one can go further and test violations such as corrupting flow control or other link commands. Sending LCRD_A/B/D instead LCRD_A/B/C/D and similar

packet ordering errors should send the link into recovery. Exercisers should also allow users to easily adjust the timing and frequency of errors to find boundary conditions that might not turn up during simulation. With 20 new state machines and the corresponding sub-state transitions a comprehensive test plan USB 3.0 becomes insurmountable without a protocol-aware exerciser system.

Another concern with testing USB 3.0 is the high data rate and its impact on memory management within the analysis equipment. During data transfers, SuperSpeed links can flood a typical 1-gigabyte memory space in less than two seconds. Even during link synchronization, a nearly continuous stream of idle and flow control symbols can rapidly consume available capture memory. Event triggering, considered a luxury in USB 2.0, becomes essential with USB 3.0. Snapshot recording or spooling techniques are not practical at 5 Gbps. Users need the ability to trigger on specific bus conditions or symbols to isolate events of interest. As debug efforts focus on intermittent issues, more sophisticated techniques including triggering on sequential events or individual header fields are needed.

Industry Ecosystem

The similarities between SuperSpeed and PCI Express 2.0 has allowed both silicon developers and test vendors to jump start their USB 3.0 development. With PCI Express 2.0 IP and expertise under their belt, several silicon design

houses are expected to begin sampling USB 3.0 chipsets in early 2009.

Likewise, both electrical and protocol layer test vendors have leveraged their PCI Express 2.0 techniques to deliver stable tools well ahead of the mainstream market. Nowhere is this more evident than in the critical signal locking performance of these early analyzers. Custom circuitry leveraged from Teledyne LeCroy's PCI Express 2.0 analyzer has been adapted to the Teledyne LeCroy Voyager USB 3.0 tester to provide impressive signal fidelity. This allows the analyzer to sit in the data path and seamlessly recover from electrical idle and capture the link training sequence. All training parameters including timing elements are reported in the trace as are individual bus and power state transitions.

Teledyne LeCroy's SuperSpeed USB analyzer – exerciser system has already been deployed within many early development sites giving the industry a head start in the validation of USB 3.0 chipsets. The stability and the rapid pace that these development ecosystem tools have evolved suggests that USB 3.0 technology introduction will continue at SuperSpeed.

More information

The USB 3.0 specification may be obtained from the USB Sig website. Teledyne LeCroy Protocol aware test tools can be found at <http://www.teledynelecroy.com>

USB 3.0 viewer software is also available from Teledyne LeCroy via email: contact@lecroy.com

About the authors:

Mike Micheletti is the senior product marketing manager at LeCroy with over 10 years of experience in high speed serial protocol testing and is a leading member of Teledyne LeCroy's staff supporting the USB Sig. Matthew Dunn is part of the Teledyne LeCroy Protocol Group technical support staff with over 8 years of experience in USB design and development.