

PCIe clock distribution in embedded systems

AS PCIe RAPIDLY BECOMES THE I/O INTERFACE OF CHOICE FOR HIGH-PERFORMANCE SYSTEMS, CLOCK-DISTRIBUTION AND JITTER-REDUCTION REQUIREMENTS CREATE CHALLENGES FOR THE EMBEDDED-SYSTEM-DESIGN TEAM.

The PCIe (peripheral-component-interconnect express) protocol is highly desirable for communication across backplanes in embedded and other system types. However, for an embedded-system environment in which backplane connector pins are often at a premium, PCIe’s preferred clock-distribution scheme—using a star configuration of point-to-point connections—is less than ideal. You can distribute a PCIe-compatible clock using a single multidrop signal and still meet the tight jitter requirements of the PCIe Generation 2 specification.

CLOCKING IN PCIe

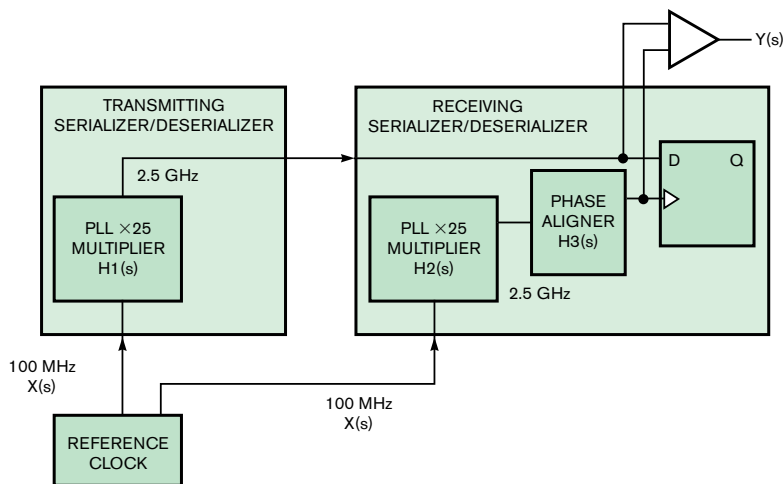
PCIe Base Specifications 1.1 and 2.0 define three clock-distribution models for the 2.5- and 5-Gbps signaling rates (figures 1 through 3). The common-clock architecture is the common method for a variety of reasons. First, most of the commercially available chips supporting PCIe interfaces use only this architecture. Second, this architecture is the only one that directly supports spread-spectrum clocking, which can be important in reducing EMI (electromagnetic-interference) peaking and, hence, simplifies the task of meeting electromagnetic emissions limits for the system (Figure 4). Finally, this architecture is the simplest to conceptualize and design.

The most significant disadvantage of the common-clock architecture is the need to distribute the reference clock to each PCIe endpoint in the system. The clock’s 100- or 125-MHz frequency and the PCIe protocol’s tight jitter requirements further complicate this task. For 2.5-Gbps operation, the limit is 86-psec p-p phase jitter for a sample set of 106 samples. The 5-Gbps operational limit is 3.1-psec-rms jitter. However, to operate at 5 Gbps, a transceiver first negotiates at 2.5 Gbps and then moves up to the higher rate if both ends can do so. That is, if the system supports any 5-Gbps links, then the reference clock must meet both jitter specifications.

The separate- and data-clock architectures avoid these limitations but substantially increase the complexity of the clock-system design and don’t support spread-spectrum clocking without the use of sideband signaling. The

governing specifications for reference-clock jitter are PCIe Base Specifications 1.1 and 2.0, and PCIe Jitter-Modeling Revision 1.0D and PCIe Jitter and BER (bit-error-rate) Revision 1.0 detail the method for verifying jitter compliance. The electromechanical specifications provide mechanical-form-factor information, electrical-signal definition, and functional definitions. Some of these specifications, such as Card-Electromechanical Specifications 1.1 and 2.0, also provide jitter budgeting among the reference clock, transmitting PLL (phase-locked loop), receiving PLL, and media. Strictly speaking, the Card-Electromechanical Specification applies only to PC-, server ATX- (advanced-technology-extended), and ATX-based form factors. Industry groups have published additional electromechanical specifications to cover other form factors, such as Mini-Card-Electromechanical Specification 1.2 for mobile-computing platforms.

For most embedded systems, these specifications provide guidelines that designers can use in whole or in part to specify the embedded system’s PCIe clock-distribution scheme. For example, many of the Card-Electromechanical documents



NOTES:
SYSTEM-TRANSFER FUNCTION IS $H(s)=H3(s) \times [H1(s) \cdot H2(s)]$.
OUTPUT $Y(s)=X(s) \times H(s)$.

Figure 1 The common-clock architecture is one of three methods for clock distribution that the PCIe base specification defines.

specify the use of the HCSL (host-clock-signal-level) protocol for distributing the reference clock. However, many embedded systems use LVPECL (low-voltage-positive-emitter-coupled-logic) signaling or M-LVDS (multipoint-low-voltage-differential signaling) to achieve a greater reach, noise margin, or both on their clock-distribution network.

Many embedded systems distribute a large number of high-speed signals, including clocks, across their backplanes. To deal with the often-heavy electrical loading on those backplanes, these signals tend to have powerful drivers and, hence, high edge rates. This situation presents the danger of crosstalk and other signal-integrity problems, especially when the backplane has a lighter load than the worst-case design. Another related design challenge is that PCIe specifies reference clocks of 100 or 125 MHz, which are difficult to distribute cleanly over a long, heavily loaded backplane.

In addition to the PCIe specifications' tight jitter limits and need for a longer signal reach, the number of signals that can transit the backplane connectors and the backplane itself also constrain embedded systems. Defining the connector pinouts is one of the more critical tasks when specifying the system.

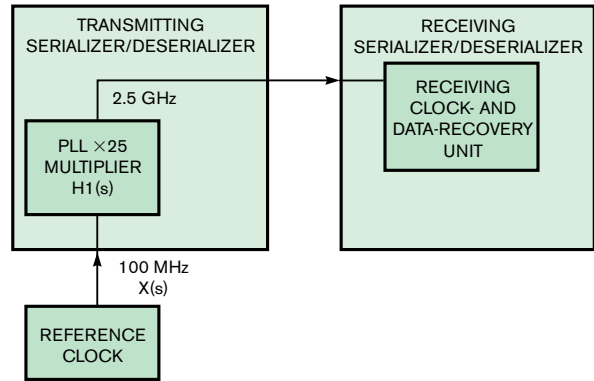
COMMON-CLOCK-DISTRIBUTION SCHEME

Due to the clock frequency and jitter constraints, most common-clock-architecture designs distribute their reference clocks using point-to-point differential-signaling pairs, one of which goes to every PCIe endpoint in the system. If your design has multiple PCIe endpoints on a single card, you can take in a reference-clock input from the backplane and provide a clock-distribution network on the card using zero-delay buffers. Even this task can be difficult to design, however, given the jitter constraints of 5-Gbps PCIe operation.

Assuming that you could design such on-card distribution schemes, they still require a point-to-point connection from the PCIe root to every card in the system. In embedded systems, this requirement adds a lot of connector pins to the root-card slots and a lot of traces with special routing requirements to the backplane. It also means that the slot that the root card plugs into has a different pinout from that of the other slots.

One approach to solving these problems is to divide the PCIe reference clock on the root card and distribute it across the backplane using a multidrop M-LVDS and then to multiply it to the desired frequency or frequencies on the destination cards. Although conceptually simple, this approach is tricky to achieve within the jitter constraints of PCIe (Figure 5).

This approach allows you to use an M-LVDS pair to drive or receive a PCIe-compliant reference clock. In many embedded systems, the cards operate as roots or endpoints depending on the application, the slot assignment, or both. A card that operates in only one of those modes would be simpler than the one in Figure 5. One card in the system would act as the root, generating a reference clock meeting the PCIe constraints from its onboard crystal. This clock would drive any onboard PCIe devices from an internal clock-distribution network. The clock would also go to a non-PLL divider circuit that would divide it from 100 or 125 MHz to the backplane frequency of 25 MHz. It would then drive the divided-down reference clock to the rest of the cards in the system. All the other cards in the system would disable the use of their onboard clock generators, tristate



NOTES:
SYSTEM-TRANSFER FUNCTION IS $H(s)=H1(s)$.
OUTPUT $Y(s)=X(s)\times H(s)$.

Figure 2 The PCIe data-clock architecture combines the reference clock with the transmitted data.

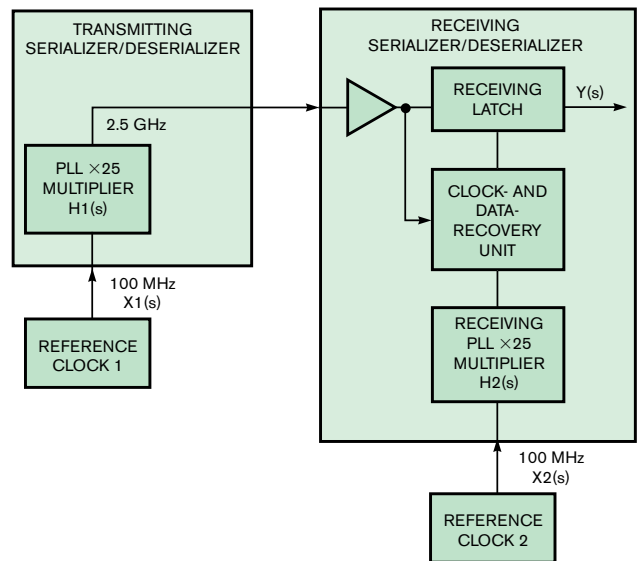


Figure 3 The PCIe separate-clock architecture employs a second reference clock in the receiver.

their drivers for the reference-clock traces, and receive the reference clock from the backplane. This clock would multiply using a PLL-based zero-delay buffer to the required onboard reference-clock frequency and then travel to the other cards. The circuitry that receives and multiplies the reference clock from the backplane would usually reside on the root card and could generate the second reference-clock frequency, if necessary. To achieve the low jitter that PCIe requires, you can incorporate jitter attenuators for the clock synthesizer and the zero-delay buffer.

One of the main challenges of a design such as this is that PLLs filter high-frequency jitter higher than their loop bandwidth but add jitter at modulation frequencies lower than their loop bandwidth. PLLs also induce tracking skew because they

do not perfectly track phase and frequency variations of the reference-clock input. For a backplane-PCIe implementation such as this one, which involves two or more cascaded PLLs for frequency generation and translation, you must take great care to minimize phase jitter and PLL-tracking skew.

PCIe-JITTER MEASUREMENT

Before diving into an analysis of the performance of this design, you must understand the process by which PCIe analyzes jitter performance. One of the overarching concerns of the PCIe Jitter Working Group was to neither overspecify nor underspecify the reference clock. To that end, the group accounted for the filtering effect of the transmitting and receiving PLLs and phase interpolator on the reference clock and the peaking effects of these PLLs.

Although the group has yet to detail many portions, the process now has four high-level steps. First, determine the accumulated phase error for each cycle. For serial-data transfer, the accumulated phase error is more important than cycle-to-cycle jitter or period jitter, which are important characteristics of parallel buses. Second, apply the DFT (discrete Fourier transform) to the accumulated phase-error data to change from time-domain to frequency-domain analysis. Then, apply the system-transfer function to the DFT of the accumulated phase-error data and perform an inverse DFT to transfer the filtered accumulated phase-error data back into the time domain.

You perform the filtering analysis of the PLL system in the complex frequency domain by setting $s=j\omega$ in the system-transfer functions. This equation works well for continuous systems, but most modern PLL implementations are not pure-analog systems because they have digital components, such as the phase detector and feedback divider; thus, Z-domain digital analysis is more accurate. However, brief studies by the PCIe Jitter Working Group showed that S-domain analysis imposes minimal error, so the group used S-domain analysis for modeling. The S-domain approximation deviates significantly from reality when the reference frequency is less than 10 times the PLL bandwidth, and designers must keep that fact in mind when selecting a PLL (Reference 1).

JITTER-MEASUREMENT TIPS

An improper measurement method can easily lead to jitter measurements that are twice as great or more than you would get using correct techniques. Here are a few tips:

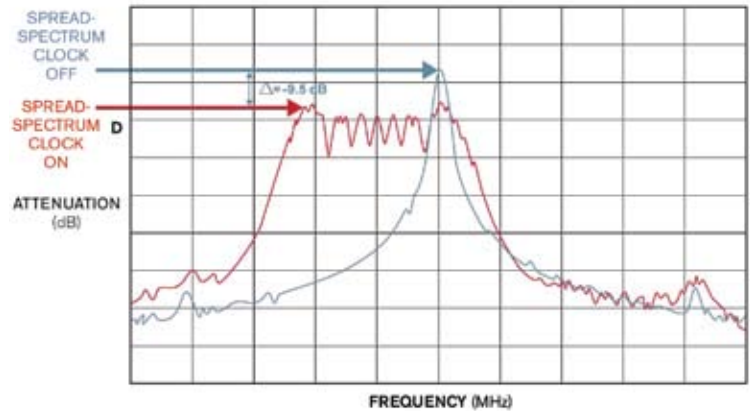


Figure 4 Spread-spectrum clocking can reduce EMI peaking to meet system-emissions limits.

Use shielded coaxial cables from the device under test to the oscilloscope, and terminate the clock to the oscilloscope input. If using high-impedance probes, use a low-capacitance probe and a ground clip rather than a wire. Use the highest possible sampling rate consistent with the required sample size. Maximize the vertical scale on the oscilloscope screen for accurate voltage measurements. Keep monitors, switching power sup-

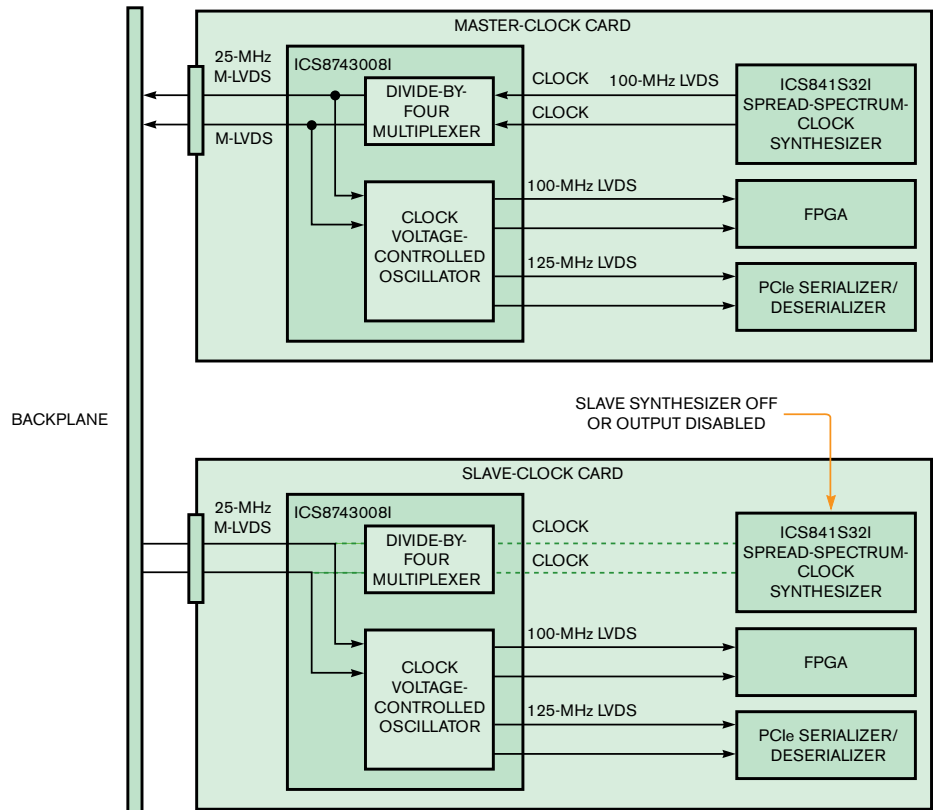


Figure 5 One technique for meeting the jitter constraints of 5-Gbps operation is to divide the reference clock, distribute it across the backplane, and restore the desired frequency at the destination.

plies, and cell phones away from the device under test. Use a linear power supply whenever feasible. When performing differential measurements, ensure that you have deskewed the two cables relative to one another.

ANALYSIS OF IDT'S SOLUTION

Engineers built a prototype of the circuit in **Figure 5** daisy-chaining an IDT (Integrated Device Technology, www.idt.com) ICS841S32I characterization board, an IDT ICS8743008I board, and a second ICS8743008I board representing the slave card. They took measurements at the output of the second ICS8743008I. They then offloaded the clock-period data from the oscilloscope and postprocessed the data with a jitter-analysis script, which performs the necessary frequency- and time-domain analysis (**Figure 6**).

The result for the 2.5-Gbps-analysis method is 18.91 psec. This result meets the PCIe peak-to-peak-phase-jitter spec of 86 psec with a factor-of-4.5 margin. For 5-Gbps operation, PCIe specifies rms phase jitter rather than peak-to-peak phase jitter. These

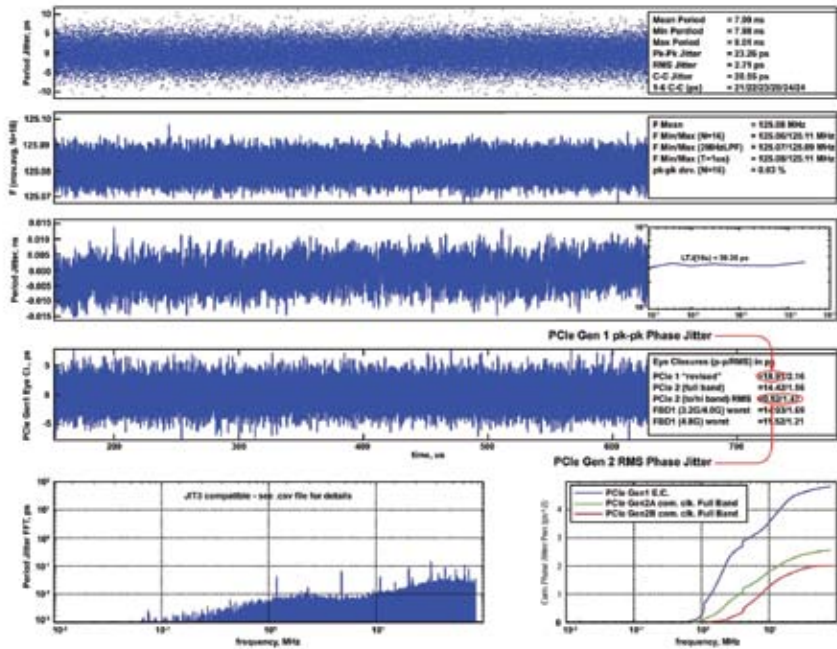


Figure 6 The results of the jitter analysis show that the clock-division strategy meets and exceeds the PCIe specifications for both 2.5- and 5-Gbps operation.

results also exceeded specifications: 0.52-psec rms low-band jitter and 1.47-psec high-band jitter versus a 3.1-psec specification limit.

For 5-Gbps operation, PCIe specifies two transfer functions and two frequency ranges for analysis in the frequency domain. The pole frequencies for these transfer functions are 5 and 16 MHz for the first transfer function and 8 and 16 MHz for the second transfer function. The two frequency bands over which you analyze the jitter are 10 kHz to 1.5 MHz for the low band and 1.5 MHz to the Nyquist frequency for the high band. For the Nyquist frequency, you analyze as much as half of the reference-clock frequency. For example, for a 100-MHz frequency, the frequency-domain analysis would extend to 50 MHz. The script reports the worst case between the two

transfer functions across each frequency-analysis band.

The originators of the PCIe standard defined it primarily for use in PC systems, but, due to its low pin count and scalable high performance, it is rapidly becoming the I/O interface of choice for components in almost all applications. The high speed of the reference clock that you must distribute, along with the option for

two compliant reference-clock speeds, poses some challenges for embedded-system designers who want to use PCIe components.

One tested approach allows a system to use components supporting the 100- and 125-MHz reference-clock options and allows you to distribute this clock over an M-LVDS differential pair to all cards in the system. This approach also allows you to configure cards so that they can act as a root or an endpoint as the application dictates and can reside in any slot in the system. Furthermore, the approach lowers the operating frequency for the reference clock on the backplane, easing the routing constraints and crosstalk performance for that signal. **EDN**

REFERENCE

1 "PCIe Reference Clock Requirements," Integrated Device Technology.

AUTHORS' BIOGRAPHIES



Ian Dobson is senior manager of system technology at Integrated Device Technology, where he has worked for five years. He identifies how IDT can best address customers' needs. Dobson holds a bachelor's degree in applied sciences from the University of Waterloo (Waterloo, ON, Canada). You can reach him at ian.dobson@idt.com.



Jim Holbrook is director of systems engineering in the communications division of Integrated Device Technology, where he has worked for seven years. He provides product and technology definition, strategic planning, and market analysis. Holbrook has 13 years' experience in the semiconductor industry. He holds a bachelor's degree in electrical engineering from the US Naval Academy (Annapolis, MD). He completed Navy nuclear-engineering training in 1988, followed by seven years in the US Nuclear Submarine Force. You can reach him at jim.holbrook@idt.com.

➤ [Go to **www.edn.com/ms4314** and click on Feedback Loop to post a comment on this article.](http://www.edn.com/ms4314)

➤ [For more technical articles, go to **www.edn.com/features**.](http://www.edn.com/features)