

## **Basics of High-Performance SerDes Design: Part I**

*by Edmund H. Suckow,  
Applications Engineer, Fairchild Semiconductor International,  
South Portland, ME*

*Existing TTL applications are easily converted to a high-speed SerDes system, as long as jitter, matched line lengths, impedance matching, and other critical issues are factored into the design*

SerDes devices have been around for more than five years but, until recently, have had little visibility as true interfaces. Due to the recent surge in LVDS technology and the realization of its common-mode versatility, SerDes now has an excellent partner for future expansion. Composed of dedicated serializer/deserializer pairs, SerDes technology opens the door to tremendous increases in bandwidth Vs. previous technologies, creating an environment in which SerDes applications are rapidly evolving to meet the demand for denser, higher-speed communications. By compressing numerous data inputs into differential pairs, increased throughput may be universally achieved in a multitude of applications.

SerDes device specifications, however, are sometimes difficult to understand which, in the past, has prevented designers from optimizing their full performance capability. Like any new interface technology, SerDes does pose a slight learning curve. To help overcome this many of the issues that arise during a SerDes design will be explored in this two-part article. More importantly, answers are provided to questions that typically surface during the process of selecting the correct interface technology for the application at hand.

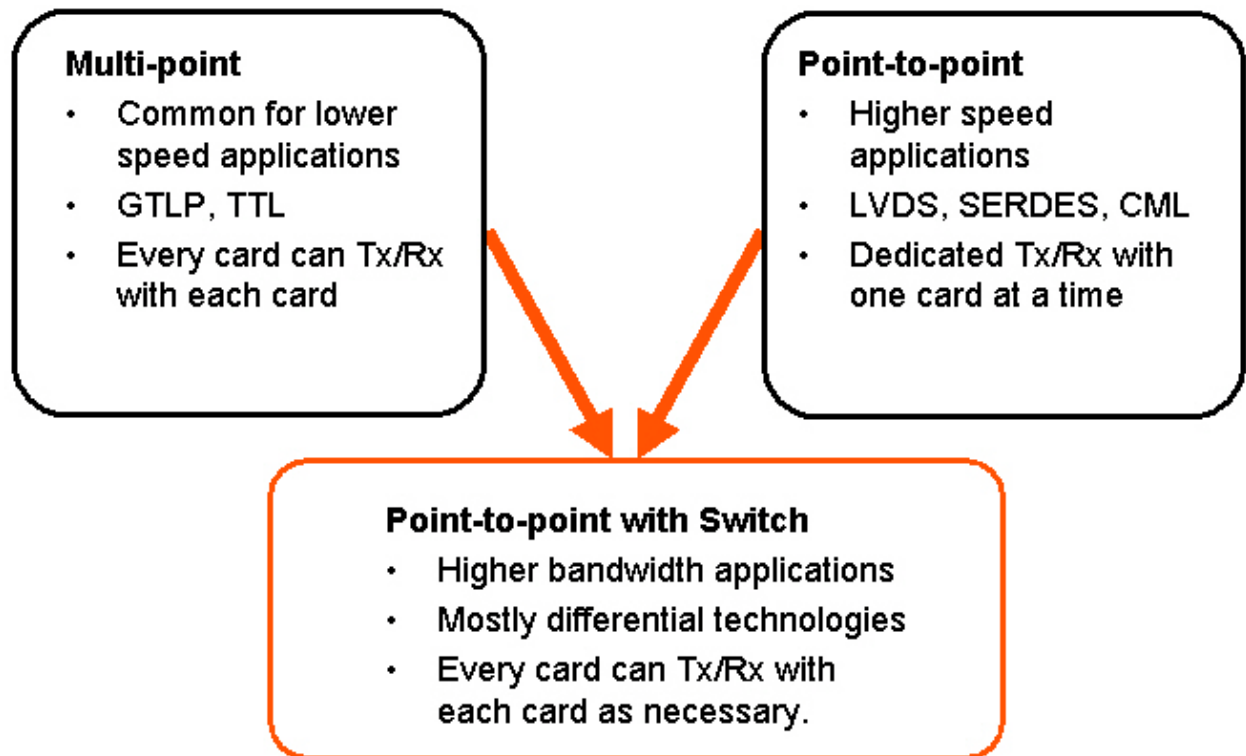
The following technical discussion of jitter, timing requirements, transmission media evaluation, and termination techniques applies to a range of SerDes devices including the popular 28:4 family used in Flat Panel Displays (FPD) and the 10:1 family frequently used in base station applications. Encoding features are also discussed for specific high-end SerDes devices. Data acquired by Fairchild Semiconductor's Ensigna Laboratory is used to illustrate the design tradeoffs associated with these specifications. Ultimately, through a more thorough technical understanding of SerDes, designers will be better equipped to select between alternative SerDes technologies to obtain optimal design performance.

### **SerDes Applications**

Applications like FPDs and wireless base stations require wide data widths with minimal skew between outputs and devices. In terms of drive needs the 10:1 devices (1023/1224) offered by various vendors offer enough drive to allow a small multi-drop or multi-point configuration. A common configuration for devices with low-voltage differential switching (LVDS) outputs is a simple point-to-point system across a printed-circuit board

(PCB) or cable. The 10:1 devices typically use an output termination resistor sized down to  $27\ \Omega$ , as required by multiple devices coupled to a single parallel, i.e., multi-drop, path. (Additional information about termination schemes follows in **Failsafe and Termination Circuitry** in Part II.)

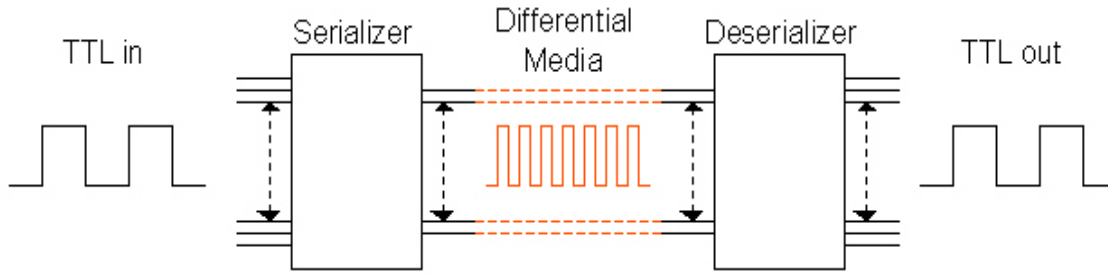
A few significant differences exist (see Fig. 1) between multi-drop and point-to-point configurations. The current direction of the backplane market is a point-to-point topology with a master switch to which SerDes is easily applied due to the minimal number of wires that must run through the switch. Redundancy is also an available advantage for a switch interface.



**Fig. 1: Configuration Comparison Between Multi-Drop/Point-To-Point Applications**

### **SerDes Defined**

As noted SerDes technology comprises a dedicated serializer/deserializer pair. Without a coupled pair the technology cannot achieve its forecasted bandwidths. A custom serialized output is the result of stringent timing components and reproducible signal positioning. On typical inputs, TTL signals with a 0 - 3V swing enter the serializer *horizontally* and are then *vertically* aligned such that in one clock period one set of parallel bits, or just one word, is transmitted. In this basic serialization concept (see Fig. 2) it is easily recognized that the internal frequency of the serializer must be faster than the incoming TTL data.

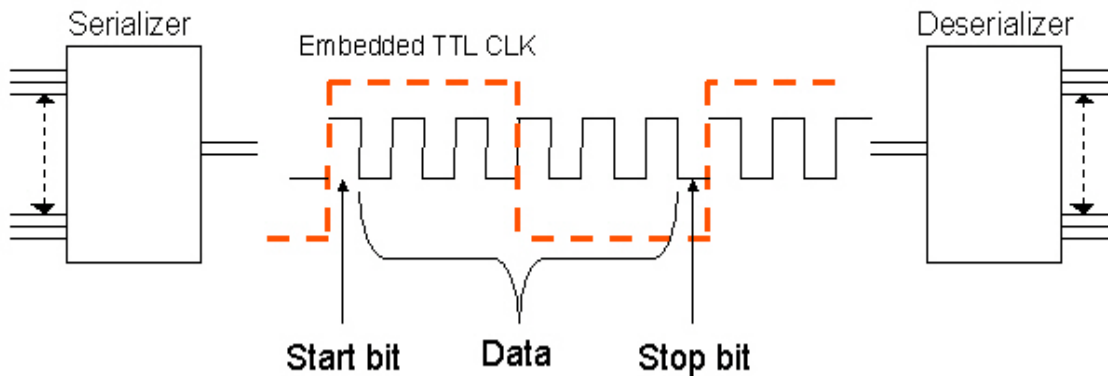


**Fig. 2: Flow Diagram For A Serializer/Deserializer Pair**

The internal clock frequency is set by the compression/decompression factor of the device pair. One would logically expect a 10:1 serializer to simply provide an internal frequency of 10 times the TTL frequency. However, this is not always the case. The 10:1 SerDes pair currently on the market began as a custom pair that required the two devices be connected only by a single data pair. How, then, do both devices share the clock?

The 10:1 family (see Fig. 3) uses an embedded clock - made up of an always high start bit and an always low stop bit - to synchronize the serializer and deserializer. The placement of these two bits is crucial to the low-skew operation of the device. Between these bits are ten distinct data bits making the total length of one serialized signal actually 12 bits. When looking at the entire sequence it is difficult to see exactly where the oscilloscope is triggered in the data pattern. For this reason it is common to insert a known data pattern, K 28.5 or similar, when trouble shooting.

**To decrease the number of wires between the Serializer and Deserializer use a device with an embedded clock**



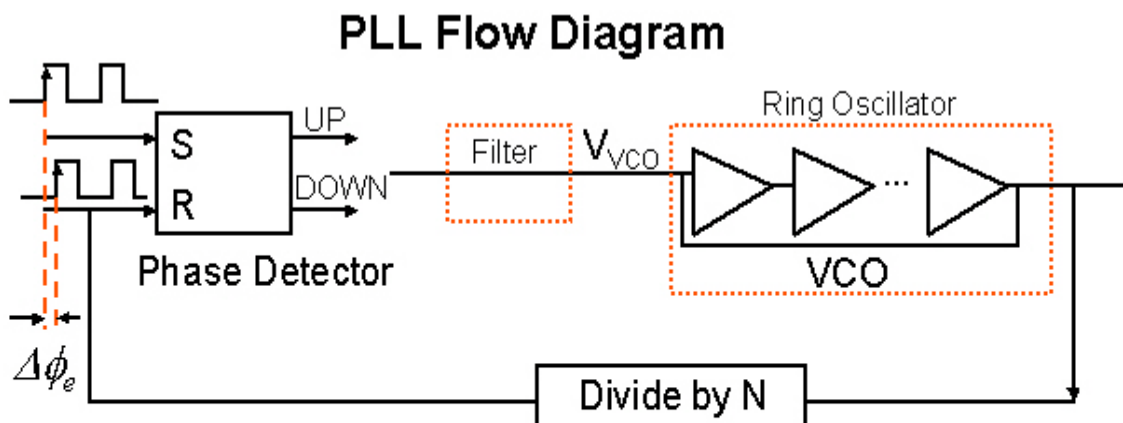
**Fig. 3: Clock Embedded Into Serializer Output**

## PLL Characteristics

A phase lock loop (PLL) is used to keep time for the serializer/deserializer pair. The PLL is internal to each device and is required to lock to the input clock frequency, perform the correct multiplication factor, and maintain its output with minimal jitter. A PLL is used because of its inherent feedback path allowing constant correction if a minor change is seen in the input signal edge position or period. To understand the SerDes technology, it is important to have a basic understanding of how a PLL operates.

All SerDes PLLs have an input frequency (typically CLKIN) and an internal core frequency that needs to be synchronized with the input. The internal frequency is responsible for the serialization timing and without the PLL running, data compression is not possible. A phase detector is used in the front of the loop to assign a phase delta value which is typically fed into a filter that controls the voltage sent to the Voltage Controlled Oscillator (VCO.) The VCO accepts this voltage and modifies the ring frequency to track. A *divide-by* transfer function is then responsible for the final ratio of the CLKIN to the PLL frequency (see Fig. 4.) It is not as simple as it appears and there are several key factors to keep in mind for PLL operation: There is the time it takes to lock, the power consumed, the resolution of each loop correction factor, and the effect of jitter on the circuit.

Perhaps, the most qualified test for comparing PLLs from separate vendors is a jitter-bandwidth test. The equipment involved makes this test expensive but the results can be very beneficial in choosing between vendors. (See **Jitter** later in Part I.)



- Every SERDES has an internal PLL
- PLL bandwidth is characteristic of jitter transfer
- Can lock to Synchronizing pulses or random data

**Fig. 4: Flow Diagram For A PLL Control Circuit**

## **Output Drive**

The most common drive used for SerDes technology is LVDS, the major advantage of which is its common-mode noise rejection. Due to the differential pair any noise is seen by both signals and the differential measurement between the two is theoretically unchanged. This noise-rejection characteristic allows the use of low-amplitude pulses (typically around 400 mV) that result in higher achievable speeds due to the decreased voltage levels.

Emitter-coupled logic (ECL) is also used in place of LVDS in some high-performance SerDes devices when higher drive and higher speeds are required. One version seen lately, the low-voltage positive-emitter-coupled logic (LVPECL), is capable of driving 50 mA continuous compared to the 5 mA or 6 mA associated with LVDS. This is often the driver of choice when driving a cable longer than 15 m at high speeds. ECL-type devices commonly achieve speeds of 2 GHz compared to the 600 MHz associated with LVDS. However, the power tradeoff is tough to accept, since power consumption with the former is roughly five to six times greater than that of LVDS. For this reason, ECL-based serializers have a more focused application list than that of the LVDS serializer.

## **Encoding**

Another characteristic of high-performance SerDes devices is 8B/10B encoding. Incorporated from FibreChannel standards, 8B/10B encoding is used to dc balance a word and minimize errors. Instead of sending an 8-bit word composed of all “zeros” and a single “one,” a device with 8B/10B encoding truncates the word into a dc-balanced 10-bit word made of zeros and ones, decreasing the amount of data-dependant jitter. However, frequency overhead is now required since it takes 10 bits to send 8 bits of data. But a system with fewer errors spends less time repeating data submissions, so increasing overall system throughput. Extra silicon is also required to facilitate the encoding algorithms which increases the price of the device. Detailed timing diagrams of this encoding scheme can be found in any FibreChannel standards document. Fairchild Semiconductor’s FIN7216 device, for example, employs this type of encoding.

## **The Importance Of Understanding Datasheet Specs**

With several vendors offering very similar SerDes devices it is important to understand datasheet specifications. For example, comparing jitter specifications can appear to be an apples-to-apples comparison but in reality a minor adjustment in one vendor’s testing methodology will effectively discredit any comparison between the two specifications. It is therefore critically important to not only understand the specification itself, but also to be familiar with the way in which the characterized data was collected. Even with lengthy interface device data sheets information for every design application is impossible to present, in terms of characterizing data applicable to every design situation. If a question arises about a specific SerDes device do not hesitate to call the vendor. Not only is it

common for vendors' application engineers to communicate with a specific designer on a design win, it is expected.

The key word associated with this comparison is "interoperability." Designs usually require a second source but there is no guarantee that each vendor will only be communicating with its own complementing pair. Interoperability is therefore crucial. Anticipating this, Fairchild Semiconductor designed a SerDes Interoperability Demo Board for the 10:1 family (see **Media Evaluation** in Part II) that not only allows chip designers to pretest set interoperability conditions, but allows evaluations hands-on. The following datasheet specifications seem to have more questions than others.

## Jitter

When differential signals reach speeds over several hundred MHz eye-opening measurements, often characterized as jitter, are needed to evaluate signal integrity. Jitter can be defined as simply the time delta between the actual occurrence of the event and when the event was *supposed* to occur - with an event usually describing a rising or falling edge. Jitter can be described in several ways.

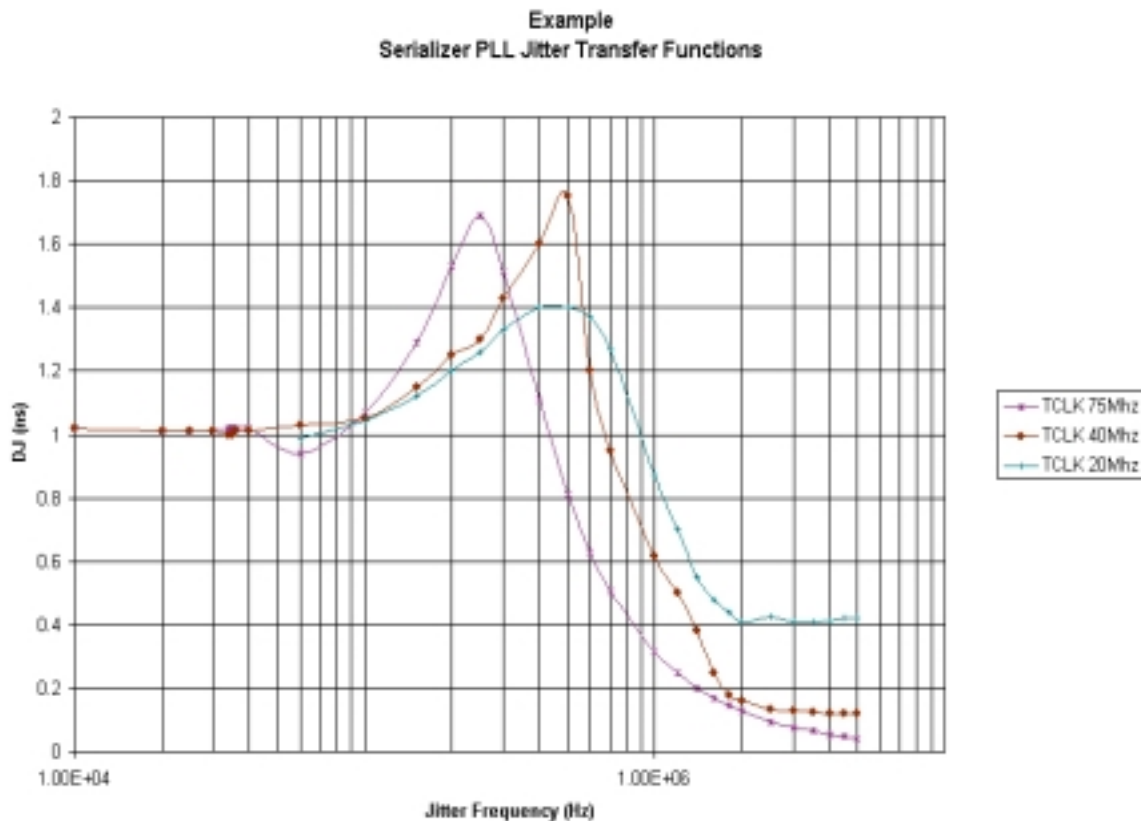
Total jitter (TJ), the collective jitter of a signal, is made up of deterministic and random components. Deterministic jitter (DJ) is a bounded measure, in that it is repeatable over time and measured between two bounded peaks in a jitter histogram. Random jitter (RJ) is not bounded and has a Gaussian distribution meaning that the measured RJ histogram is constantly growing with the number of hits in the measurement region or field.

Should a designer be concerned with jitter when designing a SERDES application? In a word, yes. It is important to remember that the link between the serializer and deserializer is fixed. All drive amounts and jitter are set by the type serializer. However, in order for the serializer to provide a clean signal to the deserializer a tight clock must be provided. This specification appears in serializer data sheets as  $t_{JIT}$  and applies to the TTL signal coming into the TCLK pin of the device. The way this jitter value is specified can vary between vendors so a call to the respective vendor may be in order. Naturally, the designer wants to avoid introducing further jitter to the SerDes link. Common differential design techniques include: matched trace lengths, close-coupling and tightly-controlled impedance (100  $\Omega$  for LVDS signals.) When placing bypass capacitors for each device using several vias and keeping power traces wider than typical signal traces can decrease inductance.

Any jitter on the TCLK input trace directly affects the internal PLL of the serializer and thus the placement of data bits in the serialized stream. The design of the internal PLL controls how jitter on this TCLK line is coupled to the serialized stream. Perhaps the best way to evaluate a specific vendor's serializer is to record a Jitter Transfer Function (JTF) which measures the amount of jitter either amplified or filtered by the PLL and lets the designer see the overall jitter performance of a device.

One of several methods of testing this consists of injecting specific controlled jitter into the TTL TCLK line and measuring the DJ on the LVDS clock out line. A common approach is to set the jitter amplitude to a small percentage of the bit interval (typically 10%.) Setting the jitter amplitude smaller than 1 ns is possible, but not recommended, because it requires even more detailed jitter measurement techniques on the output. Larger amplitudes are easier to observe, the goal of the test being to characterize the jitter transfer of the PLL by noting patterns in the JTF; specific values are immaterial.

The variable being modified during the data collection is the frequency of the jitter, which is the rate that the clock edge is being shifted from -500 ps to +500 ps (if 1 ns is used.) In the transfer function (see Fig. 5) sinusoidal jitter was injected over the TTL carrier frequency required for serializer operation, typically 40 MHz to 120 MHz. When selecting the resolution for jitter frequencies where measurements will be posted note that the plot will be presented using a logarithmic scale for the x-frequency axis. The plot (Fig. 5, again) shows a starting jitter frequency of 10 kHz and a final jitter frequency of 5 MHz. The maximum frequency is often declared by jitter generator limitations.



**Fig. 5: The Jitter Transfer Function In A Serializer**

The JTF plot (Fig. 5, again) shows three devices with data collected at the same carrier frequency. At the left of the plot, the 1 ns of jitter sent into the device is evident on the output. This means that the PLL is able to track low-frequency jitter, i.e., this jitter is within the bandwidth of the PLL. As the frequency of the jitter increases, the output jitter

from the PLL gradually increases to a peak due to moderate PLL phase margin. So, for jitter at this frequency the PLL actually increases the jitter. When comparing two serializers the device with the lowest peak and lowest bandwidth generally transfers the lowest total jitter. The rate the waveform falls from the peak to the final level also plays a significant role in determining the bandwidth. In addition, the steeper slope attenuates the high-frequency jitter components better. At very high jitter frequencies the final resting point of the waveform can be thought of as the jitter inherent to the PLL design. As the carrier frequency increases (not shown in this plot) the core PLL jitter decreases, but the product ratio of inherent PLL jitter and clock frequency may actually increase at higher carrier frequencies. Therefore it is important to do any preliminary jitter evaluations at the frequency the future design will be operating.

### **About the Author**

Ed Suckow, applications engineer for SerDes products at Fairchild Semiconductor, is responsible for research and definition for SerDes and LVDS technology. He formerly worked in GTLP applications with Fairchild Semiconductor, in power supply design with NASA, and on project management for the US Army.

For more information on SerDes design:

<http://www.fairchildsemi.com/products/interface/index.html?aprilad>

