HOTLink™ Jitter Characteristics

Abstract

This application note describes the basics of jitter in transmission systems and, using HOTLink™ as the example, shows how it can be analyzed and measured. Specific characterization data is presented that will allow system integrators to understand the parameters needed to improve the reliability of their systems.

Introduction

This note examines jitter from three different perspectives. First, as a background overview, it describes a few basic "jitter" concepts that affect digital systems. Second, it describes the jitter performance and characterization of the HOTLink Transmitter (CY7B923). Third, it describes the jitter tolerance and feed through characteristics of the HOTLink Receiver (CY7B933).

Numerical characterization data is supported by descriptions of the various testing techniques and equipment that are required to obtain this information. Commercial, custom, and "home-brew" test equipment are described along with the connections used to gather data that illustrates the levels of performance attainable by HOTLink products.

The data contained in this application note will help users to understand the various characteristics of link components and HOTLink characteristics and capabilities. This data is offered to assist in the design of robust serial interconnect links.

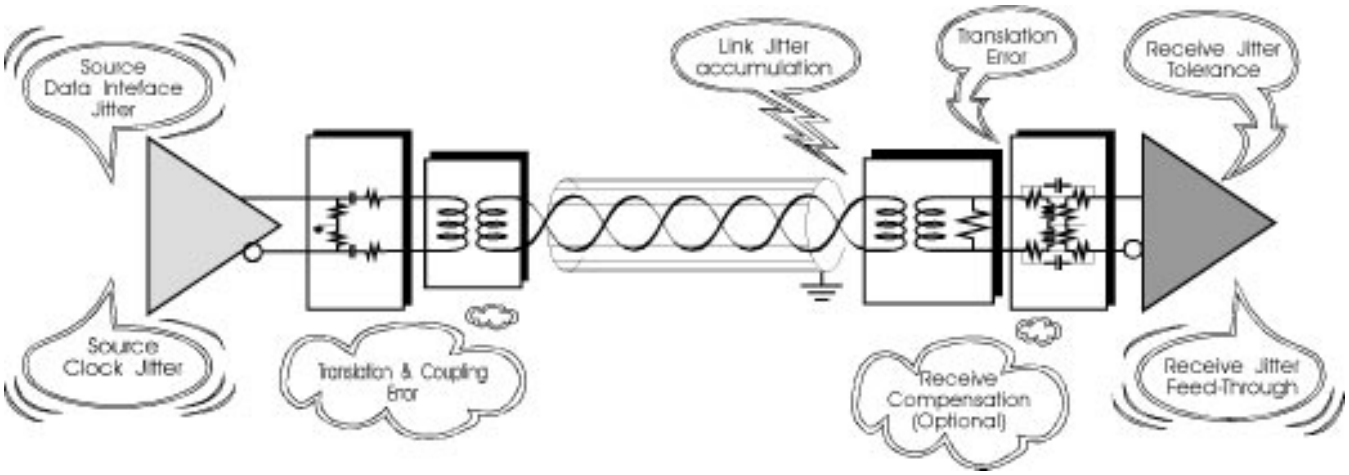

Figure 1. Link Jitter Budget Depends on Link Components

Jitter

Jitter is a high-frequency semi-random displacement of a signal from its ideal location. These displacements can occur in amplitude, phase, and pulse width, and are generally categorized as either deterministic or random. For data communications links based on (or similar to) HOTLink, measurement and specification of jitter is usually restricted to timing displacements.

Deterministic jitter are those timing variations that are repeatable within a system and whose cause can generally be directly attributable to specific physical components or events. An example of this would be the jitter caused by the frequency selective attenuation and phase delay of a signal in a transmission line.

Random jitter deals with those timing variations that are much more probabilistic in nature. While still observable and measurable in a system, this jitter is not directly predictable. Common sources for random jitter are thermal and electrical noise, both internal to and injected into a system or component.

Jitter in logic circuits is often characterized by its transfer function. This function, known as jitter feedthrough, is a measure of jitter output relative to jitter input of a system or component. Most circuits, when presented with jitter, tend to amplify that jitter in a few or many areas. Fortunately for data communications system (which are plagued by high jitter creation elements), application of properly designed PLLs (phase-locked loops) can actually reduce or remove large amounts of jitter from a clock or data stream.

Background—Jitter in Logic Systems

The timing of logic signals flowing through a logic system are often assumed to be a series of simple voltage transitions that occur after some fixed delay. While this is a convenient and usually sufficient assumption for the logical function of a device, it is insufficient to analyze the limits of the timing or the reliability of the design.

The delay through logic devices (i.e., gates, flip-flops and other common building blocks) is defined to a first order by the time it takes for the inputs, the internal circuit nodes, and the outputs to change from one voltage to another. Since there is always some uncertainty about the exact voltage present at any node in the circuit, various logic families have been devised with specific ways to assure reliable logic functions. Thresholds are well defined and intergate links have sufficient voltage margins to assure reliability. Typical components have output levels (e.g., Voh, Vol, etc.) that assure a significant voltage margin above and below the input thresholds (e.g., Vih, Vil, Vth, etc.).

Most logic model libraries document a fairly wide range of possible delays through a logic element. This range includes the effects of many internal characteristics such as differences in output resting voltage, threshold voltage, signal ramp rates, and (to some extent) the speed the signals travel along the interconnecting wire, metalization, and leadframes. These delays, while supposedly covering the minimum to maximum range for the part, assume specific external operating and signal conditions. By presenting the logic element with input, output, or power conditions beyond those assumptions, it is possible for these logic elements to exhibit apparent delays both faster and slower than the specified minimum and maximum.

The noise carried on the V_{CC} or Ground rails (both internal and external) affect the actual timing of the I/O transition by causing changes in the starting levels of the active transition. The illustration in *Figure 2* shows only the timing variation caused by ground bounce, but the influence of V_{CC} noise has a similar effect. If the signal begins its transition at some arbitrary but fixed time, and has a transition rate (i.e., rise time or fall time) that is mostly controlled by slew rate limiting effects not related to the power supply glitch, the effective timing will be determined by the placement of the glitch. If the transition begins on a glitch-peak, it will arrive at the threshold voltage a little early, and if the transition starts in a glitch-valley, it will arrive a little late. This change in timing is usually invisible to the external examiner (except as power supply induced timing variation) because much, if not all of the glitch is contained within the IC package, and is not externally observable.

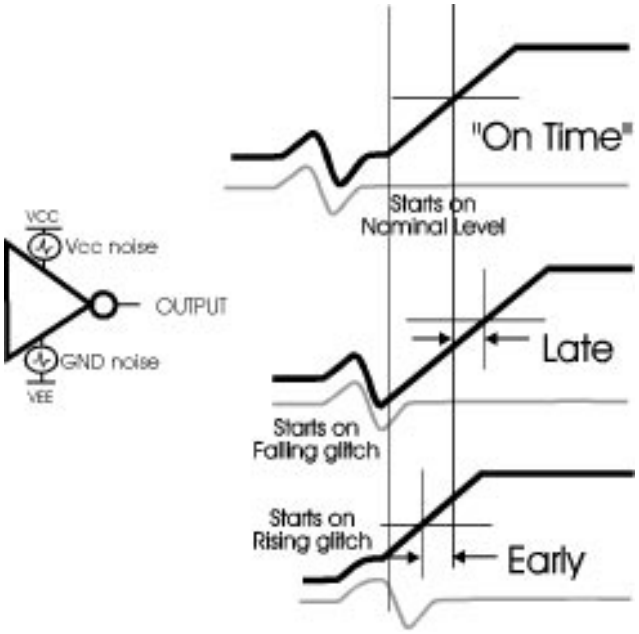

Figure 2. Power Supply Glitches Affect I/O Actual Timing

The effect of this variation in starting voltage can cause significant variations in timing. A signal that has a 1 ns/V ramp rate (TTL edges are usually between 1−2 ns/V, and can be much slower), will have an effective change in delay of about 1 picosecond per millivolt of disturbance. This equates to ±100 picoseconds of delay variation for ±100 millivolts of ground or V_{CC} noise, an amplitude which is normally deemed "quiet". When noise spikes approach 1 volt, delay variations could be expected to exceed 1 nanosecond. With a volt of power supply variation, other delay effects would surely begin to appear.

Additional timing variation can be caused by noise coupled into the external or internal logic through cross coupled logic paths (including package-pin crosstalk), or by power supply noise injection. These "minor" variations in delay are typically ignored in the analysis of the logical function, since there is sufficient overdrive (voltage noise margin) to assure that the logical function is achieved. However this assurance is not transferred to the timing margins of a logic design.

Most of the delay of today's high performance logic is caused by an output "ramping" from its resting voltage to the actual threshold voltage (the voltage at which the gate begins to make its logical decision and subsequently change its own outputs). Any disturbance in either the internal threshold or the ramping input or output will cause a change in the apparent delay through the gate (see *Figure 3*). All single-ended logic gates suffer from this variable-delay characteristic. Single ended circuits include all

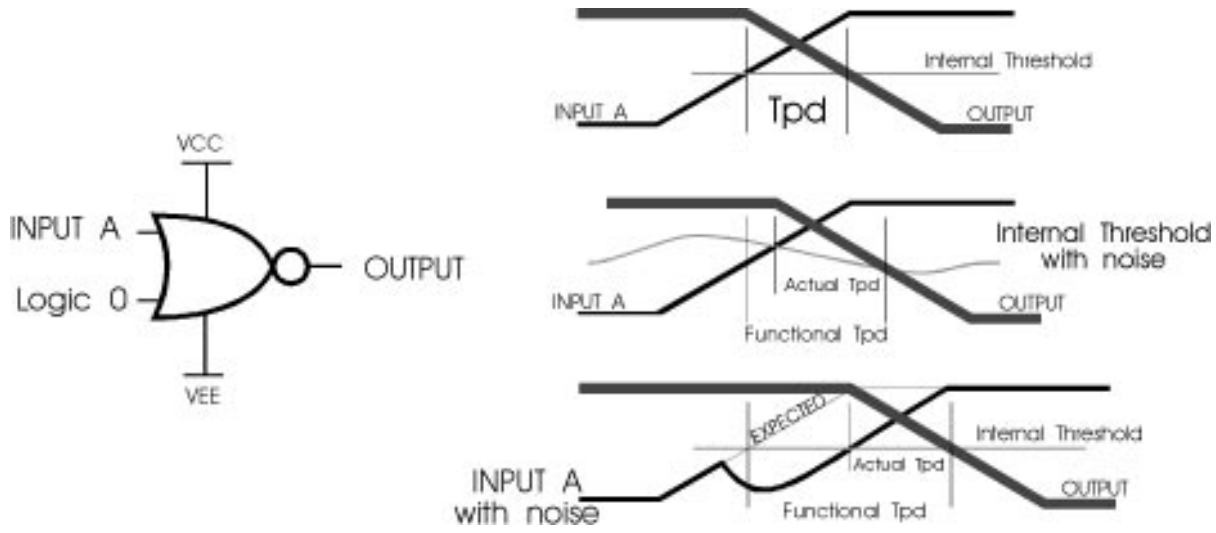

Figure 3. Delay Through a Logic Gate Changes with Injected Noise

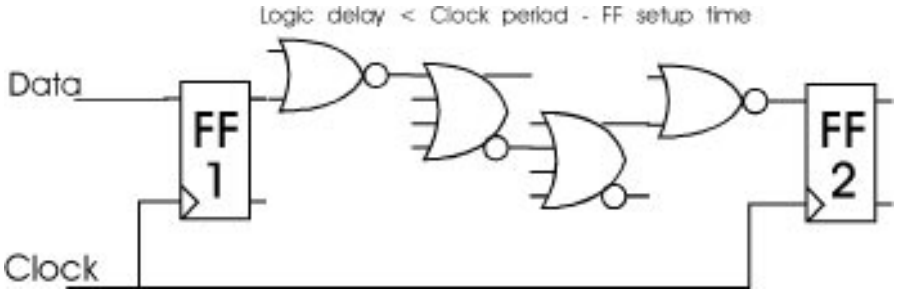

Figure 4. Typical Logic Path Delay Limited by Minimum Clock Period

TTL, CMOS, and any ECL logic that uses an internal or external threshold reference.

Differential circuitry can be used to partially mitigate the effects of injected noise, since the threshold of the gate is determined by a complementary output, hopefully carrying the same injected noise, but ramping in the opposite direction. The common mode range of such a differential gate helps to reduce many noise induced delay characteristics. All of the critical timing paths in HOTLink products are implemented with differential CML (Current-Mode Logic) signals to minimize crosstalk and V_{CC}-coupled noise-jitter effects.

Various design techniques have been developed that maximize timing margins in logic, but in most of these techniques the timing of any particular logic element is considered a constant (or a range of constants). Except for the well known metastability characteristic of storage elements, the design tools assume that each element has a fixed delay, and the only accommodation to metastability is to attempt to avoid the conditions that provoke the unpredictable behavior.

Traditional design practices work on the simple assumption that if the logic path (delay) between storage elements is less than the time between clocking edges by some comfortable margin, then the logic will behave exactly as the designer intended. As clock speeds increase and as product complexity increases this comfortable simplifying-fantasy becomes more difficult to maintain. As is well documented in other literature, if the transition on the DATA input changes later than the required set-up time prior to the active transition on the CLOCK input, the delay of FF−1 *(Figure 4)* may increase or it may refuse to store the expected data. If the path length to FF−2 is running near its maximum limit, this increased delay could propagate through the logic causing unexpected and undesirable results.

Designs that meet all manufacturers specified set-up and hold times can also experience variable delays through the flip-flop. As the input transition approaches the "actual" set-up time of the internal latches, delay will begin to change. *(Figure 5)* Typically, Tsetup is specified at the point where delay has changed by less than some arbitrary amount (usually about 10%) of the cell's "nominal" delay. Inside of that point, delay will increase radically until the flip-

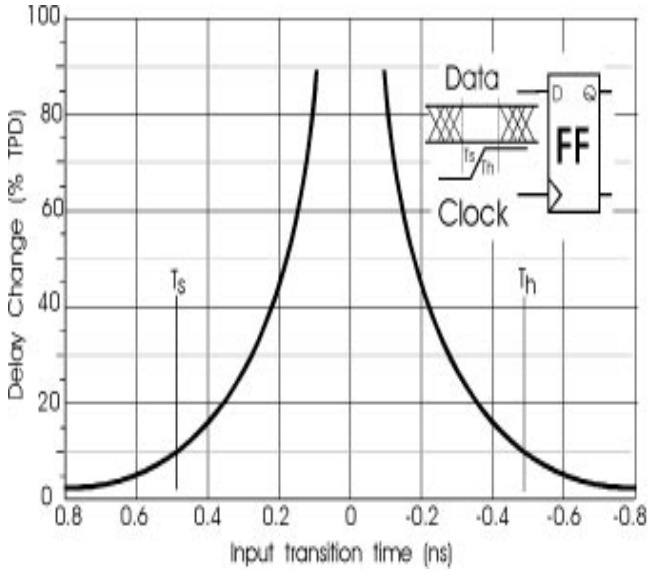

Figure 5. Propagation Delay Changes as Actual Tsetup is Approached

flop goes metastable. Similar effects occur as hold time approaches zero.

Even if the nominal delays of the intervening logic are within design margins, voltage-noise effects can change the delays of the combinational logic devices. If that happens, metastable effects might be observed in the system. Normally in digital-logic systems, great care is taken to assure adequate timing margin and then the error rate is "assumed to be zero," and ignored.

Jitter in PLL Systems

Phase Locked Loops are typically used as high speed clock multipliers or as precision clock recovery circuits. In their role as clock source generators, PLLs are characterized for their timing precision. This is usually because any jitter that appears on the clock line must be compensated by an equivalent reduction in the timing margin allowed between flip-flops.

Jitter can enter a multiplier PLL (see *Figure 6*) in several ways. The Clock input (1) can contain voltage-coupled noise or phase-noise that will affect the multiplied Bit Clock. The UP and DOWN outputs (2) of the PHASE FREQ DET are the digital to analog interface with the analog control circuits of the PLL and can suffer from the same voltage-coupled noise effects described earlier for logic. These digital signals carry the picosecond analog-timing information that controls the VCO. Any cross-talk or noise injection at this point will corrupt the "error" information that the PLL uses to maintain phase-lock with the input clock.

The output of the analog filter (3) contains both the gross center-frequency control, and the precision phase-control. Typically the input sensitivity of the VCO will be hundreds of megahertz per volt, and micro volts of crosstalk or power supply noise injection can add nanoseconds of jitter to the PLL output. Similarly, the capacitors (4) used in the FILTER (either internal or external) can be susceptible to noise injection which cannot be eliminated by any traditional circuit techniques.

HOTLink products use carefully designed, fully internal Metal/Oxide/Silicon (MOS) capacitors. These huge, matched devices minimize external noise coupling. For noise sources that cannot be avoided, the capacitors and all of the other analog circuitry are designed to make coupled noise more rejectable by using fully differential, common-mode noise reduction methods. Older PLLs often used ex-

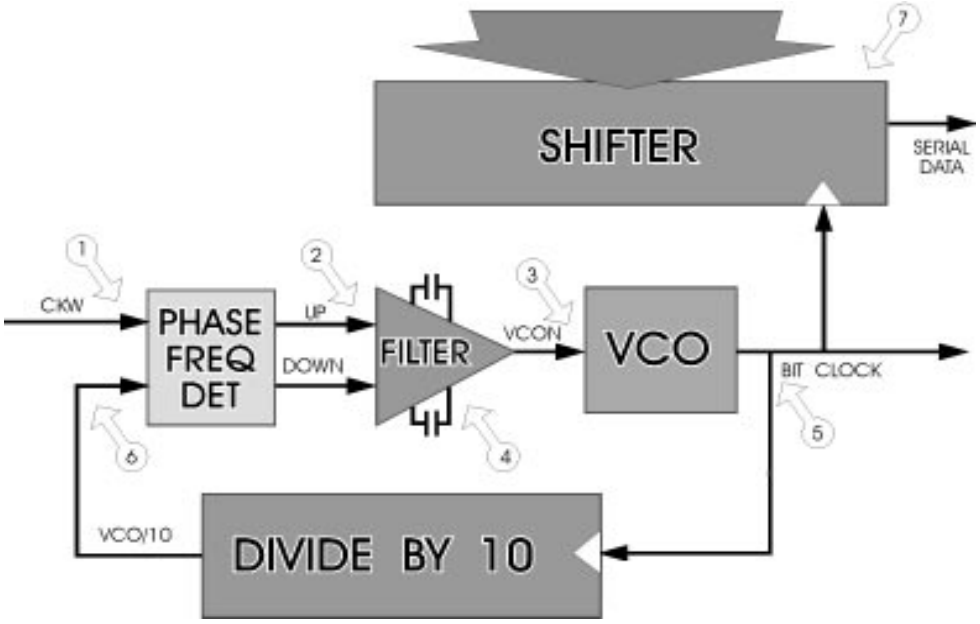

Figure 6. Clock Multiplier PLL Noise Injection Points

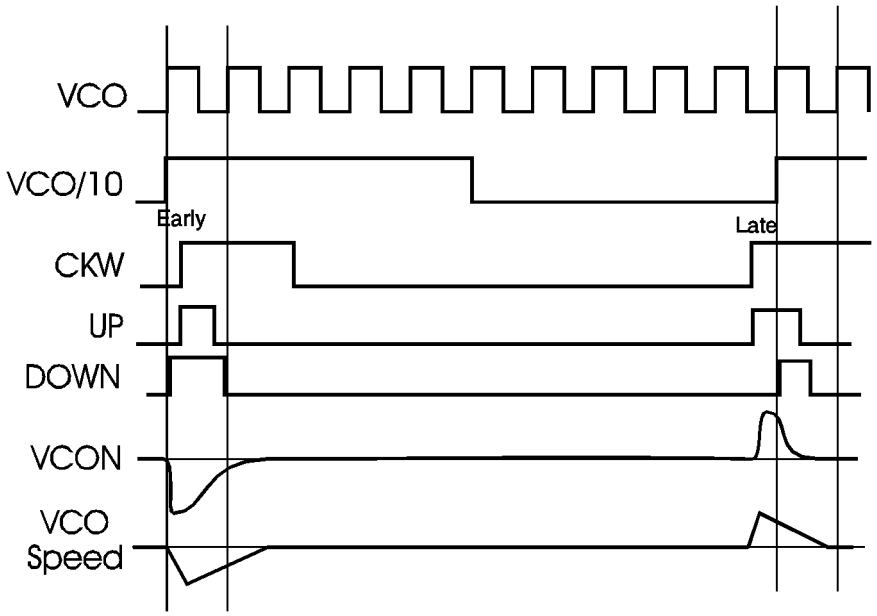

Figure 7. Phase/Frequency Corrections in Multiplier PLL

ternal capacitors which were notorious for noise injection through the external pins and circuit board traces required to connect these capacitors.

Noise injected at (2), (3) or (4), and to a lesser extent at the other points, can be only partially compensated by the normal filtering actions of the PLL. Noise at (2) or (6) will exhibit different effects than noise injected at (1) and will affect the Bit Clock (5) in different ways. These differences are illustrated in *Figure 11*, and will be discussed later.

Since the multiplier PLL only receives its correction information once every N VCO cycles (where N is the multiplication factor of the PLL, the VCO frequency divided by ten in this case), many specific errors will not cause a correction. Only the "average" of noise-induced errors will result in compensatable disturbances. "Instantaneous" errors will not be compensated by the PLL at all, especially if there are other errors of similar magnitude and opposite sign between reference updates.

Logic noise as described earlier can be injected into the Recovered Bit Clock (5) or at the feedback reference (6). These can be avoided by careful differential circuit and logic design. The parallel data input to the SHIFTER (7) can cause transmitted output jitter which is a function of the data pattern being sent (i.e., DDJ) as the set-up and hold times of the output flip-flop vary.

The operation of the PLL can cause jitter just by its normal operation *(Figure 7)*. Whenever the phase detector adjusts the frequency of the VCO, it causes an instantaneous change in phase as part of the adjustment operation. This instantaneous phase change, followed by a drift until the time of the next correction, is the normal operation of the loop. Ideally, the correction would be small, and entirely contained within one clock cycle, but if it is larger or lasts longer than one cycle of the VCO, it can cause bit-to-bit phase differences (i.e., jitter).

Clock Recovery, Data Separator PLL

The PLL used for clock synchronization and data recovery shown in *Figure 8* is different from the one described in *Figure 6*, which is used as a clock multiplier. The phase correction information comes from comparisons between an arbitrary input pulse stream and an internal bit-rate VCO. In contrast to the Phase-Frequency Detector (PFD) used in the clock multiplier, this PLL uses a detector that is sensitive only to phase errors. Missing data transitions are ignored, and corrections follow each and every data transition. In contrast to the predictable correc-

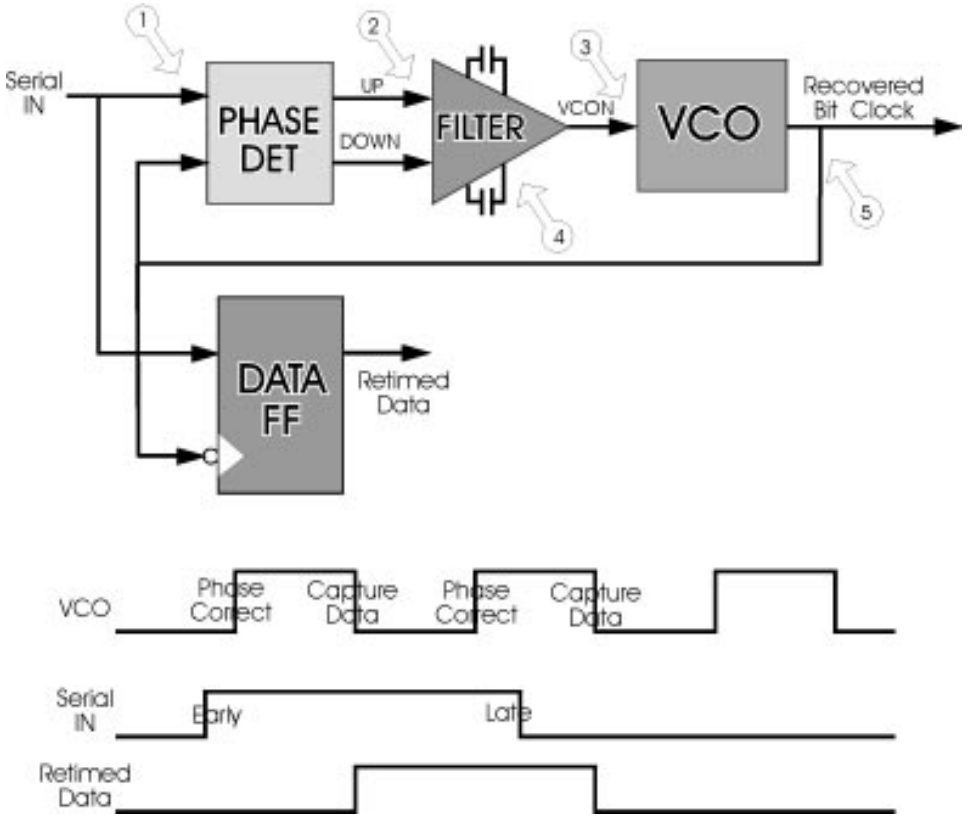

Figure 8. Receive-PLL Block Diagram

tion rate of the PFD, the Phase Detector will make corrections at the rate of the incoming data. It can vary from one correction per VCO cycle (when data contains alternating 10101...) to once per byte (or less) for some serial protocols. This variation in correction density can cause some forms of jitter and, by affecting the loop stability and bandwidth characteristics, will affect jitter feed through.

Jitter can enter a synchronizing PLL in several ways. The input data (1) will contain significant jitter which accumulates on the serial transmission link. This is the jitter that the receive PLL is intended to remove.

The noise injection points at (2), (3), (4), and (5) are the same as those in the multiplier PLL, and affect the receiver PLL in similar ways. The main difference is that this PLL gets a phase-error update on each input data transition. This allows noise events to be corrected more often than those in the multiplier PLL, but the noise induced corrections can be affected by the corrections already required by the jittered data. Conversely, these noise-induced jitter components reduce the data-recovery circuit's tolerance to input data jitter.

The Phase Detector (or PFD) in clock multiplier PLLs and in clock synchronizer PLLs is intended to give a "unit" of phase correction information for a "unit" of error. This correction should be directly proportional to the error, regardless of error magnitude. A poorly designed (or poorly implemented) phase detector in any PLL, either a multiplier or clock synchronizer loop, can exhibit what is typically called a "dead-zone" if the error/correction relationship does not hold for miniscule errors. This effect is illustrated in *Figure 9* as the less-than-ideal transfer function which effectively removes the phase correction control in the neighborhood of "zero error." This "hole" in the transfer function will cause an otherwise perfectly locked loop to exhibit jitter because the loop will be unable to maintain control and will wander between the two inflection points.

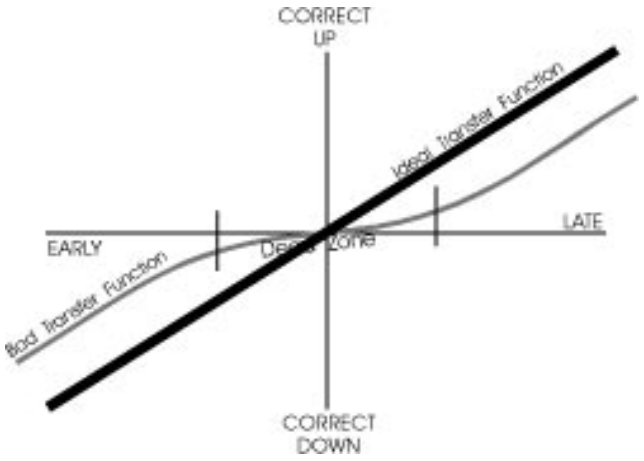

Figure 9. Phase Corrections Should Be Linear with Error Magnitude

HOTLink Transmitter and Receiver PLLs have been designed to eliminate this undesirable behavior.

The closed-loop PLL acts like a Low-Pass Filter to incoming noise (*Figure 10*). All frequency components that fall below the roll-off frequency of this filter are passed unattenuated. Frequencies above the roll-off frequency of the filter are attenuated, and frequencies around this point might be amplified to some extent. Some forms of jitter have low frequency characteristics that will pass through the PLL and appear on the resulting high frequency clock output (e.g., low-frequency wander passes unattenuated through the Receive PLL).

The PLL low-pass filter model is valid for jitter that enters the system at the PLL input. However, jitter that is injected (or is present) *inside* the loop "sees" the loop as a high-pass filter. The dynamics of the closed loop system allow it to compensate for low-frequency injected jitter with an automatic (and opposite) low-frequency phase adjustment. As the frequency of the injected jitter rises toward the roll-off frequency, the loop becomes incapable of fully compensating the injected jitter. Above the roll-off frequency, the loop will pass injected jitter without attenuation (see *Figure 11*).

Since V_{CC} noise is injected (and can result in undesirable delay variations) at multiple points inside the loop, the resulting component of jitter attributable to V_{CC} noise will probably show a peak at approximately the roll-off frequency, and less output jitter at both higher and lower frequencies. The ratio of jitter magnitude for V_{CC} noise frequencies above and be-

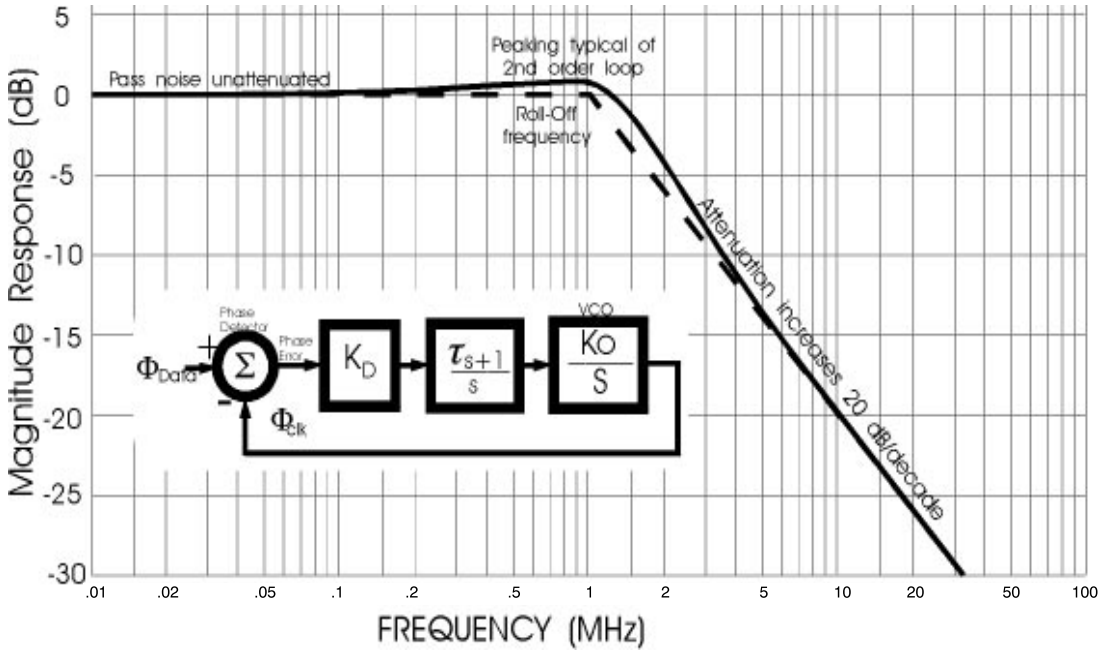

Figure 10. PLL Closed Loop Response is a Low Pass Filter

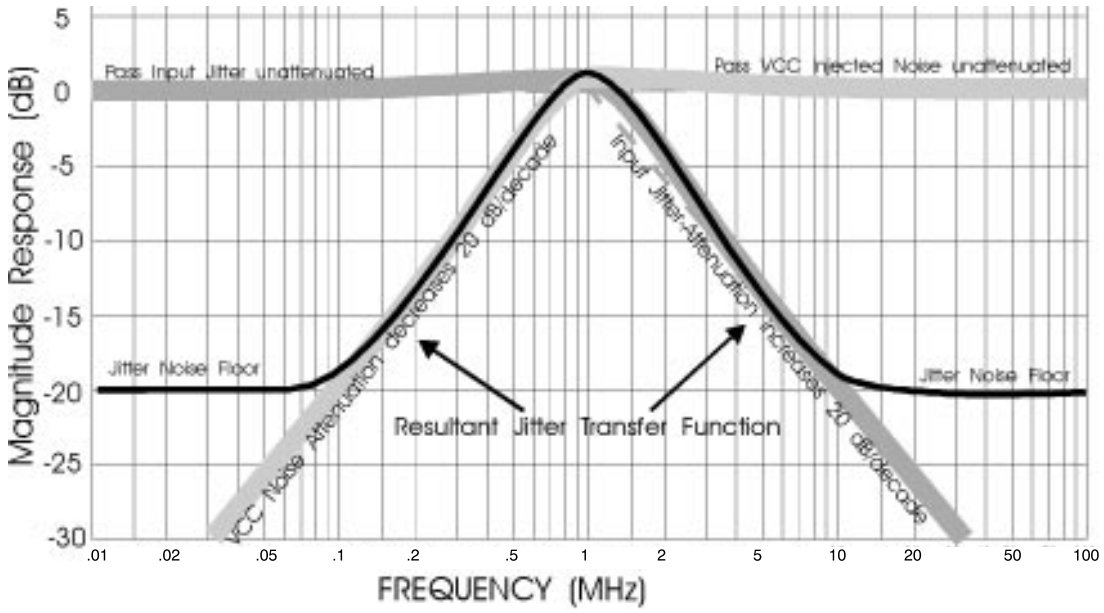

Figure 11. V_{CC} Noise Injection Transfer Function

low the PLL roll-off frequency might give a clue to the probable noise injection point(s). Larger jitter magnitude for V_{CC} noise frequencies below roll-off is probably attributable to logic delays in the input path the PLL (e.g., logic delay changes in gates prior to the loop, passing through the PLL low-pass filter). Larger jitter magnitude for V_{CC} noise frequencies above roll-off is probably attributable to noise injection into some circuit inside the loop (e.g., jitter inside the loop being affected by the PLL as a high-pass filter). Jitter magnitudes that are approximately the same above and below roll-off are probably attributable to delay changes being introduced into circuits that follow the loop.

Jitter on a Multiplier-PLL clock can affect the operation of a system in several ways. The most obvious effect is cycle-to-cycle jitter (high-frequency or random effects) that causes a reduction in clock-to-clock spacing and affects logic timing as was discussed previously. Another undesirable effect is the longer term clock wander that can cause problems in a system when two parts of the system are clocked by different clocks of the same frequency but with a variable phase relationship.

Jitter can affect a Clock-Synchronizer PLL in at least two ways. First, the jitter in the PLL clock directly reduces the jitter tolerance of the Clock/Data separator. Jitter tolerance is a measure of design margin in a serial communication link. Second, the jitter of the PLL clock is transferred directly to the receiving host system clock. This jitter-reduced timing margin affects the system in exactly the same way that Multiplier PLL jitter affects logic timing.

Bit Error Rate

Jitter is often directly equated with Bit Error Rate (BER). While it is true that jitter accumulation on a serial link is the primary determinant of the link's reliability, its statistical character makes it difficult to understand and impossible to accurately predict. Since PLLs also exhibit a statistical nature which must be added to the jitter of the link, some designers incorrectly assume that the PLL jitter is what causes the errors. In fact, the PLLs used to create the high frequency clocks in the transmitter and receiver serve to increase reliability rather than decrease it.

BER is a term which is common to both serial communication devices and communication systems, and could also be applied to any system where data moves from one storage location to another. BER is the ratio of "corrupted data" received to "good data" sent. Sometimes BER can mean "Byte Error

Rate" in systems that transmit multi-bit wide information. Errors that affect the fill-bits (the "non-information" bits that occupy the time between actual data transfers) are not usually counted toward BER, but might be used to predict overall system margins.

$$BER = \frac{Bits\ in\ error}{Bits\ transmitted}$$

Usually BER is expressed as a large, negative exponent (e.g., 2.5 x 10^{-12} or 1 x 10^{-9}) because acceptable systems perform almost flawlessly. For example, a 250 Mbaud system operating with a BER of $1x10^{-12}$ would only experience about one error per hour while sending continuous information (see *Figure 12*).

While one error per hour may seem excessive, it comes naturally from the technology of the serial interconnect used. Even when adequate margins are designed into the link, there are physical and electrical effects which will cause occasional errors. Communication systems have been engineered to accept this error rate by including some level of error checking in the physical layer hardware, and extensive error recovery built into most of the communication protocols.

The PLL(s) used in the communication link don't cause the failures. Their ability to recover timing information from a severely distorted serial interface is one reason that these types of links are possible at all. The PLL supplies the logical clock necessary to correctly process the data. Without it the difficulty and cost of aligning data and clock across any distance would be immense. But because of the designer's inability to predict to the exact statistical nature of the interconnect link, and because PLL jitter contributes to this uncertainty, PLLs are often wrongly equated with BER.

HOTLink CY9266−C Evaluation boards operating at 250 Mbaud with a short coax link have been tested continuously for over 4000 hours without ***any*** errors. During this time, over $3.6x10^{15}$ bits were sent, received, and checked by the HOTLinks using the BIST function. This error-free time yields an estimate of BER less than $8x10^{-16}$ with greater than 95% confidence. Link error rate is not impaired by the addition of a deterministic interconnect link, such as a long coaxial cable. HOTLink CY9266−C Evaluation boards running the BIST test at 250 Mbaud, and interconnected with 300 feet of RG−59 coaxial cable (90% of the maximum uncompensated distance) have operated with no errors during a 1500 hour test. This shorter test still yields an estimate of BER less than $2x10^{-15}$ with greater than 95% confidence. No error has *ever* been recorded in all the time spent testing HOTLinks with deterministic links (sensitivity to environmental-noise injection was not included in this test). Actual BER rates have never been determined because after an uneventful six month test, the tests were terminated to free the equipment for other uses.

Example: Bits transmitted $= 250 \times 10^6 \frac{bits}{sec} \times 3600 \frac{sec}{hour} = 9 \times 10^{11} \frac{bits}{hour}$

$$BER = \frac{N}{9 \times 10^{11}} = N \times 1.11 \times 10^{-12}$$

MTBF of a link running at 250 Mbaud @ BER of 1×10^{-12} = *1 hour between errors*

MTBF of a link running at 250 Mbaud @ BER of 1×10^{-15} = *46 days between errors*

MTBF of a link running at 250 Mbaud @ BER of 1×10^{-18} = *127 years between errors*

Figure 12. BER Example Calculations