

ADN2817/ADN2818 BER Monitor and Sample Phase Adjust User Guide

by Eric Evans

INTRODUCTION

The ADN2817/ADN2818 provide a bit error rate (BER) measurement feature for estimating the actual BER of the IC. The feature also allows data eye jitter profiling and Q-factor estimation.

Other capabilities offered include:

- The ability to scan a region of the input data eye, which is offset from the actual sampling instant, to build up a pseudo bit error ratio profile.
- The ability to apply algorithms to process this data to obtain an accurate estimate of the BER at the actual sampling instant. User processing of the data results in greater accuracy and flexibility. A standby mode conserves power.
- Decomposition into random jitter (RJ) and deterministic jitter (DJ). The dual-Dirac model is used for DJ.
- Voltage output mode provides indication of BER and eye opening.
- Sample phase adjust ability. This mode is not concurrent with BER monitoring.

Understanding the BER of the circuit is useful for the following applications:

- BER monitoring indicates the onset of laser fading, and slow system degradation.
- Margin measurement, which is the difference between received SNR and the SNR required to guarantee a certain BER, such as $1e-10$.
- Driving adaptive equalizers.
- Sample phase adjust
- Optimum slice threshold adjust (ADN2817 only).
- Determination of dominant noise sources, that is, independent of power, proportional to $\sqrt{\text{power}}$, and proportional to power.

Circuitry within the ADN2817/ADN2818 allows measurement of the pseudo bit error ratio at phases that are offset from the actual sampling instant by more than approximately 0.05 UI. The implementation relies on the fact that by knowing the BER at sampling phases offset from the ideal sampling phase, it is possible to extrapolate to obtain an estimate of the BER at the actual sampling instant. This extrapolation relies on the assumption that the input jitter is composed of deterministic and random (Gaussian) components.

See the References section for resources that provide further information on BER estimation. The implementation requires off-chip control and data processing to estimate the actual BER. Additionally, there is a lower accuracy voltage output mode, which does not require user processing or I²C intervention.

TABLE OF CONTENTS

Introduction	1	Voltage Output BER Mode.....	14
BERMON and Sample Phase Adjust Specifications	3	Sample Phase Adjust Mode.....	16
BERMON Extrapolation Mode Specifications.....	3	Processing Algorithm for BER Extrapolation Mode	17
Available Output Modes	3	Outline of BER Extrapolation Algorithm	17
BERMON Analog Voltage Output Mode Specifications	4	Taking Samples for a Given PBER	18
Sample Phase Adjust Specifications	4	Parsing Based on Experimental Observations	21
A Review of BER Extrapolation Theory.....	5	Converting Parsed PBER Data to Q Factors.....	22
Characteristics of Bit Errors.....	5	Use Linear Regression to Find a Line Equation for Q vs. Phase	22
The Relationship Between Eye Diagram Statistics, BER, and Q Factor	6	Convert Extrapolated Q Factor to a BER Value at the Normal Decision Instant.....	25
Outline of BER Monitor Strategy	7	How to Detect When the BER > 1e-3.....	26
Characteristics of Pseudo-BER Data from Offset Decision Circuit	8	MATLAB Code.....	26
Modes of Operation	12	References.....	27
BER Extrapolation Mode Through I ² C Readback	12	Appendix A: Summary of Jitter Statistics Theory.....	28

BERMON AND SAMPLE PHASE ADJUST SPECIFICATIONS

BERMON EXTRAPOLATION MODE SPECIFICATIONS

The ADN2817/ADN2818 do not output the BER at the normal decision instant. It can output pseudo BER measurements to the left and right of the normal decision instant, from which the user must calculate the BER at the normal decision instant. A microprocessor is required to parse the data, detecting and removing non-Gaussian regions and using the remaining data for BER extrapolation.

AVAILABLE OUTPUT MODES

The following output modes are available:

- Normal decision circuit (NDC) retimed data and clock
- Sample phase adjusted retimed data and clock
- Normal decision circuit (NDC) retimed data and half frequency clock
- Sample phase adjusted retimed data and half frequency clock

Table 1. BERMON Extrapolation Mode Specifications

Parameter	Conditions	Min	Typ	Max	Unit
BERMON Extrapolation Mode	I ² C-controlled eye profiling				
Final Computed BER Accuracy	Input BER range 1×10^{-3} to 1×10^{-12} , input DJ < 0.4 UI, DJ ceiling > 1×10^{-2} ; asymmetry < 0.1 UI; requires external data processing algorithms to implement Q factor extrapolation		± 1		Decades
Number of Bits (NumBits)	Number of data bits to collect pseudo errors; user programmable in increment factors of 2^3 over the range 2^{18} to 2^{39}	2^{18}		2^{39}	UI
PBER Measurement Time			NumBits/ data rate		sec
BER Range				5×10^{-2}	BER
Sample Phase Adjust Resolution			6		Degrees
Sample Phase Adjust Accuracy			<6		Degrees
Sample Phase Adjust Range	With respect to normal sampling instant	-0.5		+0.5	UI
Minimum Input Signal Level	Differential peak-to-peak	4			mV
Power Increase	BER enabled		160		mW
	BER standby		77		mW

BERMON ANALOG VOLTAGE OUTPUT MODE SPECIFICATIONS

There is also an analog voltage output mode in the ADN2817 BER feature. The BER logic is set to operate at a single phase sampling point only. The reported pseudo BER (PBER) at this single sampling point is used to estimate the BER at the normal decision instant. This PBER value is decoded and then applied to a 6-bit DAC to provide an analog voltage output representative of the input BER.

This mode of operation is selected by bringing the BERMODE pin low. The DAC is current mode output. The user reads a voltage on the VBER pin that is a linear function of $\text{Log}(\text{BER})$, in the range of $1\text{e-}3$ to $1\text{e-}9$. This voltage is guaranteed to saturate when the input BER is above $\sim 1\text{e-}3$.

A 6-bit code word representative of the output BER can also be accessed over the I²C.

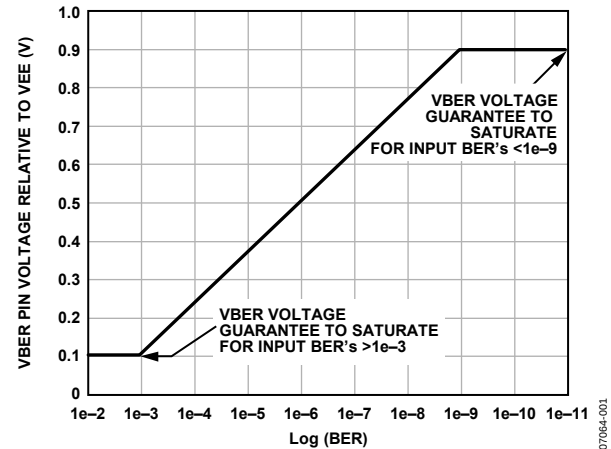


Figure 1. Analog BER Output Voltage Characteristics

Table 2. Analog Voltage Output Mode Specifications

Parameter	Conditions	Min	Typ	Max	Unit
BERMON Voltage Output Mode	Analog voltage output				
BER Accuracy	Input BER range 1×10^{-3} to 1×10^{-9} , input DJ = 0 UI, DJ ceiling $> 1 \times 10^{-2}$; asymmetry = 0 UI; BER is read as a voltage on the VBER pin, when the BER mode pin = VEE		± 1		Decades
	Input BER range 1×10^{-3} to 1×10^{-9} , input DJ = 0.2 UI, DJ ceiling $> 1 \times 10^{-2}$; asymmetry = 0 UI; BER is read as a voltage on the VBER pin, when the BER mode pin = VEE		+1/-2		Decades
NumBits	Number of data bits to collect pseudo errors		2^{27}		UI
Measurement Time	2.5 Gbps		0.054		sec
	1 Gbps		0.134		sec
	155 Mbps		0.865		sec
	10 Mbps		1.34		sec
VBER Voltage Range	Via 3 k Ω resistor to VEE	0.1		0.9	V
Minimum Input Signal Level	Differential peak-to-peak	4			mV
Power Increase	BER voltage mode		160		mW

SAMPLE PHASE ADJUST SPECIFICATIONS

Table 3. Sample Phase Adjust Specifications

Parameter	Conditions	Min	Typ	Max	Unit
Sample Phase Adjust Mode					
Sample Phase Adjust Step Size	Monotonic		6		Degrees
Sample Phase Adjust Accuracy			<6		Degrees
Sample Phase Adjust Range	With respect to normal sampling instant	-0.5		+0.5	UI
Power Increase			160		mW

A REVIEW OF BER EXTRAPOLATION THEORY

CHARACTERISTICS OF BIT ERRORS

Bit errors occur when a data edge and clock edge are sufficiently displaced from their ideal locations. This causes the retiming flip-flop in the ADN2817/ADN2818 CDR to sample the data signal at a location in time where the wrong polarity of bits is sampled. Displacement of the clock and data edges has many underlying mechanisms, including VCO phase noise, electrical receiver noise, pattern dependent causes (such as ISI and capacitor droop), and crosstalk. These jitter mechanisms are conveniently described with statistical distributions, which describe the statistical behavior of the displacement of the data edges. Knowledge of statistical characteristics of the data edges, combined with those of the retiming clock edges can be used to determine the BER of the receiver.

A common model for the received data edge statistics consists of a region of bounded deterministic jitter (DJ) existing around the mean data transition locations (known as the UI boundaries), plus a region of unbounded random jitter (RJ), extending from the edge of the DJ region, towards the center of the data eye. The RJ of the retiming clock is usually considered to be summed in with the random jitter of the data eye. The DJ can be modeled as a dual-Dirac delta function, and the RJ is modeled by a normal/Gaussian distribution. For further information, refer to the Agilent Technologies white paper and the Fibre Channel publication listed in the References section. Figure 2 shows a jittered data eye, along with probability density functions for the left and right data edge timing. A common method for representing the distribution is with a bathtub curve.

Figure 3 gives an example of a bathtub curve. This curve shows the cumulative probability of data edge occurrence as a function of time from the UI boundaries. One way to interpret this curve is to ideally sample the data eye at any given location in time. You could then determine the probability of sampling the wrong data bit by reading the corresponding probability from the y-axis, as this represents the probability that the data edge will occur either too early or too late for the sampling clock.

Additionally, if you were to assume a data transition density of 0.5, the bit error rate would be half this probability value.

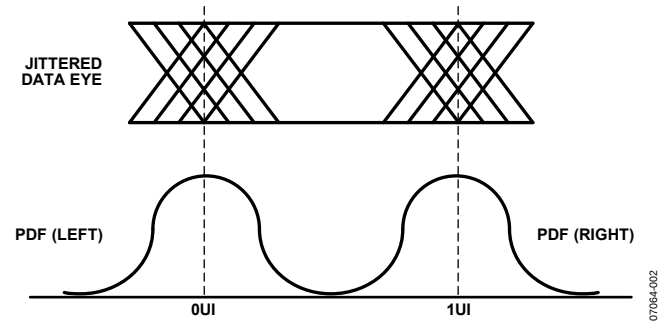


Figure 2. Probability Density Functions of a Jittered Data Eye

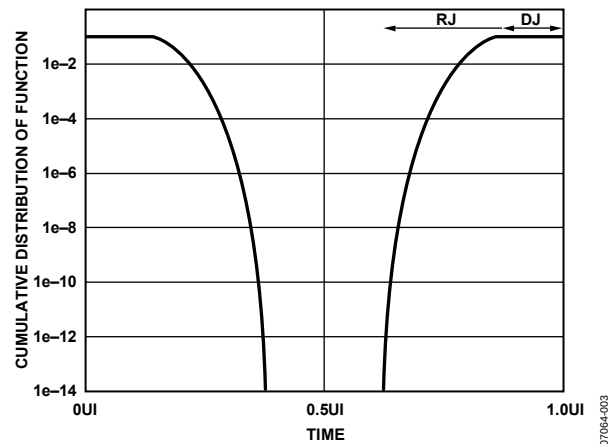


Figure 3. Example Input Jitter Statistics/Bathtub Curve for a Data Eye

Depending on the system application, the bathtub curve as shown in Figure 3 may be too simplistic. Due to pattern run length effects, the region of deterministic jitter may be inadequately described by a double delta function, with a probability of error of 0.25, as illustrated in Figure 3. Figure 4 illustrates a data characteristic where the bounded DJ region has a BER that reduces from 0.25 to $\sim 1e-3$. This lower limit is known as the DJ ceiling.

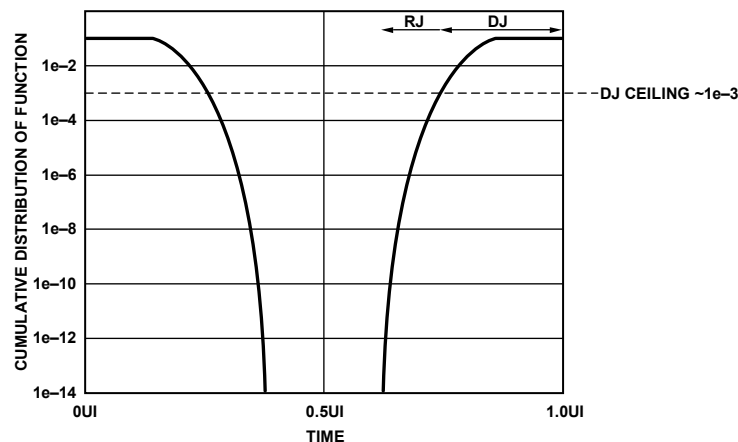


Figure 4. Example Input Jitter Statistics/Bathtub Curve Showing a Low DJ Ceiling

In certain data communication specifications, a jitter type known as bounded, uncorrelated, random jitter (or bounded Gaussian jitter) is defined. Although this jitter displays a Gaussian distribution, it is bounded in the maximum phase over which it exists. The problem with these jitter distributions is in distinguishing the non-Gaussian regions from the Gaussian regions. BER extrapolation techniques rely on extrapolating the BER characteristic for a known Gaussian region, out to the normal decision instant.

The goal is to be able to determine the eye statistics (bathtub curve) of the received data on the ADN2817. This information allows the user to determine:

- The BER of the CDR
- If the CDR is sampling at the optimum location
- How to adjust the sampling point of the CDR to achieve the lowest BER

THE RELATIONSHIP BETWEEN EYE DIAGRAM STATISTICS, BER, AND Q FACTOR

A sample input data eye diagram and its associated jitter cumulative distribution function (CDF) are illustrated in Figure 5. Bit errors occur when a data edge and the clock edge are displaced by jitter, which causes the retiming flip-flop in the phase detector to sample the data signal at a location in time where the wrong polarity of bit is sampled.

The data transition probability density functions (PDFs) for the left and right data edges are composed of a bounded deterministic jitter region plus an unbounded random jitter region that has a Gaussian (normal) distribution. Referring to Figure 5, the probability of a bit error is determined by the probability of the left-most data edge occurring to the right of the sampling instant, plus the probability of the right-most data edge occurring to the left of the sampling instant. This can be found by integrating the Gaussian tails of the left and right PDFs from the sampling instant to infinity, as

$$BER = \frac{1}{2} \int_{t_s}^{\infty} PDF_l + \frac{1}{2} \int_{-\infty}^{t_s} PDF_r \quad (1)$$

where t_s is the sampling instant and the factor of 1/2 accounts for a transition density of 0.5.

The integral terms are the cumulative distribution functions and for Gaussian PDFs these can be evaluated using the complimentary error function

$$\left(\text{erfc} = 2 \int_x^{\infty} PDF \right)$$

For further details, see Appendix A: Summary of Jitter Statistics Theory, or see Montgomery and Runger in the References section. For any sampling instant in the Gaussian region of the distribution function the BER can be predicted using

$$BER = \frac{1}{4} \text{erfc} \left(\frac{|DecisionThreshold_l|}{\sqrt{2} \times \sigma_l} \right) + \frac{1}{4} \text{erfc} \left(\frac{|DecisionThreshold_r|}{\sqrt{2} \times \sigma_r} \right) \quad (2)$$

Equation 2 assumes equiprobable bits and a transition density of 0.5, where the decision threshold is the time between the beginning of the Gaussian region and the sampling instant, and σ represents the standard deviation of the Gaussian region. For example, if the sampling instant is at $T/2$ then the BER can be predicted using

$$Pe = BER = \frac{1}{4} \text{erfc} \left(\frac{|T/2 - DJ_l|}{\sqrt{2} \times \sigma_l} \right) + \frac{1}{4} \text{erfc} \left(\frac{|T/2 - DJ_r|}{\sqrt{2} \times \sigma_r} \right) \quad (3)$$

which becomes

$$BER = \frac{1}{4} \text{erfc} \left(\frac{Q_l}{\sqrt{2}} \right) + \frac{1}{4} \text{erfc} \left(\frac{Q_r}{\sqrt{2}} \right) \quad (4)$$

where:

$$Q_x = \frac{|T/2 - DJ_x|}{\sigma_x}$$

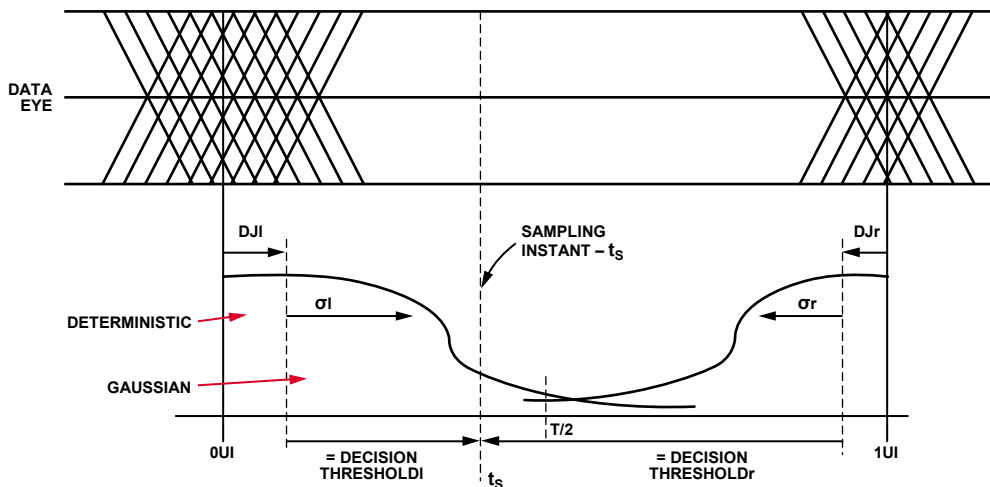


Figure 5. Asymmetric CDFs with DJ, Nonoptimum Threshold Setting

For example, if the BER is measured to the left of the minimum, the BER is dominated by the left-most distribution and is thus given by

$$BER \approx \frac{1}{4} \operatorname{erfc}\left(\frac{QI}{\sqrt{2}}\right)$$

With this definition, it is evident that the Q factor at any sampling instant in the RJ region can be determined from a measurement of the BER, or, alternatively, that the BER at any sampling instant in the RJ region can be estimated if the Q factor is known. The Q factor is representative of the signal (data period) to noise (jitter) ratio at that sampling instant, and the concept of Q factor only applies to the region of Gaussian jitter. The relationship between Q factor and BER is defined by the complimentary error function (erfc), and, in practice, can be obtained from lookup tables or a polynomial.

Therefore, if the BER is known at any sampling instant away from the minimum BER, the Q factor can be determined at that sampling instant. Additionally, if the sampling instant is known then, from Equation 3, the σ of the Gaussian distribution can be calculated.

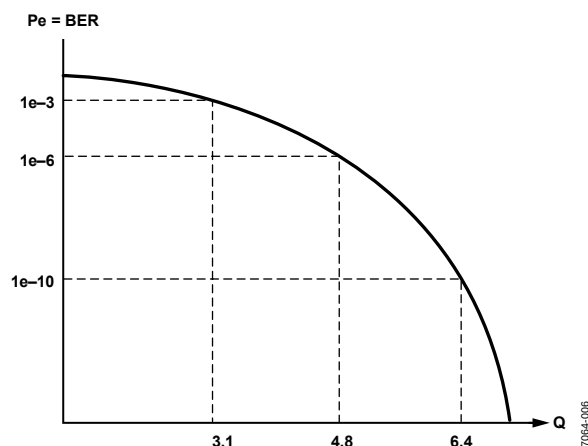


Figure 6. Relationship Between the BER and the Q Factor

Table 4. BER vs. Q Factor

BER	Q	ΔQ	$\Delta Q\%$
1e-1	1.28		
1e-2	2.32	1.04	81
1e-3	3.09	0.77	33
1e-4	3.72	0.63	20
1e-5	4.26	0.54	15
1e-6	4.75	0.49	11
1e-7	5.2	0.45	9.4
1e-8	5.61	0.41	7.9
1e-9	6	0.39	6.9
1e-10	6.36	0.36	6
1e-11	6.71	0.35	5.5
1e-12	7.03	0.32	4.8

OUTLINE OF BER MONITOR STRATEGY

The BER monitor strategy is a four-step process.

1. Degrade the BER in an offset decision circuit (ODC) by sampling away from the normal decision instant (see Figure 7).
2. By ensuring that the sampling clock in the offset decision circuit is sufficiently far from the normal sampling clock, the BER in the offset decision circuit is much higher than that in the normal decision circuit (NDC). The data pattern in the offset decision circuit is compared with the data in the normal decision circuit. The number of bits received differently is a very good measure of the error rate of the offset decision circuit at the programmed phase offset. This error rate is referred to as the pseudo BER (PBER). This gives pairs of PBER and phase data.
3. This data is parsed to extract only the data that is in the Gaussian region.
4. By knowing the PBER at some offset decision instants, Q factors can be calculated at those instants. Some checks for non-Gaussian data can also be performed. The Q factor is then linearly extrapolated to determine the Q factor at the normal decision instant. From these extrapolated Q factor values, the BER at the normal decision instant is calculated. Similar methods are being incorporated into certain test standards (TIA/EIA OF-STP8 and FC MJSQ) and are used in BER test equipment.

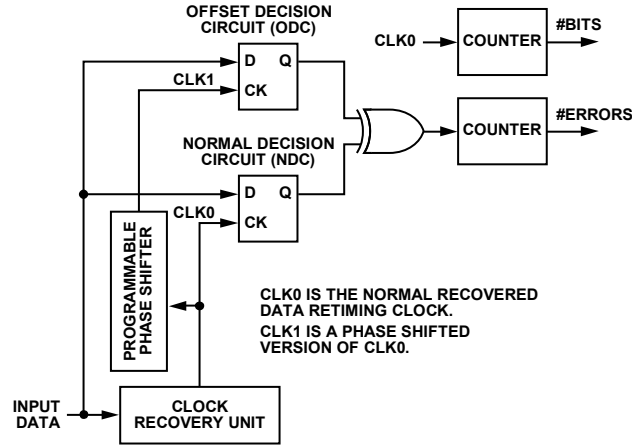
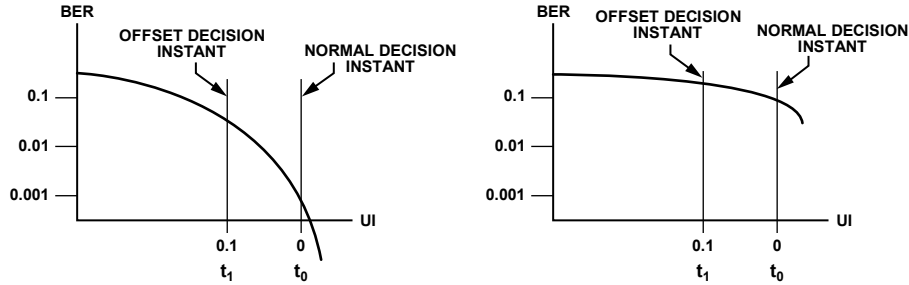


Figure 7. BER Monitor Architecture Showing Two Decision Circuits

Figure 8. Cumulative Distribution Functions (CDFs) of Input Jitter for Input BERs of $\sim 1e-3$, and $\sim 1e-1$

CHARACTERISTICS OF PSEUDO-BER DATA FROM OFFSET DECISION CIRCUIT

Definition of Pseudo BER

The pseudo BER data that is reported from the chip is actually the difference between the BER at the normal decision instant and the BER at the offset decision instant. This is illustrated in Figure 8. This shows the CDFs of the input jitter for typical input BERs of $\sim 1e-3$ and $\sim 1e-1$. The normal sampling instant t_0 (at 0 UI) and the offset sampling instant t_1 (at -0.1 UI, for example) are also shown. Data edges that cross to the right of t_1 are counted as pseudo errors, unless they also cross to the right of t_0 .

The BER monitor circuitry measures the PBER at the offset decision instant, t_1 . Mathematically, the PBER measurement is given as

$$PBER = \left| 0.5 \times \left(\int_{t_1}^{\infty} PDF - \int_{t_0}^{\infty} PDF \right) \right| \quad (5)$$

Provided that the actual bit error rate at t_1 is at least, say, ten times greater than that at t_0 , then the first term is much larger than the second term and thus the PBER at sample point t_1 is approximately given by

$$PBER \approx \left| 0.5 \times \left(\int_{t_1}^{\infty} PDF \right) \right| \quad (6)$$

This is the equation for the BER at t_1 from Equation 1, assuming that the BER at t_1 is dominated by one side of the distribution only. The integral in Equation 6 is exactly the complementary error function (erfc) used in Equation 4. Thus, using a duplicate decision channel allows one to accurately estimate the BER CDF at sample points offset from the normal decision instant. Knowing that the distribution is Gaussian and has data points on that distribution is what allows one to extrapolate to estimate the BER at the normal decision instant. The approximation used in Equation 6 gives rise to a few issues that need to be accounted for during the parsing of the data.

Pseudo BER Implementation Null

When the two sampling instants are brought close together, then the first and second terms in Equation 5 become closer in value to each other. The resulting PBER value then becomes a small value, eventually becoming zero as the two sample points are placed at the same instant. In this situation, the error between the right-hand side of Equation 5 (pseudo BER) and the right-hand side of Equation 6 (ideal BER) becomes large. From theory and measurements, it is found that if the sample phase of the ODC is kept greater than 0.05 UI from the sample phase of the NDC, then this error is kept acceptably small. This is for actual BERs of less than $1e-2$, and any sensible combinations of DJ and RJ. An example of this implementation null is shown in Figure 9. Do not use data within 0.05 UI of the NDC sample phase for the Q factor extrapolation algorithm.

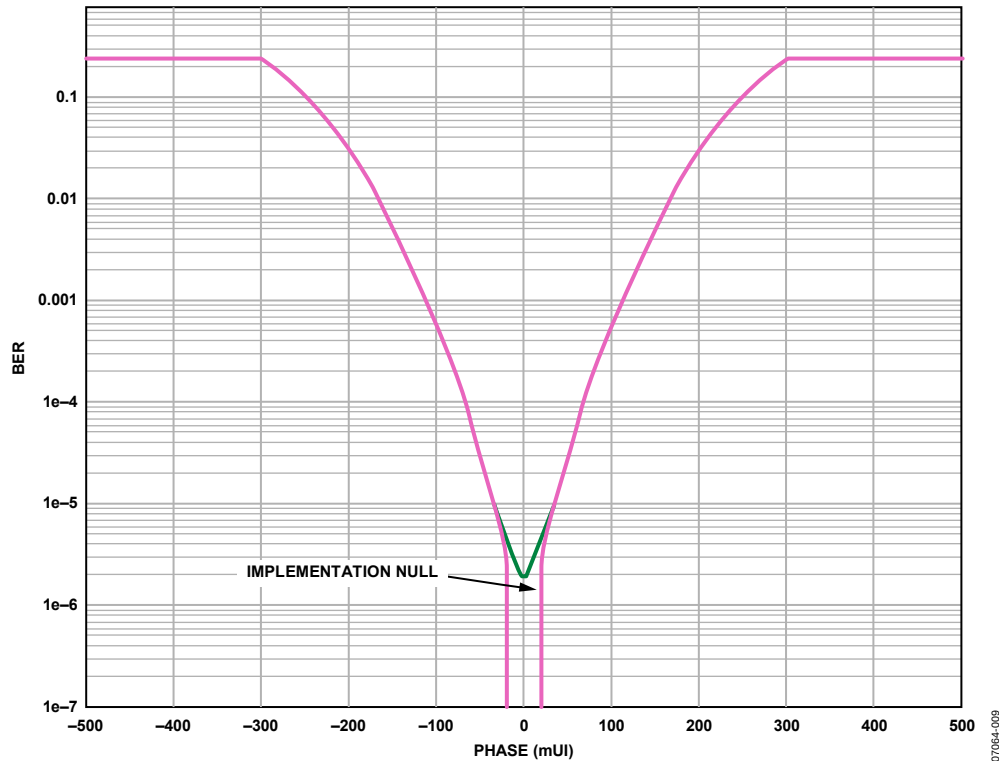


Figure 9. Implementation Null Around 0 UI Offset to the NDC Sampling Instant. $RJ = 0.065$ UI, $DJ = 0.2$ UI.
The green trace represents the ideal BER characteristic while the pink trace shows a typical reported pseudo BER characteristic.

Pseudo BER Nonmonotonic at Very High Bit Error Rates

The reported pseudo BER for small sample phase offsets is nonmonotonic at very high input BERs. A typical characteristic is shown in Figure 10. The blue trace shows the input BER as the input random jitter in UIs is increased (x-axis). The green trace is the actual BER when sampling at 0.1 UIs offset to the normal decision instant. The pink trace shows the reported pseudo BER at 0.1 UI. At low BERs ($<1e-2$) the measured PBER is a fairly good estimate of the actual input BER at 0.1 UI offset (green vs. pink). At higher input BERs (above $\sim 2e-2$) the reported PBER actually starts decreasing as the input BER continues to increase.

Figure 8 shows the CDFs of the input jitter for example input BERs of $\sim 1e-3$ and $\sim 1e-1$. For very high input BERs, however, the second term in Equation 5 is no longer negligible. In fact, as the input BER increases above, for example, $1e-2$ the second

term becomes very close in magnitude to the first term, with the result that the reported PBER computed in Equation 5 starts to get smaller as the input BER increases. This is why the PBER value is nonmonotonic with increasing input BER at very high BERs, as shown with the pink trace in Figure 10.

In practice, this is not a problem because the ADN2817 loss of lock detector is guaranteed to declare loss of lock when the input BER is in the range of $1e-2$ to $1e-1$. Consequently, it is not possible for the input BER to be $>1e-2$, and to have the output indicate that it is $<1e-3$. In addition, PBER data greater than $1e-2$ should not be used for Q factor extrapolation because it is likely to be contaminated by DJ.

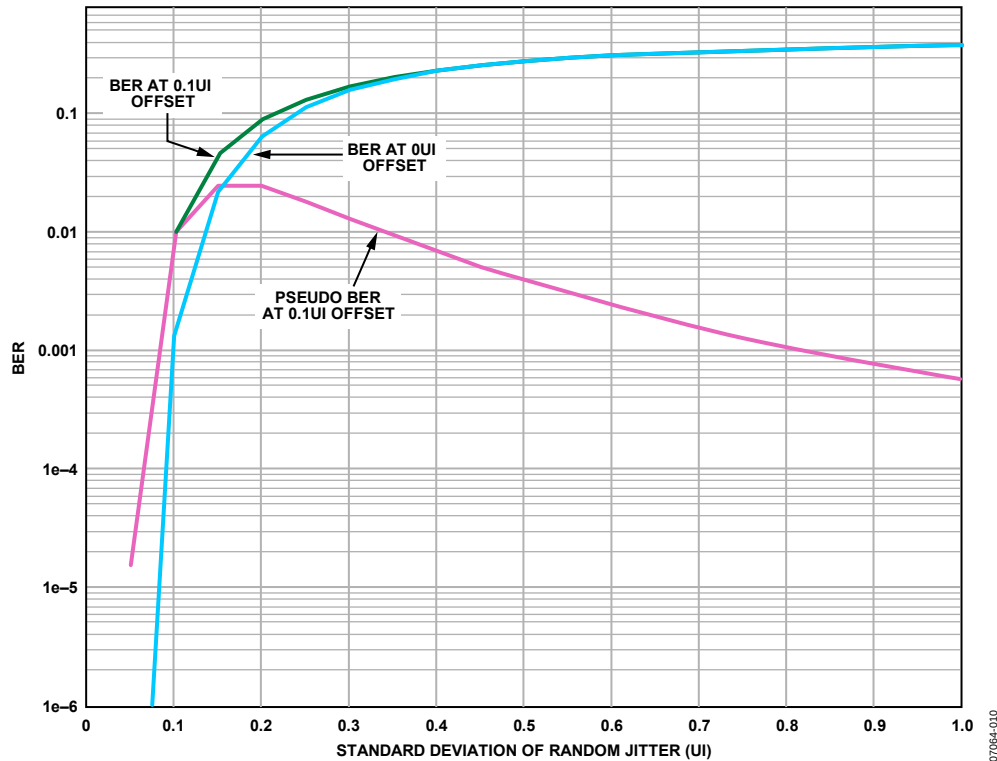


Figure 10. Nonmonotonic Report RBER at Very High Input BERs. The blue trace shows the input BER as the input random jitter in UIs is increased (x-axis). The green trace is the actual BER when sampling at 0.1 UIs offset to the normal decision instant. The pink trace shows the reported pseudo BER at 0.1 UI.

Pseudo BER Plateau with Asymmetric Jitter Distributions

The left and right BER profiles around the UI boundaries are rarely symmetric in optical communication receivers. Another interpretation of this is that the normal decision circuit is not sampling at the optimum BER location. When the NDC is sampling away from the optimum (lowest input BER) and the ODC retiming phase is moved by small amounts left and right of the normal decision phase, the reported PBER is always increased due to the implementation null. In addition, the asymmetry results in a plateau on the pseudo BER curve to one side of the NDC decision instant, beyond the implementation null. An example is shown in Figure 11.

Each side of the BER characteristic is dominated by jittered data edges from that side, where the input BER minimum is the boundary between the left and right-hand side. For example, if the ODC sampling instant is set to -0.2 UI in Figure 11, pseudo errors are counted every time a left-hand side data edge occurs to the right of the ODC sampling instant, but not to the right of the NDC sampling instant. Thus, when the ODC sampling instant is to the left of the NDC sampling instant, recorded pseudo errors are dominated by left-hand side data edges, and there is almost no bit error due to right-hand side data edges.

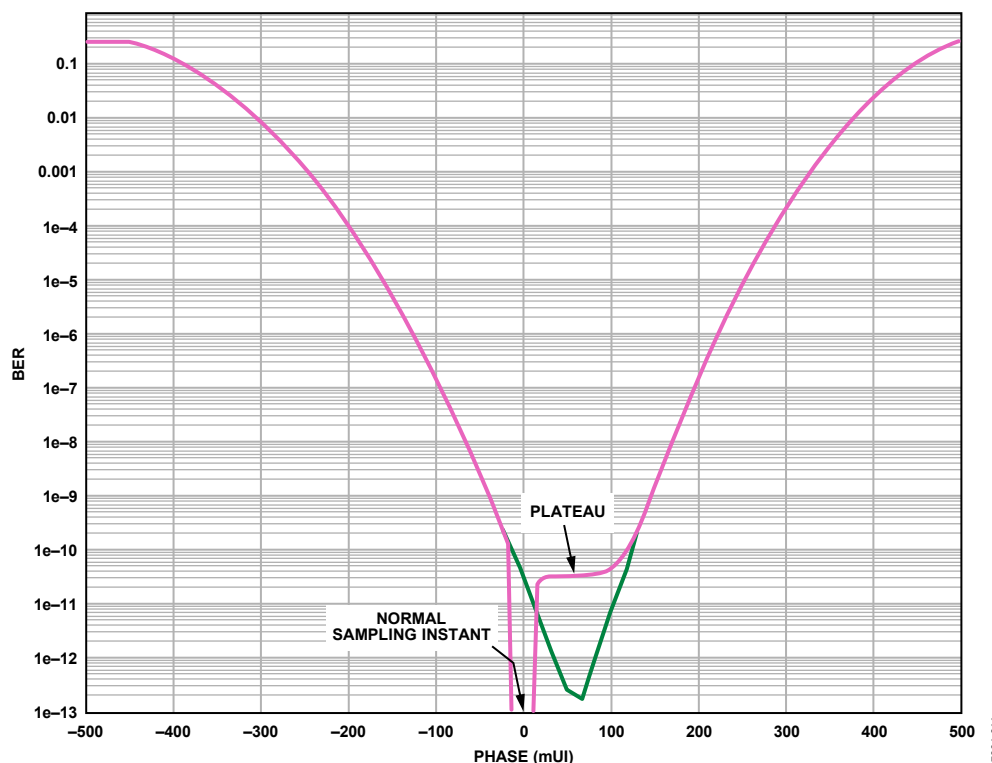


Figure 11. Plateau Due to the Asymmetric Input Distribution. Implementation Null Around 0 UI Offset to the NDC Sampling Instant. The green trace represents the input BER characteristic while the pink trace shows a typical reported pseudo BER characteristic.

When the ODC sampling instant is set to +0.05 UI in Figure 11, then pseudo errors are counted every time a left-hand side data edge occurs to the right of the NDC sampling instant (0 UI), but not to the right of the ODC sampling instant (now +0.05 UI). (From the right-hand BER characteristic in Figure 11, it is evident that there are much fewer occurrences of right-hand sided data edges occurring to the left of either +0.05 UI or 0 UI, so these can be ignored.) This means that the reported PBER value is just equal to the BER of the NDC, sampling at 0 UI.

This is the case for all ODC sampling instants where the ODC sampling instant BER is less than the NDC sampling instant BER, and beyond the implementation null. This is what provides the plateau.

The data used for Q factor extrapolation can be checked to ensure that it is not affected by this plateau. Fortunately, this plateau effect is benign as far as BER extrapolation is concerned. It is clear from Figure 11 that when there is an asymmetry-induced plateau that the BER at the normal decision instant (phase = 0) is dominated by the statistics from the side where the plateau is not present.

In realistic situations (as in Figure 11), a poorly estimated Q factor on the right-hand side is still going to predict a lower BER contribution than the accurate BER contribution/estimate from the left-hand side, and so this error is less important. It is more important to ignore data affected by the plateau when trying to estimate the optimum sampling instant, although even here the BER extrapolation algorithm is good with asymmetries of less than 0.2 UI.

Assumptions Regarding Deterministic Jitter

The BER algorithm can only work if there is a region of valid, normally distributed (Gaussian) jitter. If there is no significant region of Gaussian jitter, then the algorithm still provides some indication of BER, but accuracy is degraded. There are two requirements.

1. The DJ ceiling must be greater than some minimum value. A minimum BER of $1e-2$ is used here. For example, at phase offsets where the probability of error is $<1e-2$, the jitter characteristics are ideally purely Gaussian. Data dependent jitter with long pattern lengths may violate this assumption leading to a degradation in the BER extrapolation accuracy. This is to ensure that there is at least a small region of Gaussian jitter that can be used for extrapolation when the BER is only $1e-3$, for example. For further information, refer to the Fibre Channel publication in the References section.
2. The models for DJ assume that it can be represented as a dual-delta function, as shown in the Fibre Channel publication listed in the References section. The implication is that the BER within the DJ region is assumed to be fixed at 0.25 UI; otherwise, the jitter has a Gaussian characteristic. This is the dual-delta function definition of the deterministic jitter as is also described in the Agilent Technologies white paper listed in the References section. It is always less than the actual deterministic jitter. This does not affect the ability to determine the BER of the ADN2817.

MODES OF OPERATION

Two modes of operation are available for the BER monitor feature. There is also a sample phase adjust mode. Only one mode can be operational at a time. The primary mode is the BER extrapolation mode as discussed in the A Review of BER Extrapolation Theory section and the Processing Algorithm for BER Extrapolation Mode section. This is where the user scans the input eye in the range of ± 0.5 UI of the data center, reads back the measured pseudo BER over the I²C, and then processes the data to determine the BER. Using the BER feature in this way provides for the greatest accuracy in BER estimation. This is because the magnitude of both random (Gaussian) jitter and deterministic jitter can be estimated and used to predict the actual BER.

The voltage output mode is a secondary BER monitor mode where the ADN2817 autonomously samples the PBER at 0.1 UI offset and decodes this value to provide an estimate of the input BER. This estimate is output via a DAC as an analog voltage to the BERMON pin. This mode is characterized by lower accuracy because it cannot account for large amounts of deterministic jitter on the input data eye. This mode is described further in the Voltage Output BER Mode section.

With the sample phase adjust mode, the user can set the sampling instant of the input data relative to the sample instant acquired by the CDR to optimize the received BER.

BER EXTRAPOLATION MODE THROUGH I²C READBACK

In the BER extrapolation mode, the BER is measured as outlined above.

Power Saving

Three power settings can be applied.

- In the BER enabled setting, all of the BER functionality is powered up. The user can perform a PBER measurements through I²C control.
- In the BER standby setting, the BER is placed into a low power mode, saving approximately 80 mW. Note that this mode can only be invoked if the BER Enabled mode has been previously set.
- In the BER off setting, all the BER circuitry is powered down, and operation of the CDR is unaffected, saving approximately 160 mW of power.

These modes are defined to allow optimal power saving opportunities. It is not possible to switch between BER off mode and BER enabled mode without losing lock. Switching between BER standby

mode and BER enabled mode is achieved without interrupting the data recovery. Setting BER standby mode without having set the BER enable mode is not supported.

BER On Setting

The BER on setting allows the user to scan the incoming data eye in the time dimension and to build up a profile of the pseudo BER statistics.

User Protocol

The user protocol consists of seven steps.

1. Power up the BER circuitry through the I²C, (BERCTLB[5] = 1). The BERMODE pin is set high. The part deasserts the Loss of Lock (LOL) pin.
2. Program the sample phase offset, (PHASE[5:0]), and the number of data bits to be counted (BERCTLA[7:5], NumBits is the choice between 2^{18} , 2^{21} , 2^{24} , 2^{27} , 2^{30} , 2^{33} , 2^{36} or 2^{39}) through the I²C.
3. Apply a reset pulse to the BER circuit through the I²C by writing a 1 followed by a 0 to BERCTLA[3].
4. Note that BER logic indicates End Of BER Measurement with EOBM I²C bit (BERSTS[0] = 1).
5. Poll the I²C to determine if EOBM has been asserted.
6. Read back NumErrors [39:0] through the I²C by setting BERCTLA[2:0] and reading BER_RES[7:0]. PBER for programmed sample phase is calculated as NumErrors/NumBits.
7. Power down the BER, or request another BER measurement (see Step 2).

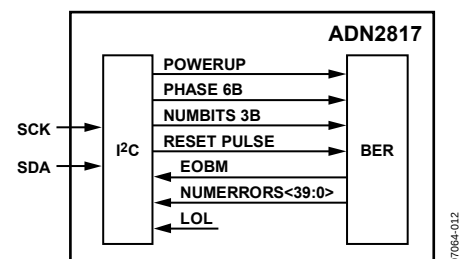


Figure 12. I²C Interface to BER Monitor Circuitry

The user sweeps the phase over 0 to 1 UI to get the BER profile required. The sample phase code word can be incremented in integer steps over the range of -30 to $+30$, which corresponds to a sample phase range of -0.5 UI to $+0.5$ UI. The resulting resolution is 1/60th of a UI, or 6° . The phase code (I²C: PHASE[5:0]) is in twos complement format.

An example of I²C code for a complete scan of the eye follows:

1. WR BERCTLB = 0x20 *** POWER UP BER (after acquiring)
1. *** DO A PBER MEASUREMENT
2. WR PHASE = 0x22 *** Phase = -0.5UI = -30 code.
3. WR BERCTLA 0x48 *** Apply BER Startpulse. Numbits initially set to 2²⁴ bits.
4. WR BERCTLA 0x40
5. RD BERSTS *** Monitor EOBM until set.
6. *** Readback NumErrors
7. WR BERCTLA 0x00 *** NumErrors Byte 0 (Least Significant Byte)
8. RD BER_RES
9. WR BERCTLA 0x01 *** NumErrors Byte 1
10. RD BER_RES
11. WR BERCTLA 0x02 *** NumErrors Byte 2
12. RD BER_RES
13. WR BERCTLA 0x03 *** NumErrors Byte 3
14. RD BER_RES
15. WR BERCTLA 0x04 *** NumErrors Byte 4
16. RD BER_RES
17. Repeat Step 2 to Step 16, however, in Step 2 increment the phase code step by 1 to cover the range of -30 to +30. Depending on the previous measurement, determine if the NumBits value should be increased/decreased. NumBits should always be set to give greater than 1000 pseudo errors, if possible. Within code values of approximately -6 to +6, it may not be possible to select a large enough value for NumBits. The user should reduce the number of phase locations used when applying the algorithms described in the A Review of BER Extrapolation Theory section and the Processing Algorithm for BER Extrapolation Mode section.

Note that WR indicates an I²C write operation, and RD indicates an I²C read operation.

VOLTAGE OUTPUT BER MODE

A secondary BER mode of operation is implemented. This mode gives easy access to an estimate of the BER. The measurement is performed autonomously by the ADN2817, and the result is output as a voltage on a pin, from which the actual BER can be inferred.

The user merely has to keep the BERMODE pin low and read the voltage on the VBER pin. A 3.0 kΩ resistor is placed between the VBER pin and the VEE pin. Alternatively, a 6-bit value can be read over the I²C (I²C: BER_DAC[5:0]). In this mode, the dynamic range is limited. Figure 14 indicates the relationship between the input BER and the voltage on the VBER pin.

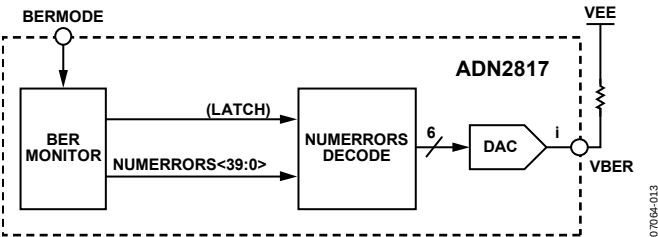


Figure 13. Voltage Output Mode. I²C control not needed.

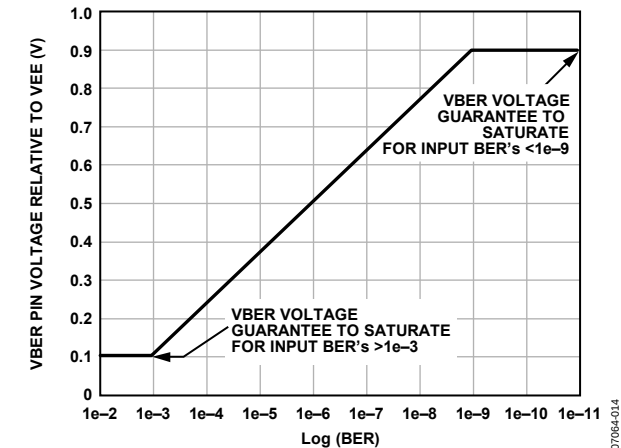


Figure 14. Output Voltage vs. BER

Because there is no method of differentiating between deterministic and random jitter, this mode of operation is inherently less accurate than the BER extrapolation mode. The typical accuracy limits for this mode of operation are shown in Figure 15 for the case of 0 UIp-p, and 0.2 UIp-p of applied DJ. This mode cannot run concurrently with BER extrapolation mode.

The update rate on the VBER pin is a function of data rate, and is determined as

$$VoltageUpdatedTime = 2^{27}/(DataRate).$$

Table 5. Update Rates for Some Key Data Rates

Data Rate	Measurement Time = No. of Bits/DataRate
2.5 G	54 ms
1 Gbps	134 ms
155 Mbps	865 ms
10 Mbps	1.34 sec

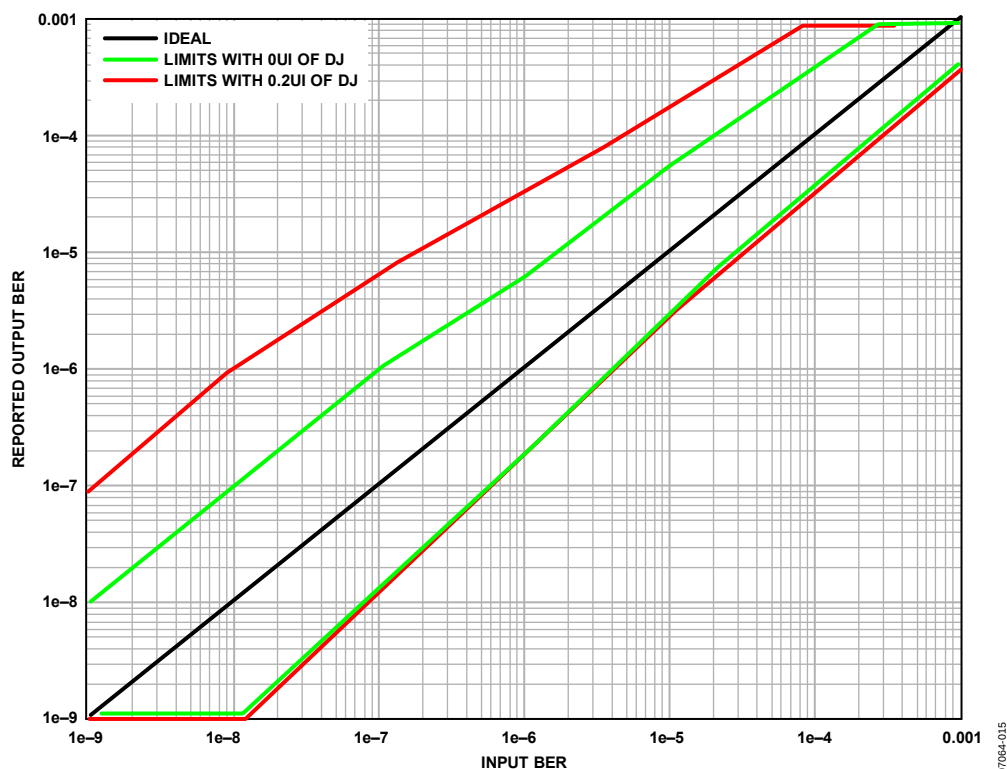


Figure 15. Accuracy of BER Voltage Output Mode

07964-015

SAMPLE PHASE ADJUST MODE

When BER monitoring is not required, the sample phase of the offset decision circuit can be adjusted as in BER extrapolation mode, and the corresponding clock and sampled data can be taken off-chip at the normal outputs. This allows implementation of the sample phase adjust mode. This may be required if the user wishes to independently choose the data sampling instant, relative to the normal CDR sampling instant. Switching between this mode, and the BER monitoring mode causes a temporary loss of output data.

The sample phase code word can be incremented in integer steps over the range of -30 to $+30$, which corresponds to a sample phase range of -0.5 UI to $+0.5$ UI. The resulting

resolution is $1/60$ th of a UI or 6° . The phase code (I²C: PHASE[5:0]) is in twos complement format. Example I²C code showing how to set the part into this mode is given as follows:

- The part is allowed to acquire normally. The BERMODE pin is high.
- WR BERCTLB=0x21*** POWER UP BER. Route ODC clock & Data to Output (implements Sample Phase Adjust).
- WR PHASE = 0x06*** Phase = $+0.1$ UI = code +6.

PROCESSING ALGORITHM FOR BER EXTRAPOLATION MODE

This section provides a detailed example of a processing algorithm applied to obtain a BER estimate from the pseudo BER data obtained from the ADN2817. The proposed algorithm can be used to parse the pseudo BER data from the ADN2817, estimate the Q factor, and consequently estimate the BER at the normal decision instant.

OUTLINE OF BER EXTRAPOLATION ALGORITHM

Figure 16 shows an eye diagram statistic and corresponding Q factor characteristic. For every BER value in the Gaussian region, its corresponding Q factor can be obtained using either a lookup table or a known polynomial. The left and right data are processed independently before combining the left and right estimate Q factors at the normal decision instant to obtain the BER at that sampling instant, according to Equation 4, and repeated in Equation 7. Note that for the Gaussian region the Q factor is, by definition, a straight line.

$$BER = \frac{1}{4} \operatorname{erfc}\left(\frac{Ql}{\sqrt{2}}\right) + \frac{1}{4} \operatorname{erfc}\left(\frac{Qr}{\sqrt{2}}\right) \quad (7)$$

For the nonhorizontal region of the Q factor graph

$$Q = mt + C$$

where:

m is the slope.

t is the sampling instant with respect to 0 UI.

C is the Y intercept for $t = 0$.

For a pair of data points on the Q factor graph, the straight line equations become

$$\frac{t - t_1}{t_2 - t_1} = \frac{Q - Q_1}{Q_2 - Q_1}$$

which implies

$$Q = \left(\frac{Q_2 - Q_1}{t_2 - t_1} \right) \times t - \left(\frac{Q_2 - Q_1}{t_2 - t_1} \right) \times t_1 + Q_1 = \text{Slope} \times t + \text{Intercept} \quad (8)$$

where:

$$\text{Slope} = \left(\frac{Q_2 - Q_1}{t_2 - t_1} \right), \text{Intercept} = - \left(\frac{Q_2 - Q_1}{t_2 - t_1} \right) \times t_1 + Q_1 \quad (9)$$

Additionally,

$$\text{Slope} = \frac{\Delta Q}{\Delta t} = \frac{(\Delta t) / (\sigma_j)}{\Delta t} = \frac{1}{\sigma} \Rightarrow \sigma = \frac{t_2 - t_1}{Q_2 - Q_1} \quad (10)$$

Thus, for a minimum of two pairs of Q and sampling phase data, both the standard deviation σ of the jitter distribution, and an equation for the Q factor at any sampling instant can be determined. Now, if the line end data point is chosen, $t_2 = DJL$, and $Q_2 = 0$, and substituted into Equation 10, the result is

$$\sigma = \frac{t_1 - DJL}{Q_1}$$

rearranging this gives

$$Q_1 = \frac{t_1 - DJL}{\sigma}$$

substituting this into Equation 8 produces

$$Q = \frac{\Delta Q}{\Delta t} \times t - \frac{\Delta Q}{\Delta t} \times t_1 + \frac{t_1 - DJL}{\sigma} \quad (11)$$

$$Q = \frac{1}{\sigma} \times t - \frac{1}{\sigma} \times t_1 - \frac{1}{\sigma} \times DJL \quad (12)$$

which gives

$$Q = \frac{1}{\sigma} \times t - \frac{1}{\sigma} \times DJL \text{ or } Q = \frac{1}{\sigma} \times t - \frac{1}{\sigma} \times t_1 + Q_1 \quad (13)$$

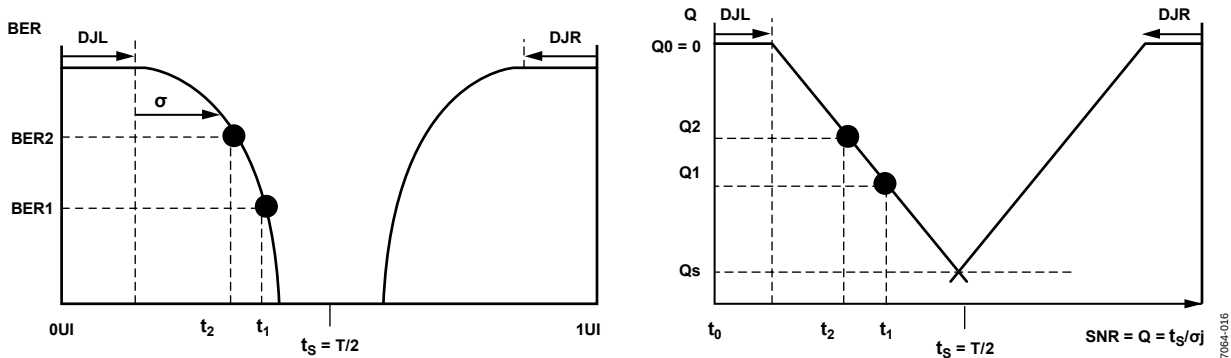


Figure 16. Eye Diagram Statistics Bathtub Curves

Comparing this with Equation 8 gives

$$\sigma = \frac{t_2 - t_1}{Q_2 - Q_1}, \text{ and } DJL = t_1 - \sigma \times Q_1 = -\frac{\text{Intercept}}{\text{Slope}} \quad (14)$$

Given σ and DJL from Equation 14, it can be predicted that the left side Q factor (QL) at the normal decision instant $T/2$ is

$$QL\left(\frac{T}{2}\right) = \frac{1}{\sigma} \times \left(\frac{T}{2} - DJL\right)$$

Referencing Equation 14, the values for σ and DJL are easily computed as $1/\text{slope}$, and Y-intercept divided by the slope. The Q factor equation must be found for the left and right distributions. By knowing left and right Q factors at the normal sample instant, either a lookup table or a known polynomial can be used to find the left and right BER contributions, which can then be added together, according to Equation 7, to find the BER at the normal decision instant.

To estimate the slope for Q vs. sample instant and, consequently, the value for DJ, it is necessary to take at least two sample pairs for (PBER, t) and map these to pairs of (Q, t). In implementations with reasonable amounts of processing capabilities, a better approach is to take many samples of PBER (in the $1e-3$ to $1e-8$ range, for example), and then do an LMS best fit to get the straight line equation for Q vs. t , from which σ (that is, $1/\text{slope}$) and DJL (that is, Y-intercept divided by slope) are determined.

Optimum Sampling Instant

With asymmetric jitter distributions as shown in Figure 17, the actual BER at $T = T/2$ is dominated by the worst-case distribution. Therefore, each side of the distribution/bathtub must be analyzed, and the combined Q estimates must be used to determine the bit error rate at $T/2$.

This information is used to determine the optimum decision instant.

$$T_{\text{Optimum}} = \frac{\sigma_L \times DJR + \sigma_R \times DJL}{\sigma_L + \sigma_R}$$

TAKING SAMPLES FOR A GIVEN PBER

It is important to know how many bits to process to have confidence that the measured PBER value is a good estimate of the actual BER at a given sampling point. To do so, first determine the confidence level for the measured PBER value being within a certain range of the actual BER at a given sampling point. To calculate the exact value for the PBER at any sample instant, one would need to consider the number of errors that occurred over all the data bits transmitted down the link. When only a fixed size sample of data bits is taken, there is a probability distribution that the current fixed size sample of data bits actually has a greater or lesser number of errors than if the total population of data bits is taken.

Each time this experiment is repeated, a different answer is obtained from the actual long-term value with a well defined probability. As sample size is increased (that is, increase the number of pseudo errors collected at any sample point), then the probability of the estimated PBER being within $\alpha\%$ of the exact value increases.

Note that this does not refer to the long-term probability of receiving a bit error here, instead, what is being determining is the probability of the reported number of pseudo errors being within a certain range of the true value, given a certain BER, or probability of bit error, and the number of data bits received.

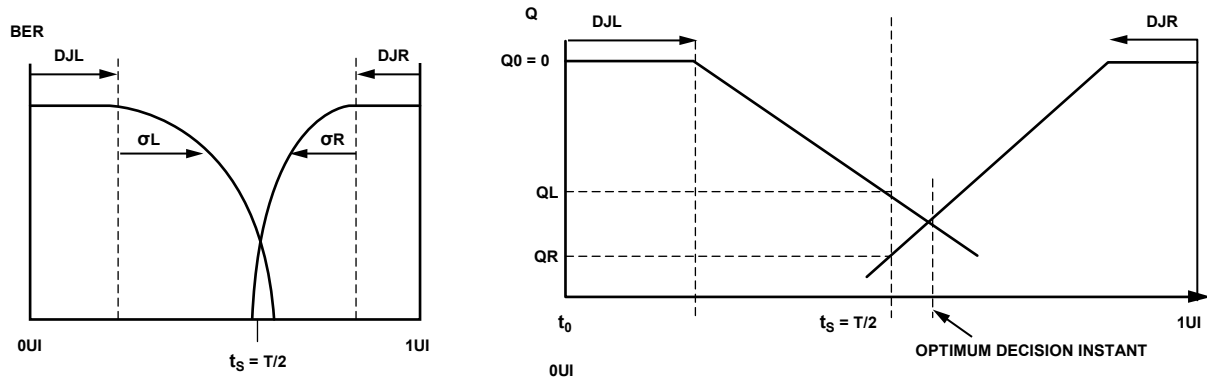


Figure 17. Eye Diagram Statistics Bathtub Curves

For this, the confidence level or cumulative density function is calculated, which is based on the binomial distribution. The function to be computed, using a Poisson approximation, is

$$P(\varepsilon \leq k) = \sum_{k=0}^N P(k) = \sum_{k=0}^N \binom{N}{k} \times p^k \times q^{(N-k)} \approx \frac{(Np)^k \times e^{-Np}}{k!} \quad (15)$$

where:

ε is the number of errors counted.

N is the number of data bits or samples.

$P(\varepsilon \leq k)$ is the probability of receiving less than k errors.

$P(k)$ is the probability of receiving exactly k errors.

p is the probability of receiving an error.

q is the probability of not receiving an error. For further details, refer to Montgomery and Runger in the References section.

This probability is plotted as a function of k in Figure 18, for $N = 1e10$, $BER = p = 1e-8$. It is expected that, on average, there will be 100 errors. From the graph in Figure 18, note that the probability of getting less than 80 errors is less than 2.5%, and the probability of getting more than 120 errors is less than 2.5% also. This is very similar to what is seen with $BER = 1e-3$ and $N = 1e5$. From this it can be concluded that if the number of

bits used in measuring a PBER value is $>100/BER$ then the PBER value obtained has a very high probability ($>95\%$) of being correct to within $\pm 20\%$. Increasing the number of bits used further improves the probability of an accurate measurement.

The problem with Equation 15 is that it is somewhat cumbersome to use. To explore the expected accuracy when $N \times p$ is increased to 1000, (for example, $p = 1e-8$, $N = 1e11$), use the fact that the normal distribution is a very good estimate of a binomial distribution when $N \times p \gg 5$. The mean is given by $\mu = Np$ and the standard deviation is given as $\sigma = \sqrt{npq}$ which at a low bit error rates is closely approximated as $\sigma = \sqrt{\mu}$. Thus, in this example, the mean number of errors is expected to be 1000, and the $\pm 2\sigma$ ($\pm 95\%$) confidence interval to be in the 937 to 1063 range. This means that 95% of all measurement would lie within $\pm 6.3\%$ of the expected value.

In summary, if a sample size of at least $1000/BER$ is always taken, then the measurement of PBER should be very accurate. However, taking a sample size of $100/BER$ also provides reasonable repeatability, and shorter measurement time.

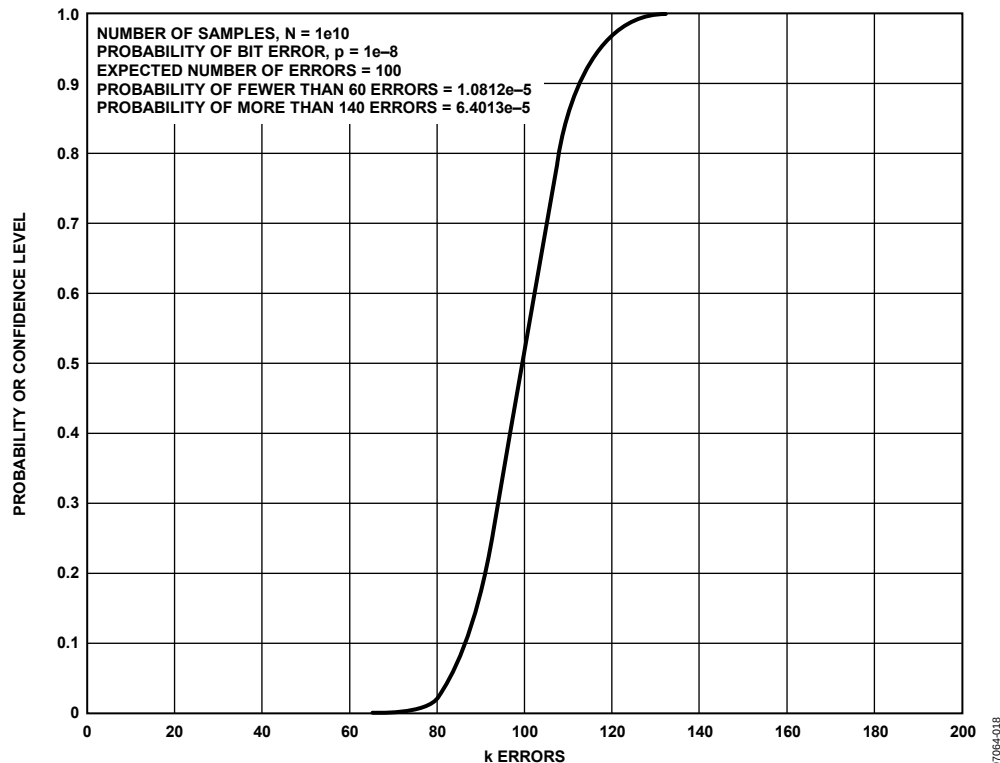


Figure 18. Probability of Getting <K Errors When the Bit Error Rate is $1e-8$ and the Number of Bits is $1e10$

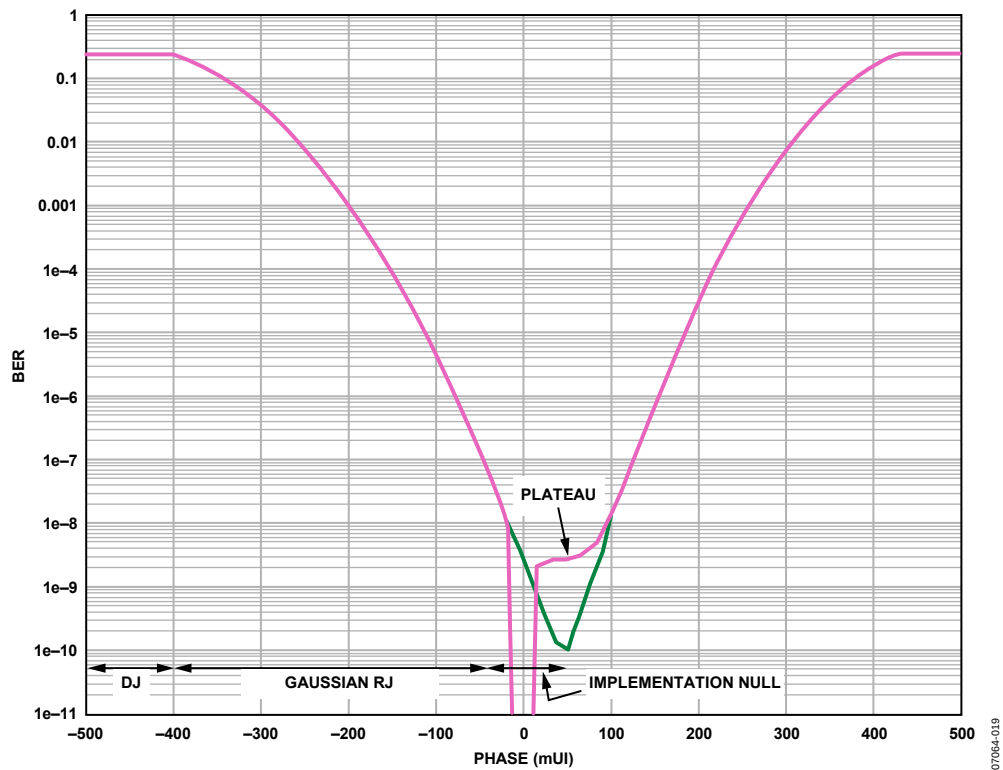


Figure 19. Example Input BER Data (Green), and Reported PBER Data (Pink)

07064-019

PARSING BASED ON EXPERIMENTAL OBSERVATIONS

Figure 19 shows an example input BER characteristic, along with the reported PBER values from the ADN2817. Note that the phase programming resolution is $1/60$ UI = 6° .

The first processing task is to parse the data to remove any non-Gaussian data (that is, exclude the DJ region, and the region affected by the implementation null and plateau). Clearly, the success of this technique relies on the input deterministic BER component being limited to more than one decade above the minimum BER, there being a region of Gaussian data, and there being a region where the reported PBER Gaussian data is not severely affected by the implementation null. There are many ways to perform this task, and one simple parsing algorithm based on experimental evidence is summarized as follows:

1. Discard PBER values that are larger than $1e-2$. As shown in the MJSQ report (see the Fibre Channel publication in the References section) a model for the DJ region has it corresponding to a BER of 0.25. In reality, the BER in the DJ region can be lower than this due to data dependent effects. Therefore, discarding PBER values greater than $1e-2$ removes some of these DJ effects.

2. Discard data that is within 0.1 UI of the NDC sampling phase. For BERs less than $1e-3$, this limits the magnitude of the errors due to the implementation null, yet still gives enough PBER readings less than the above BER limit of $1e-2$.
3. It is not advisable to collect PBER data values less than $1e-9$, as these values take too long to collect. Using a value of $1e-8$ or higher is a better trade off. For low DUT BERs, collecting high confidence data as low as $1e-8$ is important to provide sufficient accuracy.
4. Greatest accuracy is achieved by processing only that remaining data that has the lowest PBERs. For example, if PBER data is available at every phase increment, then using only the five lowest PBER values for BER extrapolation is a good option.

For the input BER and reported PBER data shown in Figure 19, the resulting parsed data is shown in Figure 20. Notice that the number of PBER data points that can be used for extrapolation is greatly reduced.

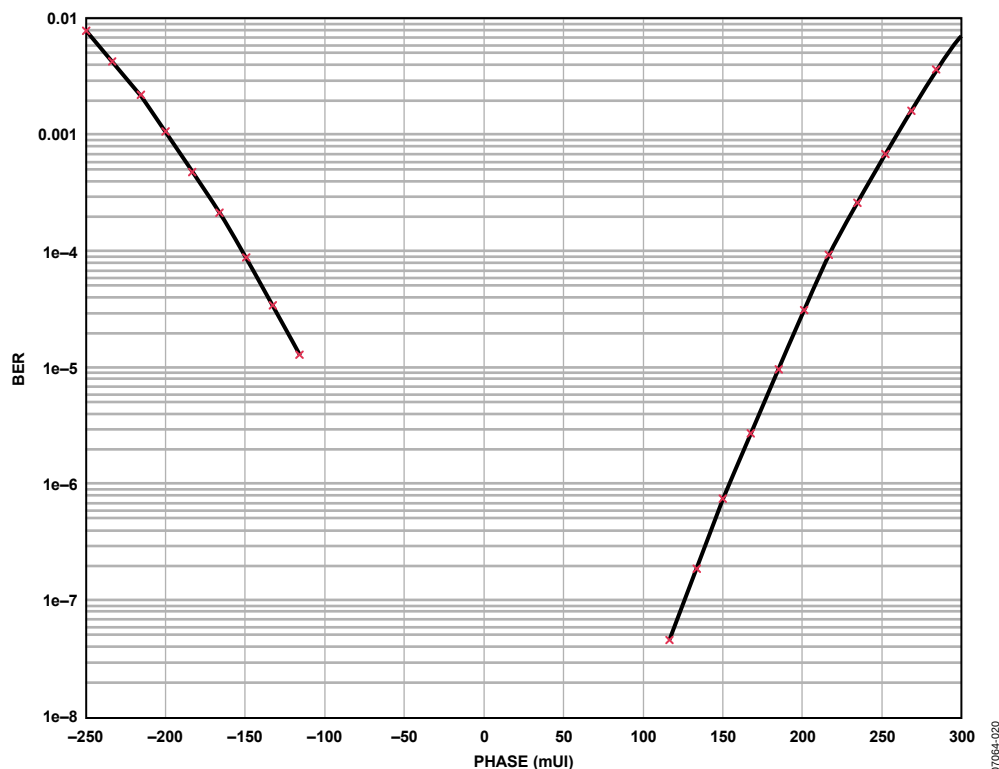


Figure 20. Parsed PBER Data: Remove $1e-9 > \text{BER} > 1e-2$; Remove $|\text{PHASE}| < 0.1$ UI

CONVERTING PARSED PBER DATA TO Q FACTORS

The next step is to convert the parsed PBER data to corresponding Q factors. The Q factor for any PBER value in the range $[1e-2:1e-9]$ from the Gaussian region can be closely approximated using the equation

$$Q = (d0 + d1 \times \log(pber) + d2 \times (\log(pber))^2 + d3 \times (\log(pber))^3)$$

where:

$$d0 = 0.1346273,$$

$$d1 = -1.135047,$$

$$d2 = -0.08386615,$$

$$d3 = -0.003196661.$$

Figure 21 shows the computed Q factors for the data in Figure 19.

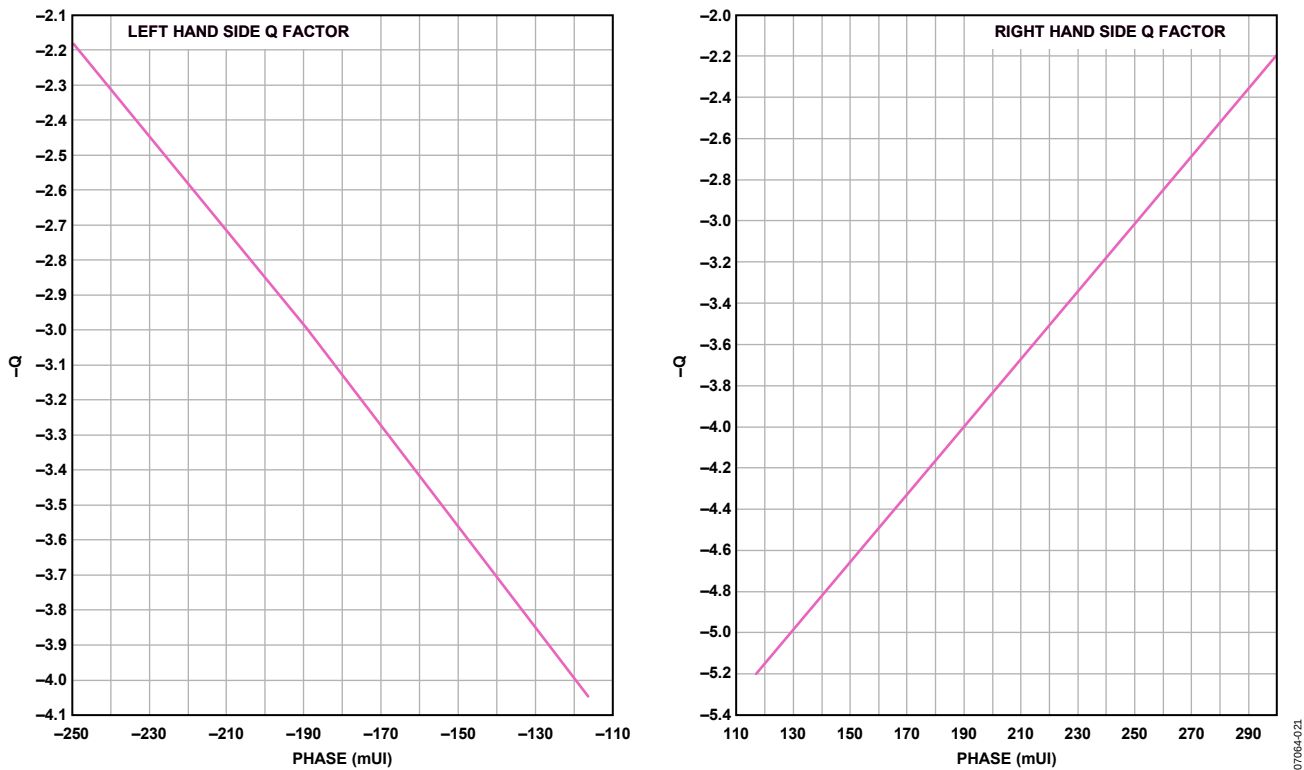


Figure 21. Calculated Q, Over Range of Valid Data. Note that $-Q$ is plotted to reflect conventions.

USE LINEAR REGRESSION TO FIND A LINE EQUATION FOR Q vs. PHASE

Linear regression based on the method of least squares can be used to find the slope and Y-intercept for Q as a function of phase. See Montgomery and Runger in the References section for more details. An example implementation of this regression algorithm is given on the following page (for one side only).

Note that the code also computes the variance of the error and the correlation coefficient. It is strictly the square root of the coefficient of determination as the x_i are not from a set of random variables. These are a measure of how well the data is representative of a straight line. This is also a measure of how well the data used in the BER extrapolation matches the required Gaussian distribution.

```

*****
* xi are the phase values, sum_xi is the sum of all the phase values,
* sum_sq_xi is the sum of all the squared phase values, yi are the Q factors,
* sum_yi is the sum of all the Q factors, sum_sq_yi is the sum of all the
* squared Q factors, sum_xiyi is the sum of all the products, num is the
* number of data pairs.
sum_xi = 0
sum_sq_xi = 0
sum_yi = 0
sum_xiyi = 0
sum_sq_yi = 0
xi = phasel_p
yi = Ql_a
num = pcountl

do loop_phase = loop_phase_start to num
  sum_xi = sum_xi + xi(loop_phase)
  sum_sq_xi = sum_sq_xi + (xi(loop_phase))**2
  sum_yi = sum_yi + yi(loop_phase)
  sum_xiyi = sum_xiyi + xi(loop_phase)*yi(loop_phase)
  sum_sq_yi = sum_sq_yi + (yi(loop_phase))**2
enddo

sxx = sum_sq_xi - (sum_xi**2)/num
sxy = sum_xiyi - sum_xi*sum_yi/num
mean_xi = sum_xi/num
mean_yi = sum_yi/num

print " SLOPE = " slope_l = sxy/sxx
print " INTERCEPT = " intercept_l = mean_yi - slope_l*mean_xi
* Estimate Vaildity of Regression, from Variance of the Error Terms.
SSt = sum_sq_yi - num*(mean_yi**2)
SSe = SSt - slope_l*sxy
SSr = slope_l*sxyR_sq = SSr/SSt
variance_of_error = 1e9
if (num > 2 ) variance_of_error = abs(SSe/(num-2))
print " CORRELATION COEFFICIENT = " R = sqrt(R_sq)
print " VARIANCE OF ERROR = " var_e = variance_of_error
*****

```

SSr/SSt is a measure of how good the line equation is at estimating the change in yi for a given change in xi.

- SSr is a measure of the amount of change in yi that is expected from the line equation $y = mx + c$, for any change in xi. It is a measure of the predictable amount of change in yi.
- SSe is a measure of the amount of change in yi that cannot be explained by the model.
- SSt is a measure of the total change in yi due to a change in xi, or more accurately, the total corrected sum of squares.

Figure 22 shows the left and right extrapolated Q factors for the data in Figure 19. Note that the figure plots $-Q$ vs. phase, to conform to convention. The green trace is a plot of the left-hand side equation $q = \text{intercept_l} + \text{slope_l} \times \text{phase}$, where intercept_l and slope_l are as computed above. The pink trace is the corresponding right Q factor equation.

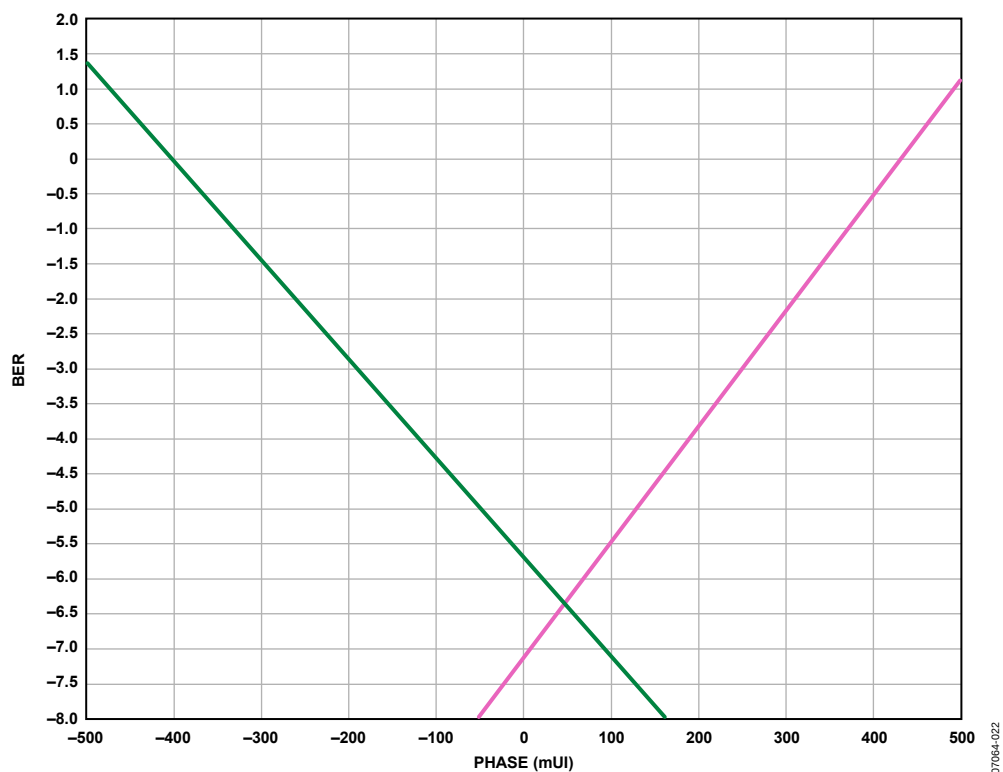


Figure 22. Extrapolated Q Factors for Data in Figure 19 Based on Application of Linear Regression to the Measured Q Factors

CONVERT EXTRAPOLATED Q FACTOR TO A BER VALUE AT THE NORMAL DECISION INSTANT

Using the slope and intercept, the value for the Q factor for both the left and the right distribution can be calculated at the normal sampling phase, (phase = 0 in Figure 22). The left and right values of the extrapolated Q factor at the normal sampling instant (such as Ql and Qr) are used to estimate the BER at the normal sampling instant as

$$BER = \frac{1}{4} \operatorname{erfc}\left(\frac{Q_l}{\sqrt{2}}\right) + \frac{1}{4} \operatorname{erfc}\left(\frac{Q_r}{\sqrt{2}}\right) = \frac{1}{4} \operatorname{erfc}\left(\frac{5.7}{\sqrt{2}}\right) + \frac{1}{4} \operatorname{erfc}\left(\frac{7.05}{\sqrt{2}}\right)$$

Because the $\operatorname{erfc}()$ function is not readily available, the calculation is performed with the help of a polynomial fit for the $\operatorname{Log}(\operatorname{BER})$ vs. Q factor. Thus,

* Left $\operatorname{Log}(\operatorname{BER})$

$$\operatorname{Logberl} = (c_0 + c_1 \times (ql) + c_2 \times (ql^2) + c_3 \times (ql^3))$$

* Right $\operatorname{Log}(\operatorname{BER})$

$$\operatorname{logberr} = (c_0 + c_1 \times (qr) + c_2 \times (qr^2) + c_3 \times (qr^3))$$

* Extrapolated BER value.

$$\operatorname{ber} = 10^{(\operatorname{logberl})} + 10^{(\operatorname{logberr})}$$

where:

$$c_0 = -0.6270002$$

$$c_1 = -0.2820845$$

$$c_2 = -0.1838629$$

$$c_3 = -0.001688561 \text{ is valid for Q factors in the range of 0.2 to 8.}$$

Figure 23 shows a comparison of the input and the reported and extrapolated BER values computed using this algorithm.

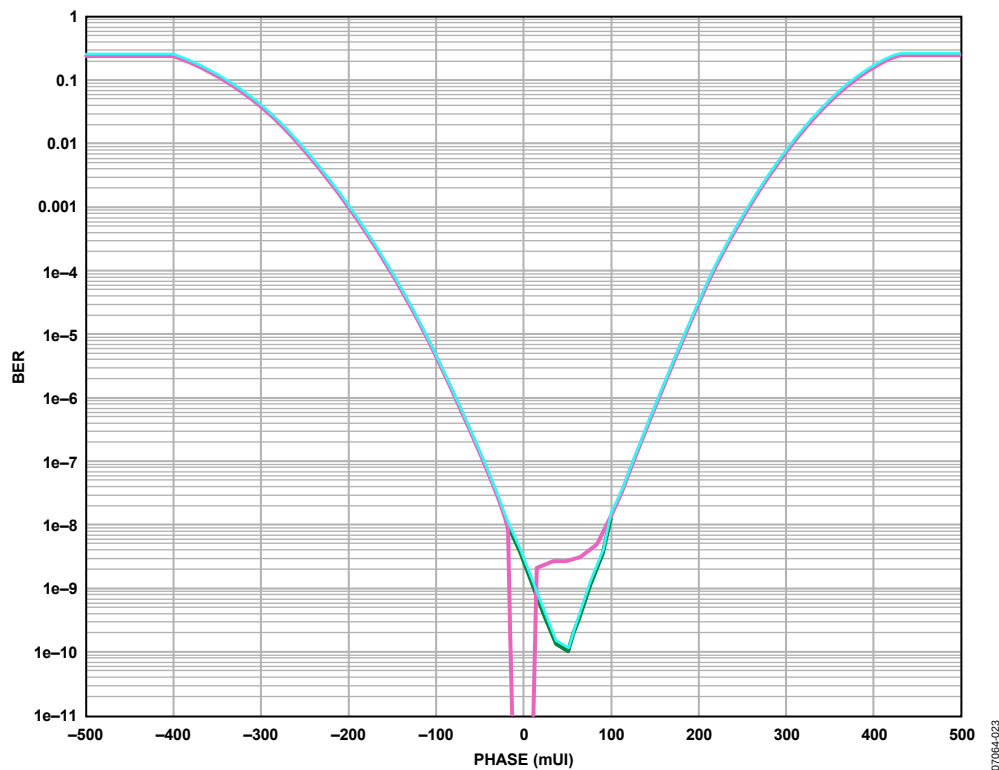


Figure 23. Plot of Actual Input BER Data (Green), Reported Pseudo BER data (Pink) and Extrapolated BER data (Blue) Based on Suggested Algorithms

HOW TO DETECT WHEN THE BER > $1e-3$

When the actual input BER is above $1e-3$, then there is very little valid Gaussian data available to use for BER extrapolation. With reference to Figure 10, note when the PBER at 0.1 UI becomes greater than $\sim 5e-3$. This can be taken as an indication that the actual BER is greater than the upper performance limit of $1e-3$.

MATLAB CODE

Example processing algorithms are available and implemented in MATLAB®. These routines are written to provide an introductory tutorial to the methods required for BER extrapolation, as well as to provide the user with a means of quickly getting started with the ADN2817 BER monitor feature. The routines are not intended to be robust in all applications, but do provide a useful starting point for the user's own code development.

To use these routines the user must provide a means of collecting pseudo BER data from the ADN2817 over the I²C, and storing it in a file called pber.dat. Alternatively, to test the algorithms in the absence of real data the user can use the MATLAB M file gen.m to create a user-defined pseudo BER characteristic. The file alg.m reads pairs of data (phase in UIs,

pseudo BER) in from the file called pber.dat. The previously described algorithms are applied to the data, graphs of relevant information are plotted, and the BER estimate is produced.

A routine called genalg.m essentially calls both the routines gen.m and alg.m, and provides an estimate of the accuracy of the algorithms, given the ideal input characteristic. It is suggested that the user start with the routine genalg.m as a means of quickly getting familiar with the processing algorithms. To do so:

1. Copy the gen.m, alg.m, and genalg.m routines to a working directory on a PC or UNIX platform with MATLAB installed.
2. Start MATLAB and type **genalg** at the command prompt.
3. Follow the instructions that display and observe the plots in Figure Windows 1 through 3.
4. Type **alg** in MATLAB to process your own BER monitor data from a file named pber.dat.
5. Type **gen** to generate example pseudo BER data in a file called pber.dat.

REFERENCES

The references listed here are cited in this application note. Many of these references can be accessed online.

Agilent Technologies. “Jitter Analysis: The Dual Dirac Model, RJ/DJ, and Q-Scale.”, White paper.

Bergano, N., F. Kerfoot, and C. Davidson. 1993. “Margin Measurements in Optical Amplifier Systems.” *IEEE Photonics Technology Letters*, Vol. 5, No. 3, (March).

Common Electrical I/O (CEI)–Electrical and Jitter Interoperability Agreements for 6 G + bps and 11 G + bps I/O, Optical Internetworking Forum, Annex 2.C.4.

Fibre Channel–Methodologies for Jitter and Signal Quality Specification, June 2004.

Galloway, F. H. and T. Knibbe, “Rapid Testing of Extremely Low BER”. Presented at Optical Network Interface Design Symposium, Agilent Technologies.

Gooding, D. J. 1968. “Performance Monitor Techniques for Digital Receivers Based on Extrapolation of Error Rate.” *IEEE Transactions on Communication Technology*, Vol. COM-16, No. 3 (June).

Keelty, J. and K. Feher. 1978. “On-Line Pseudo-Error Monitors for Digital Transmission Systems.” *IEEE Transactions on Communication Technology*, Vol. COM-26, No. 8, (August).

Lindsay, Thomas. “Innovations in BER Testers Enable Fast and Accurate Eye Diagram, Eye Mask, Q-Factor and Jitter Measurements.”

Montgomery, D. C. and G.C. Runger, “Applied Statistics and Probability for Engineers”, 3rd edition. New York: John Wiley & Sons.

Newcombe, E. and S. Pasupathy. 1980. “Error Rate Monitoring in a Partial Response System.” *IEEE Transactions On Communication Technology*, Vol. COM-28, No. 7, (July).

Palacharla, P., J. Chrostowski, and R. Neumann. 1995. “Techniques for Measurement of Low Bit Error Rates in Computer Data Links.” *Conference Proceedings of the 1995 IEEE Fourteenth Annual International Phoenix Conference on Computers and Communications*, 28-31 (March): 184-190.

Y. Cai, SA Werner, GJ Zhang, MJ Olsen, RD Brink, 2002. “Jitter Testing for Multi-Gigabit Backplane SERDES - Techniques to Decompose and Combine Various Types of Jitter”, *Proceedings of the IEEE International Test Conference*.

APPENDIX A: SUMMARY OF JITTER STATISTICS THEORY

This appendix summarizes the normal (Gaussian) statistics theory used to determine bit error ratios resulting from the sampling of jittery data. Figure 24 represents the data eye edge crossing jitter probability distributions around 0 UI and 1 UI boundaries, assuming only random jitter. Each probability density function is given as

$$PDF = \frac{1}{\sigma\sqrt{2\pi}} \exp - \left[\frac{(t - \mu)^2}{\sigma^2} \right] \quad (16)$$

The bit error rate at the sampling instant t_s is given as

$$BER = P0 \times PE0 + P1 \times PE1$$

where:

$P0$ is the probability that there was a transition scheduled to occur at the 0 UI boundary.

$P1$ is the probability that there was a transition scheduled to occur at the 1 UI boundary.

$PE0$ is the probability that the 0 UI transition occurred to the right of sampling instant t_s .

$PE1$ is the probability that the 1 UI transition occurred to the left of sampling instant t_s .

It is normally assumed that $P0 = P1 = 1/2$. Thus,

$$BER = \frac{1}{2} \times PE0 + \frac{1}{2} PE1 \quad (17)$$

$PE0$ is determined as the integral of the left normal distribution from t_s to infinity,

$$PE0 = \int_{t_s}^{\infty} \frac{1}{\sigma\sqrt{2\pi}} \exp - \left[\frac{(t - \mu)^2}{\sigma^2} \right] dt \quad (18)$$

A change of variable approach is used to convert this into something that can be evaluated.

$$\frac{d}{dt} \left(\frac{(t - \mu)}{\sigma\sqrt{2}} \right) = \frac{1}{\sigma\sqrt{2}} \quad (19)$$

$$\Rightarrow dt = \sigma\sqrt{2} \times d \left(\frac{(t - \mu)}{\sigma\sqrt{2}} \right) \quad (20)$$

substituting Equation 20 into Equation 18 gives

$$PE0 = \int_{t_s}^{\infty} \frac{1}{\sqrt{\pi}} \exp - \left[\frac{(t - \mu)^2}{\sigma^2} \right] d \left(\frac{(t - \mu)}{\sigma\sqrt{2}} \right) \quad (21)$$

To allow use of standard solutions the independent variable is changed from t (time) to x (generic) by letting

$$x = \frac{(t - \mu)}{\sigma\sqrt{2}}$$

To map any point in the t domain to the x domain, subtract μ , and multiply by $1/(\sigma\sqrt{2})$.

Therefore, the lower integration value of t_s must get mapped to $(t_s - \mu)/(\sigma\sqrt{2})$. With these two substitutions, Equation 21 now becomes

$$PE0 = \frac{1}{\sqrt{\pi}} \int_{\left(\frac{t_s - \mu}{\sigma\sqrt{2}} \right)}^{\infty} \exp - x^2 dx \quad (22)$$

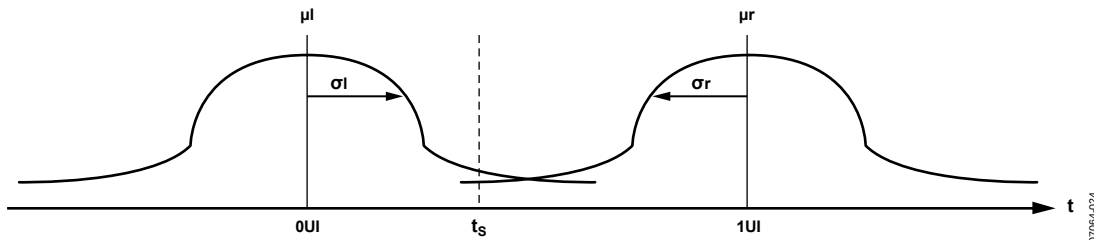


Figure 24. Example Data Eye Edge Crossing Probability Density Function

There is no closed form solution to the integral in Equation 22, however numerical tables and polynomial approximations of the complementary error function are well known. The complementary error function is defined as

$$\operatorname{erfc}(z) \equiv \frac{2}{\sqrt{\pi}} \int_z^{\infty} \exp -x^2 dx \quad (23)$$

Substituting this into Equation 22 gives

$$PE0 = \frac{1}{2} \operatorname{erfc}\left(\frac{ts - \mu}{\sigma \sqrt{2}}\right) = \frac{1}{2} \operatorname{erfc}\left(\frac{Q}{\sqrt{2}}\right) \quad (24)$$

where:

$$Q = \left(\frac{ts - \mu}{\sigma}\right) \quad (25)$$

To get the equation for the BER, consider the impact of both the left and right distributions in Figure 24, and in Equation 17.

This gives

$$BER = \frac{1}{2} \times \frac{1}{2} \operatorname{erfc}\left(\frac{ts - \mu_l}{\sigma_l \sqrt{2}}\right) + \frac{1}{2} \times \frac{1}{2} \operatorname{erfc}\left(\frac{ts - \mu_r}{\sigma_r \sqrt{2}}\right) \quad (26)$$

$$BER = \frac{1}{4} \operatorname{erfc}\left(\frac{Q_l}{\sqrt{2}}\right) + \frac{1}{4} \operatorname{erfc}\left(\frac{Q_r}{\sqrt{2}}\right) \quad (27)$$

With Equation 27, if the mean and standard deviation of the normal jitter statistics for the data eye are known, the BER can be computed.

