USB 2.0: Compliance and beyond

Martin Rowe - October 01, 2002

The Universal Serial Bus (USB), introduced to computers in 1995, now appears in all PCs and in many peripherals. Yet, the bus's 12-Mbit/s speed has limited its usefulness for products such as video cameras and DVD players. Some manufacturers have tried to meet high-speed requirements by employing the IEEE 1394 serial bus, but the USB 2.0 bus, with its 480-Mbits/s rate, promises to become the dominant PC-to-peripheral interface. In 12-to-18 months, all PCs will come with USB 2.0
ports (Ref. 1).

With the release of the USB 2.0 specification in April 2000, the USB's speed increased 40x, making it fast enough for most PC peripherals (Ref. 2). To ensure that USB 2.0 host controllers, hubs, and peripherals (called "devices" in the USB specification) communicate with each other, the USB Implementer's Forum (USB-IF, www.usb.org) developed test specifications, which complement the USB 2.0 specification by providing uniform test methods that anyone can apply.

Before you can apply a USB logo to your product, it must pass compliance tests. The USB-IF holds compliance workshops—also called "plugfests"—where members can, at no charge, bring their products for compliance testing. Plugfests take place four times a year, usually in Silicon Valley, and once a year in Asia (see www.usb.org for dates and locations). Between plugfests, you can take your product to a recognized third-party test lab (Ref. 3).

You can't self-certify your product for USB compliance, but you can perform precompliance tests in your lab that will increase your confidence of passing a compliance test. A USB 2.0-compliance product must also operate at low-speed (1.5 Mbits/s) and full-speed (12 Mbits/s) data rates in addition to operating at the high-speed (480 Mbits/s) data rate. In this article, I'll focus on testing a USB device in high-speed mode only.

**Physical changes**

To run USB 2.0 precompliance tests, you need a digital oscilloscope with two single-ended voltage probes, a differential voltage probe, and a current probe. USB 2.0 ports send data using a differential-mode voltage on the bus's two data lines, called D+ and D-, which requires a differential probe. (You need the single-ended probes to measure the data lines while the device is in low-speed or full-speed mode.) With the current probe, you measure the current on the bus's power supply lines (V\textsubscript{BUS} and GND) when you connect a device to the bus. You need not measure current when you remove a device from the bus.

To test a USB 2.0 peripheral device, you'll need a "test bed" computer that runs Windows 2000 or Windows XP. (The software you need won't run on any version of Windows 98.) You also need a USB 2.0 host controller. To add a USB 2.0 host controller to your PC, you'll need a USB 2.0 host-controller card for the PCI bus, available from several makers and sold through retail outlets. You also need an IEEE 488 interface to control your test equipment and retrieve test data.

Your test bed needs software to run the tests and analyze the results. Fortunately, you don't have to write your own software for either task. You can download software that runs the tests from the USB-IF Web site (Ref. 4). The download gives you an executable application to run the test bed, and it provides a set of Matlab scripts that analyze the data and produce reports. (You will need a copy of Matlab to run the scripts.) Using the software download guarantees that you conduct your tests using the same software used in compliance tests.

Some scopes can perform the data analysis for you. The Infinium line from Agilent Technologies has an option that adds the Matlab scripts to the scope (in this case, you don't need a copy of Matlab on your test-bed PC). The Tektronix TDS 7000 series scopes use proprietary scripts to perform the tests, which also eliminates Matlab. The USB-IF has approved both companies' scopes for compliance testing and has published test procedures for each scope (Ref. 5).

LeCroy's Wavemaster and Gould's Ultima 500 scopes let you install Matlab and the USB 2.0 test scripts. Each also runs Windows 2000, so both can double as a test bed if you install a USB 2.0 host...
controller card (neither scope has a built-in USB port that complies with USB 2.0). The USB-IF has yet to approve the LeCroy and Gould scopes for compliance testing, but you can use them for precompliance tests.

**Test fixtures**

To get test signals into and out of your UUT, you need a set of USB 2.0 test fixtures. Each fixture works with a specific test. For example, use one fixture for voltage measurements on the USB data lines and another for measuring current on the power lines.

You can either build or buy the test fixtures. To build a test fixture such as the one in Figure 1, download the files from Intel or from the USB-IF Web site (Ref. 6). You'll get complete plans for fabricating and assembling the fixtures. One board design handles several test fixtures. You just populate the board with the components you need for transmitter signal-quality tests, receiver-sensitivity tests, or DC tests (inrush current and backdrive voltage).

If you prefer, you can buy test fixtures from Agilent Technologies, Tektronix, or FuturePlus Systems. The Tektronix fixture combines all the test fixtures into one assembly, while the FuturePlus test fixture lets you connect an Agilent logic analyzer to the bus. You don't need a logic analyzer for compliance tests, but it can help you debug your product.

You need to make all measurements with your test points as close to the DUT as possible. Note the short USB cable on the test board in Figure 1. The cable's USB connector plugs into the UUT, and a USB cable from a host controller or hub plugs into the connector on the right side of the fixture. The fixture will contain pogo pins for performing signal measurements and SMA connectors for connecting a data generator. With access to the D+ and D- data lines through the pogo pins, you can perform high-speed physical-layer tests such as checking the handshaking sequence and making signal-quality measurements.

To perform high-speed signal-quality measurements, you must first get the DUT to operate in high-speed mode. USB 2.0 devices power up in full-speed mode, then go through a handshaking sequence with the host controller before they enter high-speed mode. You need to measure voltage levels and
signal timing during the handshaking sequence.

**Figure 2** shows the handshaking sequence by which a USB 2.0 device identifies itself as high-speed capable. During the sequence, the USB 2.0 host and peripheral identify each other by sending a series of J and K bits where J and K refer to non-return-to-zero inverted (NRZI) logic 1's and logic 0's, respectively. The relative polarity of the signals represents a 1 bit or a 0 bit. To generate a differential logic-1, a USB 2.0 product pumps current through the D+ line. A lack of current in the D– line forces it to pull toward 0 V. For a differential logic-0, the device pumps current through the D– line instead. The current forces a voltage at the other end of the cable, formed by the impedance of the cable and termination resistors on a host, hub, or device.

![Figure 2](image.png)

**Figure 2** A chirp signal from a USB 2.0 peripheral device tells the host controller that the device is capable of transmitting in high-speed mode. *Courtesy of Agilent Technologies.*

When a host controller detects a new USB device, it resets the bus (blue trace goes low). After some latency time (between 2.5 µs and 3.0 ms), the peripheral device sends a "chirp K signal" (yellow trace goes high) to the host. That chirp lasts between 1 ms and 7 ms. A USB 2.0 host or hub responds with a KJKJKJ series of pulses. When the peripheral device receives the pulses, it switches to high-speed mode by changing the termination resistors on its D+ and D– data lines. In high-speed mode, the device disconnects its 15–k pull-up resistor and terminates the D+ and D– data lines with 45–Ù resistors to ground. The high-speed termination resistors result in the reduced amplitude of the pulses in Figure 3.

As part of your USB 2.0 precompliance tests, you must measure the USB device latency time and chirp duration with the oscilloscope and the Matlab scripts. A compliance test will verify that your device meets these specifications.

**It's in the eye**

Because of their high bit rate, USB 2.0 devices require signal-integrity not required for low-speed or high-speed devices. For signal-integrity tests on high-speed signals, you need an eye diagram. The diagram lets you measure a signal's rise time, fall time, undershoot, overshoot, and jitter. The eye diagram will reveal problems in terminations of USB data lines (Ref. 7).
Figure 3 An eye diagram indicates that USB high-speed signals operate within specified parameters. Courtesy of Tektronix.

The USB 2.0 eye diagram (Figure 3) looks like a typical eye diagram used in telecom testing, but don't let that fool you. In a telecom eye-mask test, you set an oscilloscope for continuous triggering and let the scope build the diagram over an unlimited number of bits. With a USB 2.0 eye diagram, the test system performs the measurement over one USB data packet (488 bits). After receiving test data from the scope, testing scripts take the data from each bit period (2.0833 ns) and overlay all bits onto a USB 2.0 eye mask.

If your USB 2.0 device's transmitted signals easily fall within the eye-diagram's mask, then your product will likely pass compliance tests. If you question the device's signal integrity based on the eye masks, then you can use time-domain reflectometry (TDR) measurements to better understand the problem.

Although not required for compliance, TDR measurements can provide useful information about how a device's connectors and board layout affect signal quality. Poor impedance matching of a device to a transmission medium can result in reflections that can distort a signal. A TDR must measure the impedance of the D+ and D- signal lines from a device's USB connector to its USB controller IC. Nominal impedance is 90 Ω with an acceptable range from 70 Ω to 110 Ω (Ref. 8). The TDR injects a 400-ps pulse into the UUT and measures the reflected current to calculate impedance. Impedances outside this range often require modifications to a board layout or require additional shielding.

So far, the measurements I've discussed involve measurements of transmitted signals, but you must test USB 2.0 device receivers, too. Compliance tests for USB 2.0 receivers require measurements for voltage sensitivity and deviations in bit rate.

In a receiver-sensitivity test, the test bed places the DUT into a special test mode called SE0_NAK. After establishing communications between the host controller and the DUT, the test fixture switches the device's data lines to the output of a data generator through a relay. The test program will also program the data generator to send a simulated input-request (IN) packet to the DUT. When a USB port running in high-speed mode properly receives an IN packet, it responds with a NAK packet, which indicates that the device received the IN packet but is not ready to respond.

In this test, a data generator sends IN packets at differential voltages greater than 150 mV. The compliance test software will start with a differential voltage of 400 mV. Then, the data generator sends an IN packet and the scope detects the NAK response. Next, the test system lowers the data-generator's output voltage in 20-mV increments, sending an IN packet at each new voltage. The receiver should respond with a NAK packet when its received signal level is 150 mV or greater.
In rushes the current

Upon connection to the bus, a bus-powered USB device will produce a surge of current before reaching a steady-state (Figure 4). If a device pulls too much current from the $V_{BUS}$ line, the current could steal power from other devices connected to that host controller or hub. Such a power loss could cause a device on the bus to reset.

To measure inrush current, use the inrush current test fixture, which has wires where you can attach a current probe. Although the test is called "inrush current," you actually measure charge in coulombs. Calculate charge by integrating the current. The area under the curve from the time of the initial spike in current until the current reaches steady state must not exceed 50 µC (Ref. 9).

Inrush current measurements tell you how much µA bus-powered device will upset the bus, but some USB devices require their own power sources. Self-powered devices such as printers, scanners, or hard-disk drives shouldn't apply any voltage to the D+ or D- data lines, nor should they push current onto the $V_{BUS}$ line.

To ensure proper connections to the bus, a self-powered device must initially keep its D+ and D- lines floating until the device sees voltage on the $V_{BUS}$ line. Then, the device should connect a 15-k pull-up resistor to the D+ line. That sets the device into full-speed mode. During this enumeration, a self-powered device shouldn't supply current to $V_{BUS}$ that causes the line to change by more than 400 mV. When you perform this back-drive-voltage test, monitor the voltage on the D+, D-, and $V_{BUS}$ lines both before and after enumeration. The test software and scripts will let you see if the voltages remain within tolerance.

Tests engender confidence

Precompliance tests such as those for signal integrity, inrush current, and back-drive voltage will help you gain confidence in your device's ability to pass USB 2.0 compliance tests. Compliance tests also include interoperability tests to verify that your product will operate properly when connected to a bus containing numerous other devices. Plugfests and test labs usually have many computers with USB 2.0 ports on hand, too. Your product will need to function properly with all other devices and hosts controllers.

Just because your product can pass a compliance test doesn't mean that it will perform to user's
expectations. Therefore, you should subject your product to numerous other tests. Don't just test under normal conditions; test your product under a range of temperatures and voltage conditions. Add noise to the data lines and power-supply lines that you connect to your product. For self-powered (as opposed to bus-powered) devices, add noise to the power lines and vary the voltage levels. Test your product with devices such as cameras and joysticks connected to the bus. These devices require real-time performance that your product should not compromise.

The following company information appeared in the original print version of this article. For up-to-date information about companies, visit our Buyer's Guide.

Manufacturers of USB 2.0 test equipment
The following companies manufacture equipment for testing USB 2.0 host controllers, hubs, and devices. Although you don't need a bus exerciser or protocol analyzer for compliance tests, these tools can help you debug new designs.

Bus exercisers, protocol analyzers

<table>
<thead>
<tr>
<th>Company</th>
<th>Phone Number</th>
<th>Website</th>
</tr>
</thead>
<tbody>
<tr>
<td>BusProbes</td>
<td>408-279-2622</td>
<td><a href="http://www.busprobes.com">www.busprobes.com</a></td>
</tr>
<tr>
<td>Catalyst Enterprises</td>
<td>408-365-3846</td>
<td><a href="http://www.catalyst-ent.com">www.catalyst-ent.com</a></td>
</tr>
<tr>
<td>Computer Access Technology</td>
<td>408-727-6600</td>
<td><a href="http://www.catc.com">www.catc.com</a></td>
</tr>
</tbody>
</table>

Data Transit
408-264-4300
www.data-transit.com

Digital oscilloscopes

<table>
<thead>
<tr>
<th>Company</th>
<th>Phone Number</th>
<th>Website</th>
</tr>
</thead>
<tbody>
<tr>
<td>Agilent Technologies</td>
<td>800-452-4844</td>
<td><a href="http://www.tm.agilent.com">www.tm.agilent.com</a></td>
</tr>
<tr>
<td>LeCroy</td>
<td>914-578-6020</td>
<td><a href="http://www.lecroy.com">www.lecroy.com</a></td>
</tr>
</tbody>
</table>

Tektronix
800-426-2200
www.tektronix.com/usb

Matlab software (required to run scripts)

<table>
<thead>
<tr>
<th>Company</th>
<th>Phone Number</th>
<th>Website</th>
</tr>
</thead>
<tbody>
<tr>
<td>The MathWorks</td>
<td>508-647-7000</td>
<td><a href="http://www.mathworks.com">www.mathworks.com</a></td>
</tr>
</tbody>
</table>

Compliance test fixtures

<table>
<thead>
<tr>
<th>Company</th>
<th>Phone Number</th>
<th>Website</th>
</tr>
</thead>
<tbody>
<tr>
<td>Agilent Technologies</td>
<td>800-452-4844</td>
<td><a href="http://www.tm.agilent.com">www.tm.agilent.com</a></td>
</tr>
<tr>
<td>FuturePlus Systems</td>
<td>719-278-3540</td>
<td><a href="http://www.futureplus.com">www.futureplus.com</a></td>
</tr>
<tr>
<td>Tektronix</td>
<td>800-426-2200</td>
<td><a href="http://www.tektronix.com/usb">www.tektronix.com/usb</a></td>
</tr>
</tbody>
</table>

References


3. Download a list of approved test labs at www.usb.org/developers/compliance/labs.

4. Download the test software and Matlab analysis scripts at www.usb.org/developers/tools. Scroll down to USBHSET.msi.

5. Download the test procedures for the Agilent and Tektronix scopes at www.usb.org/developers/docs.


