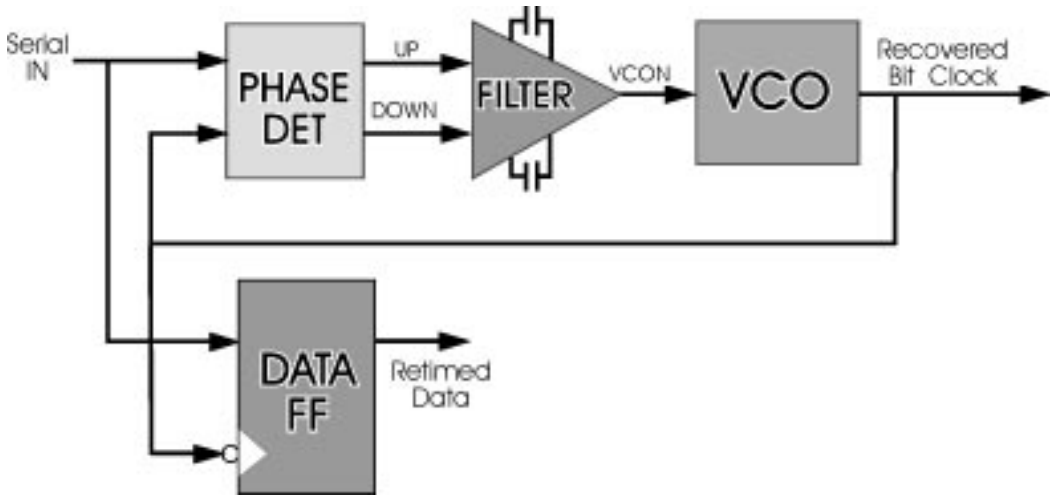


Figure 29. HOTLink Receiver PLL Block Diagram

HOTLink Receiver Jitter

The PLL used to synchronize an internal clock to a received bit stream (i.e., in the HOTLink Receiver) has different requirements than those for a multiplying PLL. This loop is effectively a one-to-one loop where the bit clock (Received Bit Clock, an internal signal) runs at the same rate as the incoming data stream (Serial IN, an external signal). The Received Bit Clock is used to sample the Serial input at regular intervals, thus extracting the serial data (Retimed Data, in *Figure 29*). This same signal runs all of the internal logic for deserializing, framing, and decoding the serial data. Any disturbance that can affect the PLL and the Recovered Bit Clock will affect both the quality of the data recovery and the quality of the byte-rate, data-synchronous clock that is provided to the receiving system.

Receiver jitter affects systems in at least two ways. Jitter tolerance is a major determinant of system margin, and Jitter feed-through can reduce timing margins in the receiving host system.

Jitter feed-through is a function of the PLL filter characteristics, and can be directly measured at the CKR output of the HOTLink Receiver in much the same way used to test Transmitter jitter feed-through.

Jitter tolerance is more complicated, since it is a measure of the Receiver's ability to correctly capture and interpret incoming data, and must be measured indirectly. Jitter tolerance is both a function of the intrinsic jitter in the receive-clock synchronization PLL and the effects of received data upon it. Tolerance is also a function of the precision-timing and alignment of internal clock edges (i.e., the clock edge used in the PLL to synchronize the data, and the clock edge used to sample the incoming data stream). The data-sampling flip-flop set-up/hold timing characteristics and their variation contribute to further jitter tolerance degradation.

To isolate the effects and tolerance limits to various types of jitter, carefully designed tests were performed on HOTLink parts selected from the full spectrum of manufacturing variation. These tests were designed to separate the effects of power supply, data characteristics, external clock sources, and various PLL characteristics. Unless otherwise noted, static variations in power supply levels (4.5V to 5.5V), ambient temperature (−55°C to 125°C), and process variations (within manufacturing tolerance limits) cause virtually no change (within the accuracy of the measurement system) to any of the following jitter tolerance or PLL characteristics.

Static Alignment and Error-Free Window

To maximize jitter tolerance, the receive circuit is designed to sample the incoming data at a point *exactly* half way between the ideal transition times of uncorrupted data. This requires that the PLL track the incoming data and align itself with the "average timing" of the received edges. The precision of this

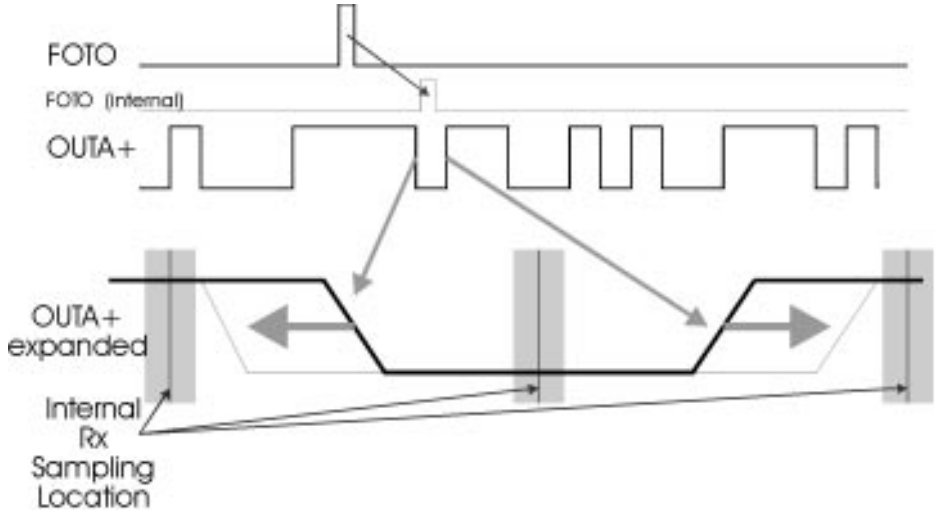


Figure 30. Technique to Measure Static Alignment

alignment is often called "Static Alignment" and should have a magnitude of zero, indicating perfect alignment of VCO and the data and perfect 50% sampling alignment. Using this recovered clock, the incoming data is sampled at the point that gives maximum tolerance to misplaced edges and maximizes the error-free window. Any misplacement of this sampling point will reduce jitter tolerance.

Static alignment of the HOTLink Receiver was evaluated using the technique shown in *Figure 30*. The HOTLink Transmitter and the Receiver under test were configured to send and receive the BIST pattern. Then, by inserting a BIST-synchronous pulse on the FOTO pin (using a generator triggered on the $\overline{RP}$ output of the HOTLink Transmitter), one transition in the transmitted data pattern was varied to find the maximum "misalignment" possible before the onset of an RVS error indication. This configuration allows the receive PLL to have about 3000 "ideal" transitions (i.e., the total number of transitions in the 511 byte BIST loop) and only one misplaced edge. Shorter patterns modified in this way (e.g., a single data byte with byte-synchronized FOTO pulses having a single misplaced transition) give an erroneous result. The very large phase error which occurs in one of the ten bit positions will be averaged out by small-compensating phase-adjustments during the other nine bit-times. The BIST pattern test allows the PLL phase-correction response from the single-edge error to settle out before the next error appears so that the averaging effect does not color the data-capture results.

Data transitions can be misplaced from their ideal position by almost half of a bit-time without erroneous sampling by the data recovery flip-flop. The data characterization summary in *Figure 31* indicates that the HOTLink Receiver will accept misplaced edges to within about 250 ps of the half-bit point. The center of the small error region where data is not sampled correctly (at approximately 180 ps after the ideal mid-bit point) is the actual PLL static alignment position. The width of the error region (about 150 ps) is attributable to both the sampling flip-flop metastable region, and the internal PLL clock jitter.

This data alone implies that any data edge could fall anywhere within a bit time (minus about 500 ps) and still be decoded correctly. This is almost correct, except for the effect of receiver clock jitter caused by the various types of incoming jitter.

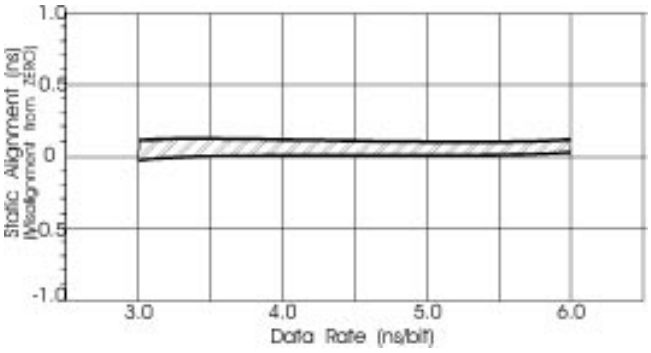


Figure 31. HOTLink Receiver Static Alignment as a Function of Frequency

Duty Cycle Distortion Jitter Tolerance

The characteristics of some types of interconnect circuits cause Duty Cycle Distortion which the receive system must tolerate. DCD jitter alters the placement of all transitions in the data stream by about the same amount (in alternating directions) regardless of the bit pattern being sent. For small amounts of jitter, this alternating error tends to cancel out, and the loop behaves normally while recovering data without error.

As the magnitude of jitter increases, phase correction pulses from adjacent misplaced edges will begin to interact. Each correction pulse has some finite duration, usually a significant percentage of the expected bit time, and is proportional to the magnitude of the edge misplacement. Since jitter is also expressed as a percentage of a bit (usually a large percentage) the interaction between jitter magnitude and phase correction pulse width will determine DCD jitter tolerance. When adjacent phase corrections interact, they sum in unexpected ways which affect the resulting correction response. When these interactions are rare or small, there is no apparent effect. If the interactions affect most of the phase correction events, the PLL stability, predictability, and output jitter will be affected and data will not be captured correctly.

Figure 32 shows HOTLink Receiver DCD jitter tolerance. This test was performed by carefully corrupting the link between a HOTLink Transmitter and Receiver with increasing magnitudes of DCD (See Jitter Generator circuit and description *Figure 49*). Using the BIST test capability included in the chips, DCD tolerance limits were declared to have been exceeded when the RVS output of the Receiver indicated approximately one error every ten seconds (i.e., BER$\leq\approx 4x10^{-10}$ at 250 Mbaud). Slight differences in jitter tolerance were found between parts from different process corners, but no appreciable variation was found for V_{CC} or temperature variation. The DCD tolerance characterization data shown above varies by less than 5 percentage points across the full process spread (e.g., from 1.42 ns to 1.39 ns out of a 3.0 ns bit time). The threshold of failure is very abrupt. At the jitter levels shown in *Figure 32*, changes in jitter amplitude of less than ±100 ps make the difference between almost-perfect data reception, and almost-total corruption.

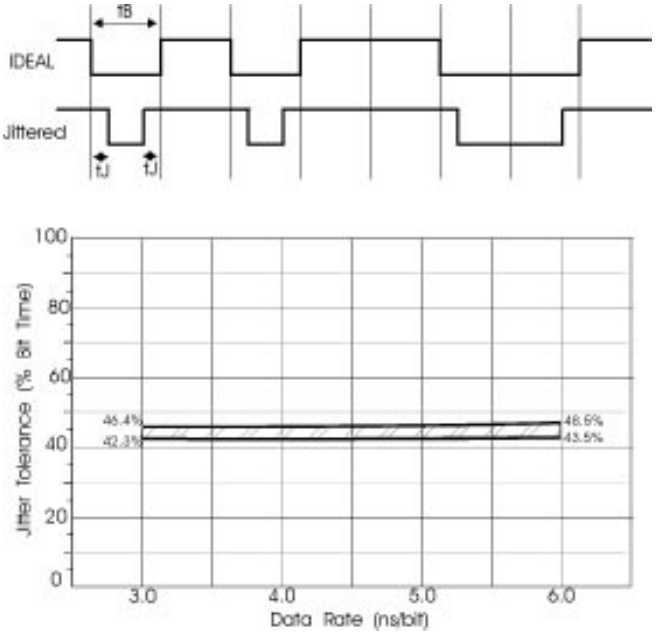


Figure 32. Duty-Cycle-Distortion Jitter Tolerance as a Function of Data Rate

In contrast to the predicted jitter tolerance that comes from the Static Alignment test, and the DDJ tolerance (see following text), DCD tolerance at first appears to be much smaller. This apparent reduction in jitter tolerance is entirely due to PLL and Phase-Detector effects, and do not result from any anomaly in the data recovery path. The data can be recovered correctly at the levels of edge misplacement that are found at the limits of DCD tolerance but not above. By carefully approaching the limit, it can be seen that the PLL loses lock at the jitter magnitudes shown in *Figure 32*, and then regains it at slightly higher jitter levels, but with a massive clock jitter, often slipping bits as the jitter goes through the "magic point," destroying any data recovery possibility. The recovered clock shows almost no jitter feed-through when DCD is present and remains below the "data-corruption" threshold, as will be shown later (*Figure 46*).

Fortunately, most transmission links don't include large amounts of DCD. The most common contributors are mismatched output loads on differential or single-ended PECL outputs, and improperly designed or operated optical interface modules. Single-ended PECL outputs can change the effective delay of the driver by about ±0.5 ns. Differential outputs are typically more symmetrical. Opti-

cal-to-electrical (receiver) interface modules running with extremely high or low light levels can have non-linear and asymmetrical delay characteristics that affect the pulse symmetry of the received output used by the PLL data recovery circuits. The optical emitter in an electrical-to-optical interface module also has non-symmetrical turn-on and turn-off characteristics which are normally compensated by careful design of the drive electronics. At the limits of performance, optical modules can add more than ±1 ns of DCD.

Data Dependent Jitter Tolerance

The characteristics of some types of interconnect circuits cause Data Dependent Jitter which the receive system must tolerate. The same "correction-pulse" interaction that limits DCD tolerance also affects DDJ tolerance. Since the collisions between adjacent correction pulses occur at a much less frequent and regular rate, the effect is smaller. The "clock-jitter" that results from these corrupted corrections reduces the jitter tolerance to less than the ideal maximum that the Static Alignment test might predict.

Figure 33 shows HOTLink Receiver DDJ jitter tolerance where the DDJ was generated by an artificial

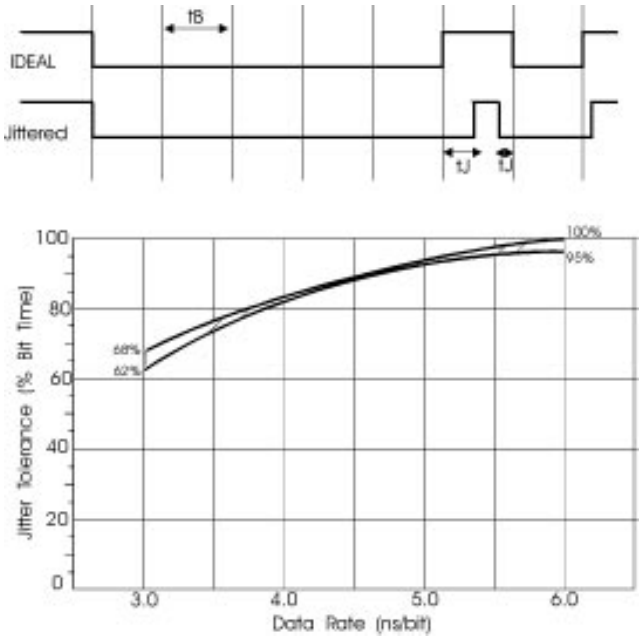


Figure 33. Data-Dependent-Jitter Tolerance as a Function of Data Rate

generator. This test was performed by carefully corrupting the link between a HOTLink Transmitter and Receiver with increasing magnitudes of DDJ (see Jitter Generator circuit and description in *Figure 50*) while sending a continuous BIST pattern. Errors were most typically associated with the long running bit pattern included in a K28.5 bit pattern, and the same tolerance was observed while receiving *only* corrupted K28.5s. The worst DDJ peak always follows the 1111101 and the 0000010 contained in the special characters. Using the BIST test capability included in the HOTLinks, DDJ tolerance limits were declared to have been exceeded when the RVS output of the receiver indicated approximately one error every ten seconds (i.e., BER $4x10^{-10}$ at 250 Mbaud). Slight differences in jitter tolerance were found between parts from different process corners, but no appreciable variation was found for V_{CC} or temperature variation. The DDJ tolerance characterization data as shown in *Figure 33* varies by less than 5 percentage points across the full process spread (e.g., from 2.04 ns to 1.86 ns out of a 3.0 ns bit time). The threshold of failure is very abrupt. At the jitter levels shown above, changes in jitter magnitude of less than ±100 ps make the difference between almost-perfect data reception, and almost-total corruption.

Interconnect Link Jitter Tolerance

The tolerance to synthetic-DDJ shown in *Figure 33* is slightly worse than that found when the jitter is natural-DDJ. The variation is caused by unintentional DCD introduced by the test system used to create a stable and repeatable test pattern at all frequencies over which HOTLink might operate. Wire transmission line jitter is dominated by DDJ caused by the variation in attenuation as a function of frequency. Higher frequencies are attenuated more than lower ones. This rising attenuation-with-frequency characteristic of wire links causes the wider pulses (i.e., multi-bit one or zero strings) to have a higher amplitude than the shorter pulses since the higher frequencies (those attenuated the most) are required to make the fast edges and narrow pulses, while the wider pulses contain more low-frequency components. This variation in amplitude results in variations in pulse placement, since the edge rate is almost constant and the variation in amplitude causes variations in the time at which a transition will cross the receiver threshold.

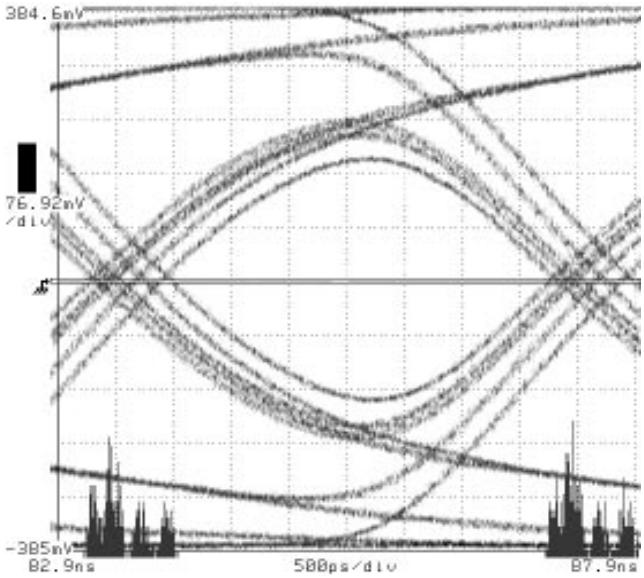


Figure 34. DDJ Characteristic of K28.5 at 250 Mbaud after 250 ft. RG−59

This effect is most visible when a single, worst-case data byte is measured. *Figure 34* shows the edge misplacement caused by the different-length pulses in a continuous K28.5 pattern (i.e., 11000001010 00111110101...). When the data is more normally distributed, it becomes more difficult to see the distinct pulse positions, and the jitter just merges into a continuous "uncertainty-zone" (see *Figure 35*).

Using actual data and real transmission lines, the HOTLink tolerance to DDJ appears to be a more constant function of bit rate than *Figure 33* shows. If about 500 ps of clear eye-opening can be maintained, the data will be recovered correctly, regardless of the data rate. However, recovered clock jitter increases with increased DDJ (see *Figure 47*).

In wire transmission links, the accumulation of DDJ determines the maximum distance over which data can be reliably communicated. The characteristics of the chosen media determines the useable distance. The total attenuation of the line is rarely sufficient to limit the maximum useable distance, even though the data bits that are incorrectly interpreted will have minimal amplitude at the time of the error. This loss of amplitude is a result of the variation in peak voltage attained during any particular pulse.

HOTLinks have been designed to offer more than 20 dB of attenuation margin between the transmitter output and the receiver input. Typical maximum-distance links have less than 10 dB of high frequency attenuation due to the transmission line and interconnect components. The remainder of the interconnect budget can be used to compensate for the difference between high and low frequency attenuation of the wire transmission line. Compensated wire links have been built that operate reliably over more than double the distances shown in *Figure 36*.

Fiber-optic links, in contrast to the wire links described above, are limited by optical attenuation, chromatic dispersion, and the resulting Random Jitter in the optical-electrical converter. At the limit of operational optical margins, the low light levels into the receiver and the dispersion from the fiber combine to create misplaced data transitions. These displacements are usually random, but in the case of some optical modules, can also include significant Duty Cycle Distortion.

Peak random jitter tolerance *should* be approximately the same as the Static-Alignment limits described above (*Figures 30* and *31*). The simplest way to generate random jitter involves a long piece of fiber-optic cable, and appropriate fiber-optic interface modules. As fiber length increases, adding chromatic dispersion (i.e., pulse distortion caused by the variations in propagation delay through the

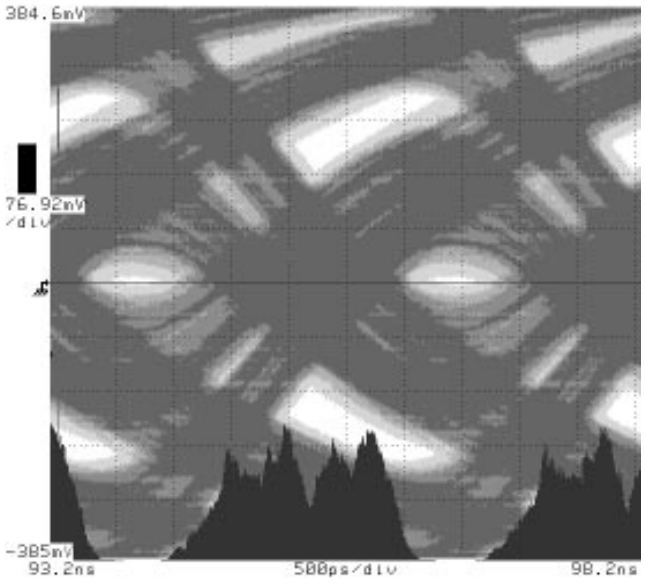


Figure 35. BIST data at 370 Mbaud after 250 ft. of RG−59 coax (BER<4.5x10^{-11} with <700 ps eye opening)

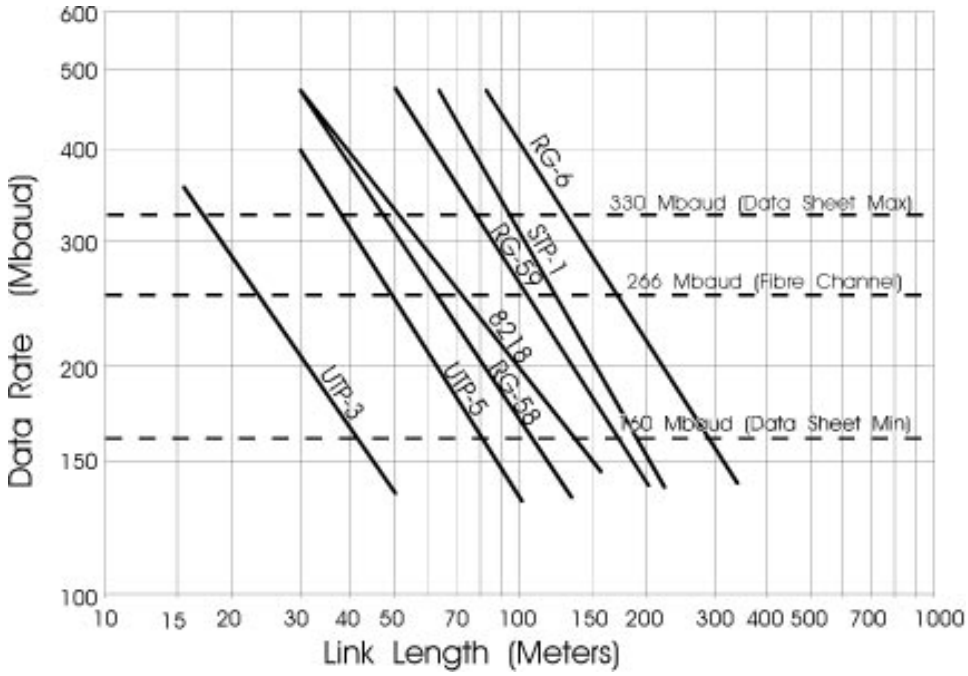
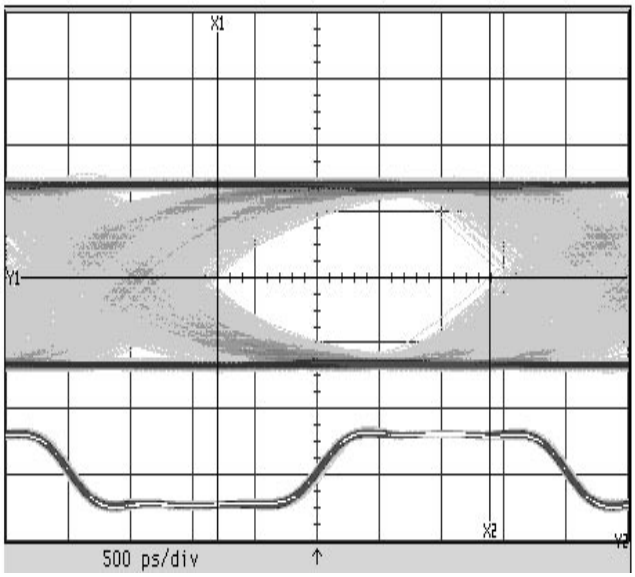


Figure 36. Maximum Data Rate vs. Uncompensated Wire Length (BER < 3 x 10^{-12})

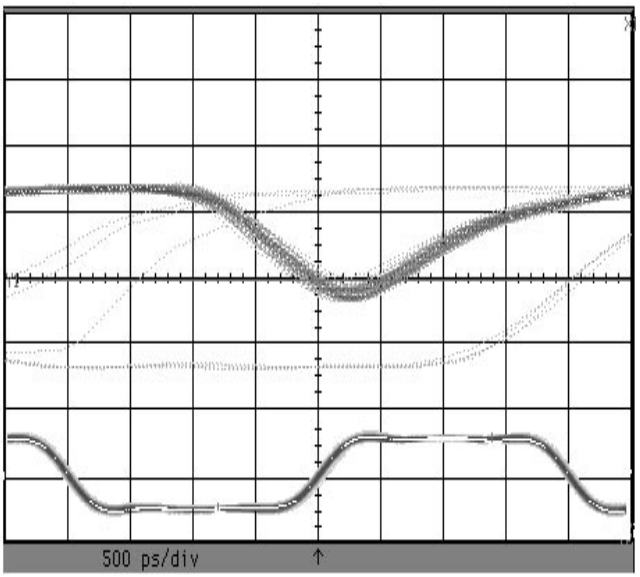
fiber, as a function of optical wave-length) and attenuation, the jitter out of the optical-to-electrical converter will increase. There is a limit to attenuation, beyond which the fiber-optic receiver cannot recover the data correctly. Attenuation alone, without the effects of long fiber-optic cable, often causes significant DCD in the link. This DCD will obscure the real random jitter behavior of the receiving PLL.

The random jitter output of a 5-km piece of 62.5 multi-mode fiber is shown in *Figures 37* and *38* and



Bit time = 4.0 ns
Eye opening = 2.185 ns (apparently)
BER = 1 x 10^{-9}

Figure 37. Random Jitter out of Fiber-Optic Link Triggered by Bit Clock



Bit time = 4.0 ns
Eye opening = < 100 ps
BER = 1 x 10^{-9}

Figure 38. Random Jitter out of Fiber-Optic Link Triggered by RVS & Bit Clock

illustrates a typical problem that occurs when trying to measure random jitter and jitter tolerance. These photos were taken at the limit of frequency/length as indicated by BIST errors appearing on RVS. The first "eye-diagram" (*Figure 37*) was taken using the traditional infinite-persistence scope measurement, where the scope is triggered by a pristine bit-clock. The trigger-clock, shown below the eye-diagram for reference, is arbitrarily placed with respect to the jittered data trace. This is the resulting display of an HP54720D at 8 Gs/s after about four hours of jitter accumulation (approximately 30,000 traces). It would appear that the jitter tolerance of the receiver is only about 45% (i.e., 4.0 ns − 2.19 ns) at the measured BER. *This conclusion is incorrect.*

Figure 38 offers another view of the same link and error rate, when triggered by the error event and shows the *actual* eye opening. This view, triggered by the pristine bit-clock qualified by RVS (ANDed), shows that when the HOTLink indicates an error event, the "eye" is actually fully closed. This photo displays only those traces that contained an error event, about one every four seconds at 250 Mbaud. It is impossible to determine from these photos exactly where the PLL and the data sampling flip-flop have placed the bit boundaries, but it is obvious that if the transition doesn't cross the threshold, the data is lost. (The "ghost" traces that appear in the photo are parts of other error-traces where the eye-closure occurred at some other bit position beyond the limits of the screen.) The discrepancy between these two figures is caused by the triggering and display characteristics of the scope. Even though there are over 30,000 patterns displayed on the first one (*Figure 37*), it just happened that none of the error bits were captured. This could have been because of the relative rarity of the events, and the trigger hold off caused by the scope processing that occurs between measurements.

Receiver Data-Phase Acquisition Time

To measure the HOTLink Receiver response to phase-hops in the incoming data stream, it is necessary to produce a data stream that has a controlled phase change. It is possible to use the two selectable inputs of the HOTLink Receiver to switch between two identical, but skewed, data streams. The data stream used for these tests comes from a HOTLink Transmitter using a good quality clock source. The HOTLink BIST function provides a convenient source of repeatable data and is accompanied by a convenient trigger pulse in the $\overline{RP}$ output that occurs once per BIST loop. The Receiver BIST comparator can be used to determine whether the receiving PLL has maintained phase lock without slipping by monitoring its RVS output. This output will pulse only if there is an error in the received data pattern.

In the test set-up shown in *Figure 39*, the input to the INB+ pin of the Receiver is skewed with respect to the INA± input using the precision skew capability of the Colby delay generator, which can add delay up to 10 ns in 1 ps increments. A carefully placed control pulse (i.e., inputs are changed only when both inputs will be staying at the same logic level for a few bit times to insure that the change does not affect the serial data stream), which is a BIST-synchronous control signal (i.e., the pulse is triggered by $\overline{RP}$ which occurs once in each BIST loop), switches the receiver input between the two data streams. As expected, when the A/$\overline{B}$ input switches between these two streams, no errors are indicated if the skew is small. When the skew is increased, and approaches almost half of a bit time (i.e., 135 to 150 degrees as seen by the PLL Phase Detector) errors are indicated by pulses on RVS. These errors are caused by "bit-slip" in the PLL as it reacquires the new data stream.

By triggering the HP54120D on the signal that changes data streams, it is possible to observe the real-time behavior of the receiving PLL. The scope can be programmed to measure either clock period, or propagation delay between two channels. The former will show each clock period as the loop acquires the new data stream. The latter set-up will show the more traditional phase-alignment measurement that defines Phase-Lock-Loop acquisition characteristics.

Measurements were taken with various amounts of phase difference between the two input channels. The figures that follow show the characteristics of the HOTLink Receiver with phase errors less than 180 degrees and with phase errors at as close to 180 degrees as possible. The first kind illustrates typical link performance. The second kind shows the worst case phase acquisition characteristic.

In the test set-up shown in *Figure 39*, the HOTLink Receiver input is switched to one of the inputs, allowed to stabilize there for a few byte times, and

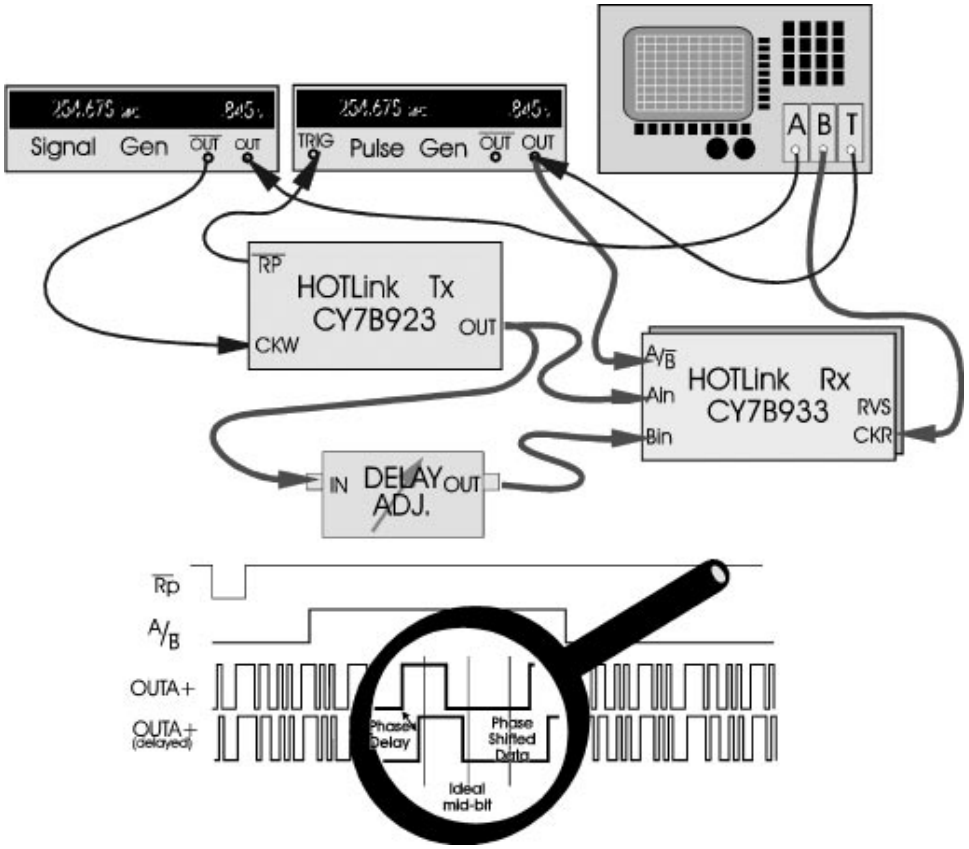


Figure 39. Set-Up to Measure HOTLink Phase Acquisition Characteristics

then switched back. The second switch is an equal phase offset, but opposite sign. During the time when the PLL is trying to regain phase alignment with the incoming data stream, it adjusts the period of the VCO, and thus the output clock of the HOTLink Receiver. As illustrated by the data shown in *Figure 40*, the phase correction begins immediately after the change in data stream. Since the phase error is less than 180 degrees, the correction is always in the expected direction. When the new data stream "lags" the current PLL position, the clock is stretched for a few cycles until it realigns with the incoming data. Likewise, when the new data stream "leads" the current PLL phase, the clock is shortened for a few cycles until it realigns with the incoming data.

The change between any pair of clock (CKR) periods is small, and the maximum deviation usually varies by less than ±1 ns midway through the seven to ten byte-times required to realign the clock. The magnitude of change that can be accommodated without error varies slightly with frequency, and the time needed to resume normal clock periods varies by one or two byte times. There is little or no correlation between settling time and the sign of the phase-change, data-speed, process-corner, V_{CC}-level, or ambient-temperature. For all frequencies, it seems that any phase change that is less than a half-bit time (less about 500 ps) will be accommodated without data corruption. The Byte-Clock adjustment shown in *Figure 40* is the accumulated sum of the ten Bit-Clock periods that combine to makeup the Byte-Clock adjustment, each of which was probably much smaller.

When the phase change is carefully adjusted to the 180 degree position, the correction behavior changes. The correction can be in either direction, since both have an equal capability to realign the PLL clock phase. One direction will cause a bit slip since the decoding logic will find the data appearing one bit earlier or later than expected. The other direction might not slip, but will probably still indicate

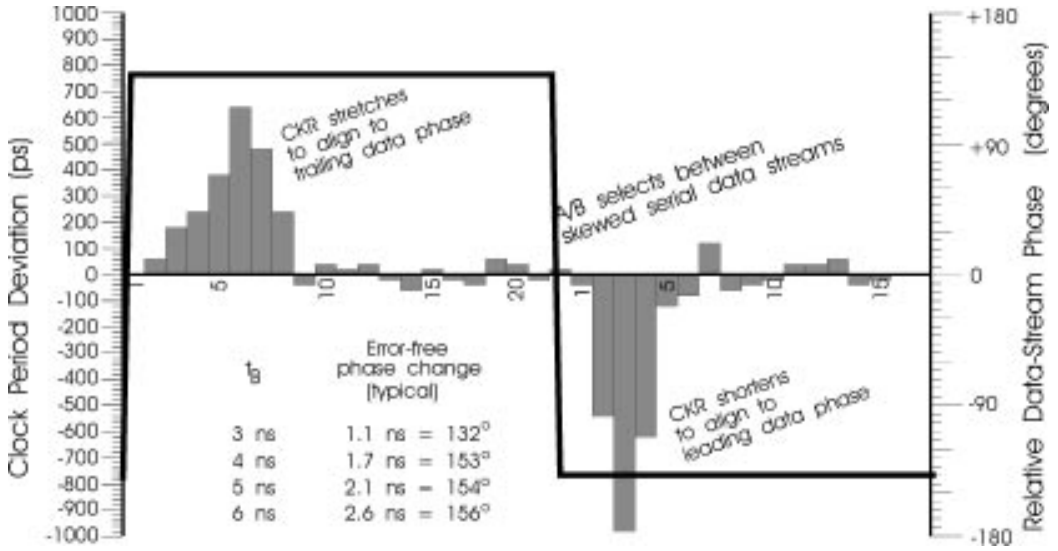


Figure 40. Phase Hop of less than 180 degrees without Data Corruption

a corrupted byte because of a metastable response from the data sampling flip-flop.

Additionally, the phase correction does not start immediately after the change in incoming data phase (see *Figure 41*). The time it *might* take cannot be calculated, because the loop is operating outside its linear response region, and will assume some metastable behavior that could theoretically take forever to clear. It takes several byte times before the PLL accumulates enough error information to cause it to realign itself. When the data has exactly 180 degrees phase offset to the PLL VCO, the Phase Detector may have either no phase-correction effect or a small reverse phase-correction effect, in contrast to its normal, increasing-correction with increasing-error, linear-phase-correction response to smaller phase errors. Once it begins to change, the PLL completes the phase hop in about the same way as the earlier example showed, although over a slightly longer duration. Perhaps counter to intuition, the quieter the received data stream, and the cleaner the VCO clock, the longer this "hang time" will become. (Products with "jitter problems" will never exhibit this "hang phenomenon.") Any jitter or frequency deviation between the incoming data and the VCO provides a tie-breaker and gives enough error information to allow the Phase Detector to begin its change. Once the relative-phase has moved only a little bit, it becomes obvious to the Phase Detector that the error is large and requires a large correction. Complete phase alignment is not

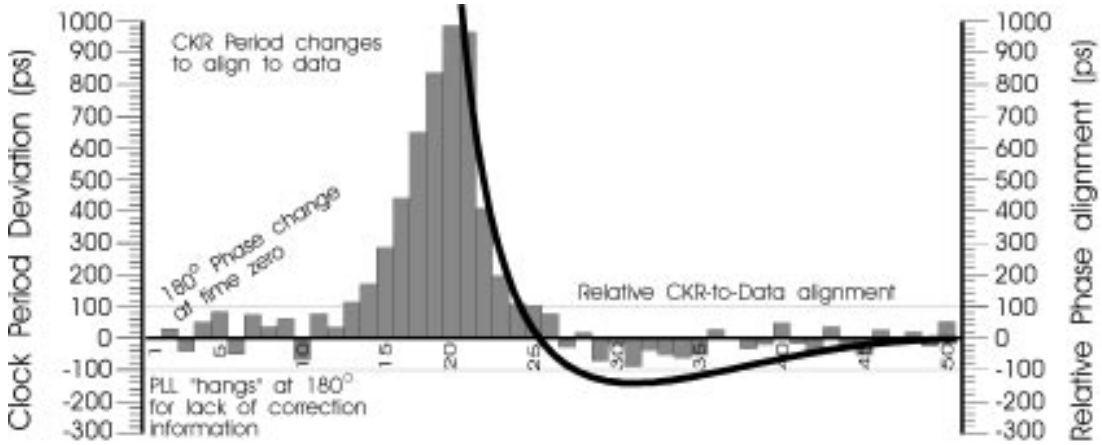


Figure 41. Phase Hop Timing with Exactly 180-Degree Phase Difference

achieved until several byte times after the CKR output has resumed its normal period.

This final alignment time is immaterial for most data-communications systems, since the receiving system will have long since resumed correct data recovery because of the wide jitter tolerance of the receiver. As was shown in the limited phase-step experiment, the HOTLink Receiver will recover the *bit-stream* correctly when the input transitions are more than about 500 ps away from the mid-bit point. However, the *data* must be "framed" to be interpreted correctly, and the time necessary to accomplish framing (HOTLink requires one or two K28.5 characters to frame, depending on current framer mode) will depend on the protocol being used, and how often SYNC characters occur.

Receiver Data-Frequency Acquisition Time

Two serial data streams rarely operate at exactly the same frequency, so the PLL must first acquire the new frequency before it can align the clock to the phase of the new data. The frequency offset that must be accommodated is different for each standard system, but is usually a few hundred parts per million (PPM) variation from a specific frequency. Fibre Channel, for instance specifies a maximum frequency offset of ±100 PPM (±0.01%). HOTLink is specified to accommodate frequencies of ±0.1%, but will typically accommodate more. The chart in *Figure 42* is an illustration of the behavior of a HOTLink receiver as it switches between two data streams that are offset from the local REFCLK frequency by ±0.2%. The acquisition time and the effective clock-period transient time is equivalent to that seen when the receiver is only adjusting for phase differences.

This is typical of the HOTLink acquisition behavior at all operating frequencies, and doesn't vary significantly across process, V_{CC}, or temperature variations. The exact transient size and duration will vary from event to event because of the statistical nature of the "first-change." The excursion could be either to a shorter or a longer clock period depending upon the perceived phase of the new data when the change occurs. Likewise, the time to achieve phase alignment will vary slightly depending on the probability of having a perfect 180 degree phase alignment after the change. None of the clock period excursions measured during this test exceeded ±1.0 ns.

When the data stream is not within the frequency tolerance limits of the receive PLL, the HOTLink automatic frequency-range-control mechanism will affect the transient behavior of CKR. This function continuously monitors the frequency of the VCO and compares it to the frequency of REFCLK. When they are different by a sufficient amount, and for a sufficient time, internal logic will force the VCO to align to the REFCLK frequency. This automatic mechanism insures that the absolute frequency of CKR is never far from its ideal period, and that

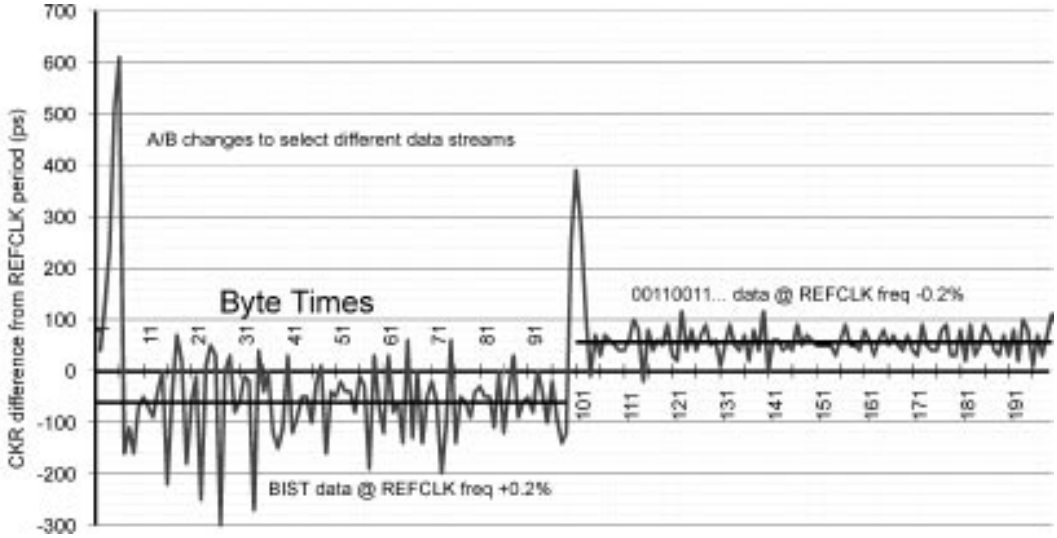


Figure 42. Frequency Hop Within ±0.2%

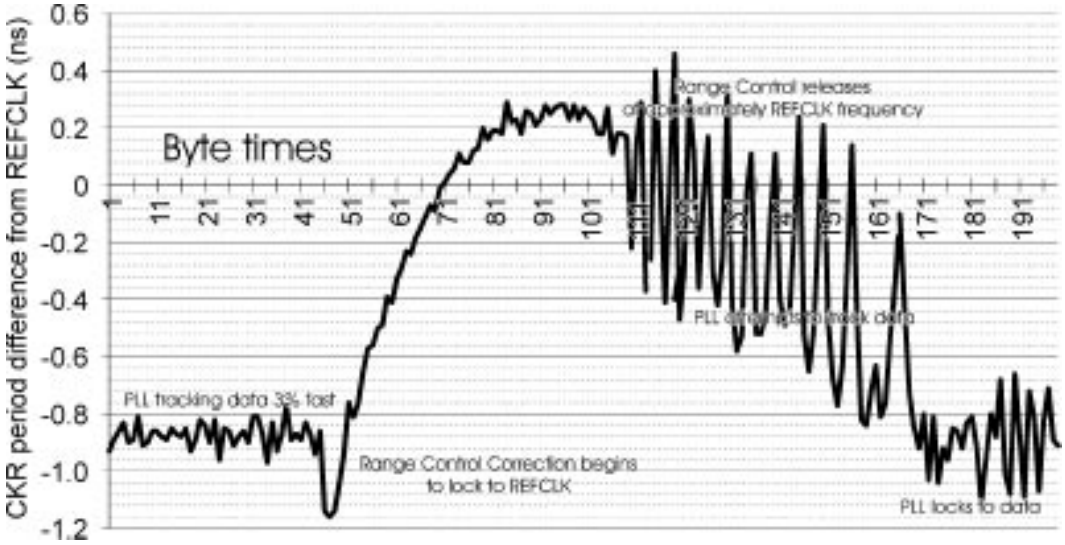


Figure 43. Frequency Acquisition from/to +3.0% Changes CKR Period

when "good" data returns, the PLL will be able to rapidly align to it, and begin correct data recovery.

This test is similar to the previous one, except that the data is not switched externally, and the recorded transient is only the result of the internal re-lock to REFCLK behavior. In this case (shown in *Figure 43*), the applied data stream was offset from REFCLK by about 3.0% (well beyond the data sheet limit of 0.1%). The HP54720 was "glitch triggered" when RVS was HIGH for >60 byte times. (RVS-HIGH for 64 byte times is the PLL out-of-lock indication, since normal data will not yield continuous error indications.)

For the first few bytes (out to about Byte-time 45 in *Figure 43*), the average period of CKR is about 3% faster than the expected 30 ns which indicates that HOTLink has been successful in acquiring the data frequency. When the built-in automatic range control is asserted, there may be a momentary transient in the CKR period caused by the phase and frequency of the PLL relative to the instantaneous bit-stream phase. Next, the VCO will be pulled to the frequency by the internal range-control logic (from about Byte 45 to about Byte 110 in *Figure 43*). Finally the PLL is released to track the incoming data, whereupon it might immediately return to the previous frequency (the frequency of the incoming bit-stream, if any), or as in this illustration, hunt around for an indeterminate time (maybe an indefinite time) until it again finds a signal within its acquisition and tracking range. The exact PLL behavior will depend on the frequency, transition density, timing characteristics and stability of the applied data stream.

CKR period excursions are slightly larger when this range control mechanism is applied, but still under about ±1.2 ns. The period of CKR is the sum of all Bit-Clock periods that occur between CKR transitions.

Receive PLL Jitter Transfer Function

PLL jitter, and consequently recovered clock jitter, can be affected by the noise characteristics and stability of the incoming data stream. The closed-loop transfer function of the PLL is a low-pass filter. Noise components below the natural frequency (f_n) of the PLL will be passed unattenuated and those above f_n will be attenuated. By injecting a measurable and controlled amount of noise (jitter) into an otherwise stable data stream as shown in *Figure 44*, the PLL transfer characteristic can be measured.

In this configuration, the noise source is added to the data-clock source by a resistive mixer, similar to that used for transmitter jitter-transfer testing. The mixer output drives the external bit-rate clock input of a high speed data generator. The Microwave Logic GigaBERT 1400 can run with clock rates above 1 GHz, and can send serial data from an internal memory using this clock. By jittering the external clock, it is possible to create a controlled seri-

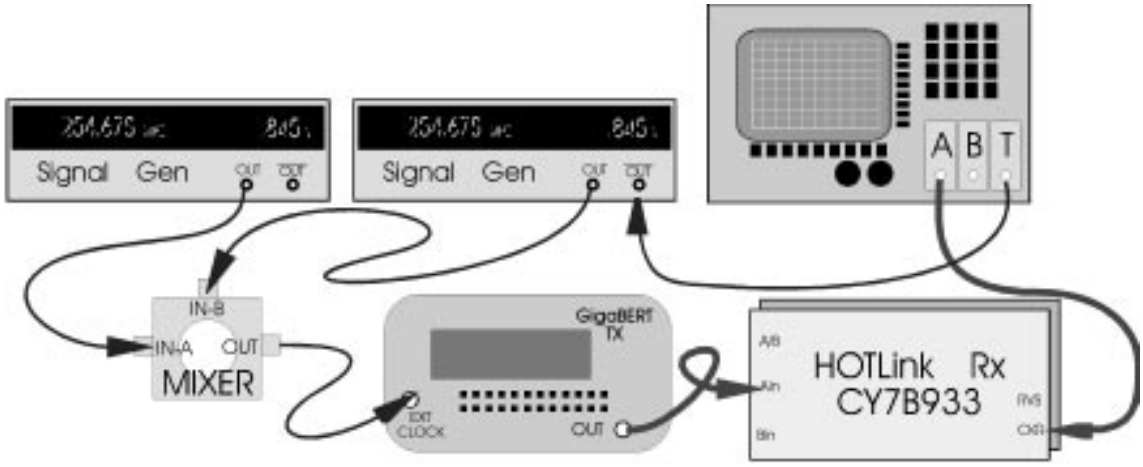


Figure 44. Data-Jitter is Generated by Mixing Noise into Serial-Data-In

al data stream with single frequency jitter noise. The amplitude of input jitter was adjusted to create the desired data jitter amplitude (ns Pk−Pk), and the frequency was varied over a wide range while the jitter was monitored on the CKR output.

Direct jitter generation is difficult to manage because of the need for a single frequency noise source superimposed on an otherwise perfect data stream. Most jitter generators seem to generate either multiple frequency noise sources or have significant DCD and DDJ. The method described for creating jitter suitable for Transmitter jitter-testing creates significant DCD which is ignored by the transmitter PLL, since it only responds to the rising edges of its reference input. Because the receiver responds to both edges of the pulse, this DCD affects the results in undesirable ways. The graph in *Figure 45* shows the relationship between input and output jitter at various input jitter-noise frequencies.

As expected, low frequency noise passes through the PLL filter unattenuated and higher frequencies are attenuated as theory would predict. Also as expected, the apparent bandwidth of the PLL filter varies as the transition density of the data stream varies. For the highest possible transition density (e.g., a 1010101... data stream) the natural frequency is highest, and for lower transition densities it is proportionally lower. The information shown here is characteristic of the HOTLink while receiving normal data. In this case the data was the BIST pattern.

Effective loop-bandwidth varies as a function of data rate, as shown in *Figure 45*. This variation is caused by various gain changes within and between

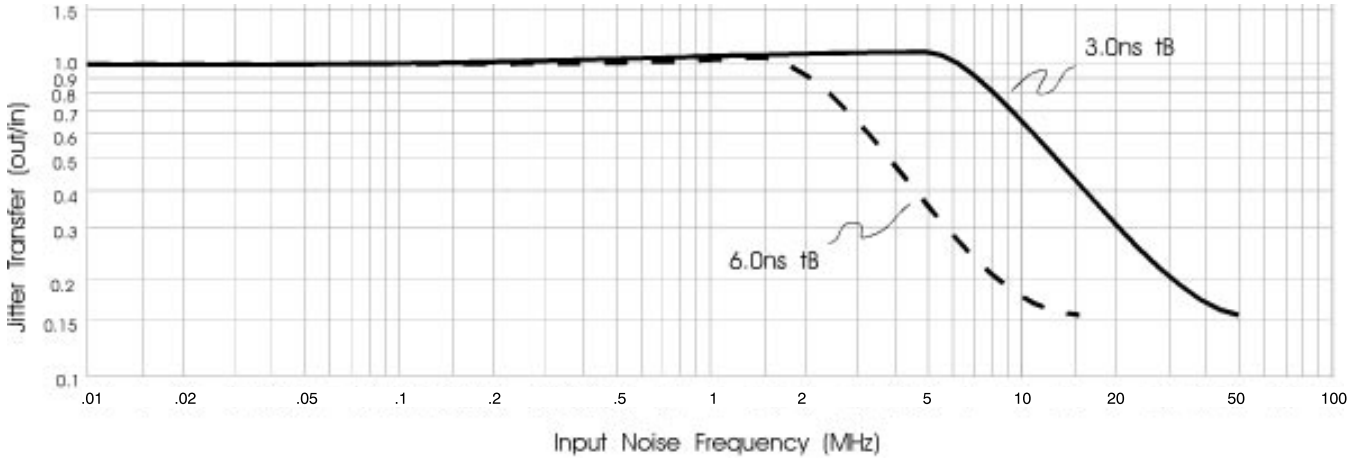
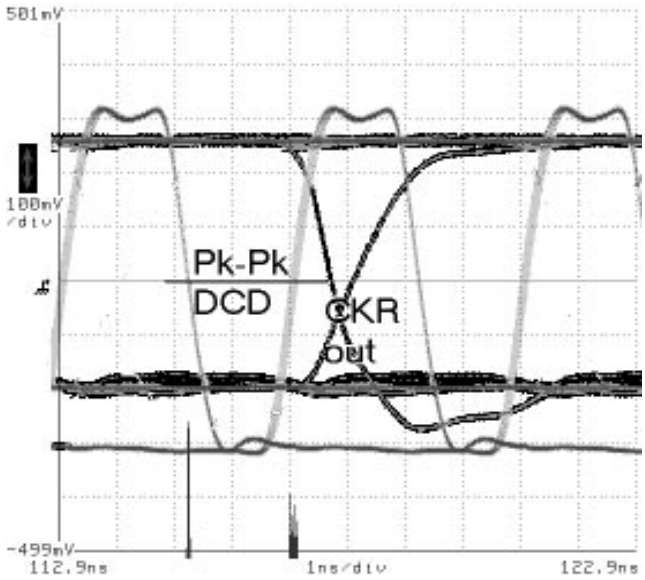


Figure 45. HOTLink Receiver Jitter Transfer Function (BIST Data)

the PLL component blocks. Some blocks have analog gain variations as a function of frequency, and others have a constant output response regardless of operating frequency. The behavior shown in *Figure 45* is unaffected by temperature, V_{CC} variation, and variations in manufacturing tolerance.

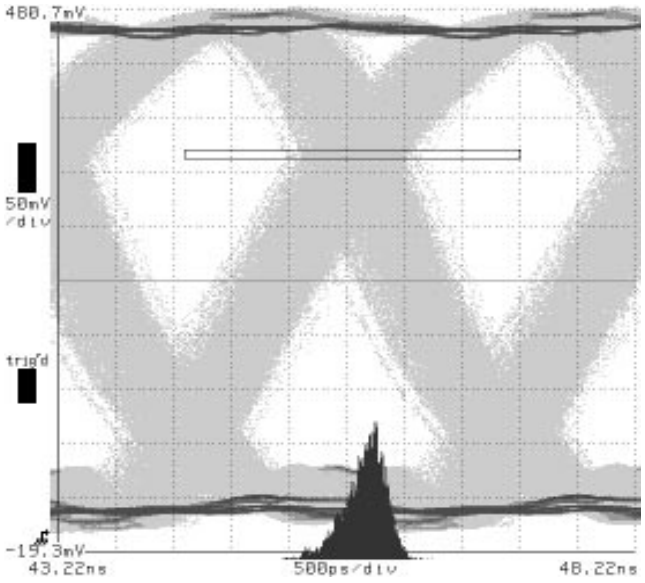
The Receive PLL transfer function is not sufficient to determine what the actual jitter out of the HOTLink Receiver might be. Different types of jitter have different transfer characteristics.

DCD-type jitter causes essentially no output jitter for input jitter magnitudes up to the point where the data is corrupted. The waveforms in *Figure 46* illustrate the jitter feed-through characteristics of the HOTLink Receiver. The input waveform is a continuous stream of 1-0-1-0-... bits that have been artificially distorted with the DCD Jitter generator described later (*Figure 49*). The 4.0 ns bits have been narrowed by about 1.96 ns (see the twin-peak histogram in *Figure 46*), and the CKR output shows less than 100 ps of jitter as illustrated by the darker trace superimposed on the input jitter waveform (note that the two traces have different vertical scales, but the same time scale).



250 Mbaud, Data = BIST,
DCD = 1.94 ns Pk−Pk

Figure 46. CKR Output Jitter as DCD Corrupted Data is Being Received



370 Mbaud DDJ in = 2.18 ns Pk−Pk,
CKR Jitter = 1.31 ns Pk−Pk

Figure 47. CKR Jitter Output as a Function of DDJ Input

When DDJ is applied to the data input, CKR jitter will increase. The illustration in *Figure 47* shows that when DDJ approaches maximum tolerable levels, the CKR output jitter increases appreciably. The test shown in *Figure 47* was performed using the same maximum tolerance jittered data shown in *Figure 35*. This 370-Mbaud signal (well beyond the datasheet limit) was generated using the BIST sequence transmitted through 250 feet of RG59 coaxial cable at 370 Mbaud, while operating with a received BER of $<4.5 \times 10^{-11}$. (The measurement in *Figure 47* is triggered by the pristine-bit-clock, which results in copies of the byte-rate TTL clock displayed at bit-clock intervals.)

This jitter feedthrough is partly caused by the low-frequency characteristic of the jitter, which is determined by its data content, and partly bacuase the actual PLL failure mode (as opposed to Data failure mode) is the same for DDJ as for DCD. In either case, when any data-pulse falls below the DCD pulsewidth limit, the PLL drops some of its tracking and locking information. In a normal data stream this loss is not regular, and causes minimal disturbance. The main effect is to increase jitter on the CKR output.