AN ACCELERATED JITTER TOLERANCE TEST TECHNIQUE ON ATE FOR 1.5GB/S AND 3GB/S SERIAL-ATA

Y. Fan[1], Y. Cai[1], L. Fang[1], A. Verma[1], W. Burchanowski[1], Z. Zilic[2], S. Kumar[1]

[1] Agere Systems
1110 American Parkway NE,
Allentown, Pennsylvania 18109
Email: yicai@agere.com

[2] Department of ECE, McGill University

ABSTRACT

In our previous publication [1], we demonstrated the ability to generate the proper mix of jitter on ATE to enable the jitter tolerance test for 1.5/3Gbps SATA applications. Obviously this is NOT the only challenge for performing this test on ATE. Jitter tolerance compliance test for SerDes calls for validation of bit-error-rate (BER) down to the 10^{-12} or lower. This requirement deemed this test to be extremely time-consuming, which normally takes more than an hour (assuming running 10^{13} bits for 10^{-12} BER level guaranteed). While in the ATE world every test is measured in seconds or even in milliseconds; it is obviously impractical to adopt this test directly. In this paper we demonstrate a new technique to perform the jitter tolerance test >1000 times faster. The technique of course involves extrapolation from the higher BER region down to the 10^{-12} level for the compliance test, but the challenge that we faced in getting the extrapolation is very different from the conventional transmitter jitter measurement world. We present a new mathematical model suitable to reason about this extrapolation process.

1. INTRODUCTION

The traditional jitter tolerance test has always been very challenging. There are two outstanding issues– the long test time, and the complexity to generate a controlled amount of jitter with the proper mix of different types of jitter [9].

The jitter tolerance test is notorious for its long test time. Since most standards for serial links define jitter tolerance performance down to the 10^{-12} BER, to check that BER level, we need to run 10^{13} bits. Even at 1.5Gbps serial data rate, it takes 111 minutes (~2 hours) for the device to run so many bits. That is the fundamental limit for running this test fast. With some applications on the trend demanding 10^{-14} BER, direct measurement is not even practical.

Knowing that the Clock Data Recovery (CDR) design is the key differentiator for a serial link, many companies have to perform very thorough characterization testing. That includes analysis across process corners, voltage limits and temperature ranges (known as PVT corners). The characterization test on ATE is known for its high throughput to provide such performance analysis across PVT on a large sample size. Our aim is to invent a way to test jitter tolerance in a much faster manner on ATE. In this paper we demonstrate a new technique to perform the jitter tolerance test >1000 times faster. The approach is straightforward: varying the amount of input jitter to get the receiver into several higher BER regions, and then extrapolating down to the 10^{-12} BER specification for jitter tolerance. The conceptual illustration is shown in Fig.1. We will present the assumptions, extrapolation models and experimental data in the rest of the paper.

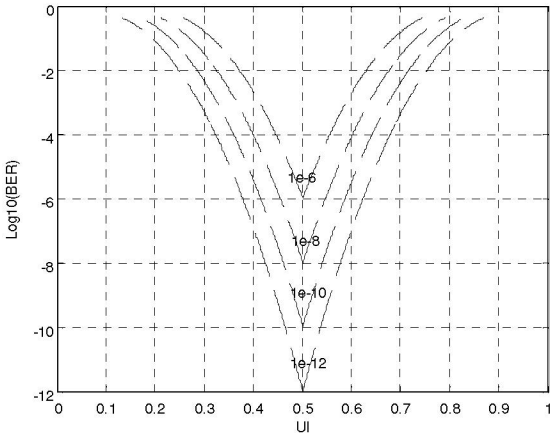


Fig.1 *Conceptual illustration of the jitter tolerance extrapolation.*

The second challenge for jitter tolerance test is to generate different kinds of jitter and mix them together. For Serial-ATA applications, the jitter tolerance test requires a proper amount of deterministic jitter, random jitter and periodical jitter. This is becoming a norm for most of the point-to-point serial links using backplanes or cables [2,3,4,5]. It is a departure from the traditional jitter tolerance test used for long haul SONET style links with cascaded repeaters, where the sinusoidal jitter tolerance and transfer characteristic are the

1-4244-0292-1/06/$20.00

main concerns. In Serial-ATA, SAS and XAUI applications, there is NOT much concern about jitter transfer, but a very elaborate jitter tolerance test is required with InterSymbol Interference (ISI), Duty Cycle Distortion (DCD), and Periodic Jitter (PJ). The reason is that the given transmission media generates these specific jitter types.

Even for laboratory use, we need several instruments to produce the different types of jitter and combine the results [9]. The new Voltage Controlled Delay Line (VCDL) based modulation can generate a much higher frequency PJ (several hundred MHz vs. the 10~20MHz in a FM based system). ISI+DCD are still introduced by adding cable or backplane of various lengths, and RJ is tuned with a white voltage noise generator.

On ATE, supplying the proper mix and amount of jitter is always more challenging [6,7,8]. Therefore, it is not readily available. Few years ago, most of the people would have to use ATE and a bench hybrid – integrating a pattern generator and an FM clock source to supply the proper PJ [6]. Although adding ISI and DCD is possible with cable equivalent filters [7,8] and RJ with white noise generators, it becomes hard to actually implement it on the ATE system and a loadboard. That is the reason why there are no publications supporting that direction.

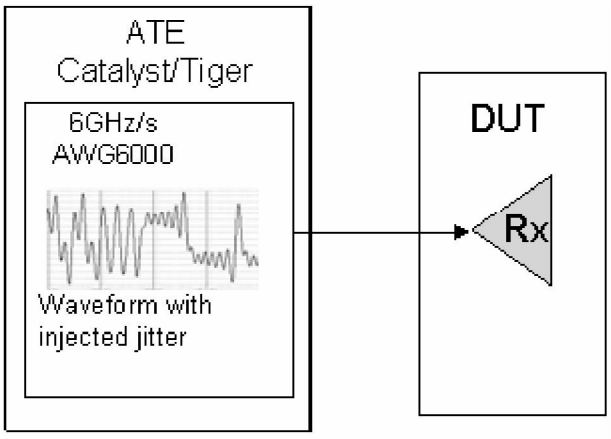


Fig.2 *Waveform generation for jitter tolerance test*

In our previous publication [1], we presented an AWG-based jitter injection technique that can produce very flexible combinations of ISI, DCD and PJ. Essentially, almost every type of jitter is embedded in the waveform. However, because the AWG waveform is of finite length, it is NOT possible to produce programmable Random Jitter (RJ). Of course the AWG sampling clock has its intrinsic RJ. We believe that this is acceptable for the jitter tolerance compliance test with the intrinsic AWG RJ; we only need to add the proper amount of ISI, DCD and PJ. In this way, we can achieve the jitter injection requirement with only one piece of equipment – AWG, as shown in Fig.2. It makes the test setup and loadboard simple. There is no intermediate add-on circuit, which means we do NOT have to switch anything out for the functional test and the input sensitivity test where clean signals are used. The AWG approach can also produce PJ of very high frequency, like the more modern VCDL modulators. This is needed for testing CDR out-of-band jitter tolerance when the CDR can no longer track the input PJ.

2. NEW METHOD TO ACCELERATE JITTER TOLERANCE TEST ON ATE

Since accelerating the jitter tolerance test is already a new and challenging concept, not to mention performing it on ATE, there are NOT many established directions to follow. Therefore, it is natural to start from borrowing ideas for transmitter jitter measurements.

We briefly browse through the popular jitter extrapolation techniques for potential ideas. The histogram tail-fitting with under-sampler, real-time sampling, and Time-Interval-Analysis require the knowledge of the actual probability density function (PDF). This is NOT as accessible for Rx CDR as the Tx output. Only the Bit-Error-Rate scan (better known as bathtub curve) uses a model based on certain assumptions.

In a Tx jitter measurement with a BER-scan, a BER-scan/bathtub-curve is usually used. As shown in Fig.3, a bathtub curve is a plot of data eye openings at various BER thresholds. The finite slope of the bathtub curve is caused by the existence of statistically random jitter. Obviously, at a lower BER, the eye opening becomes narrower.

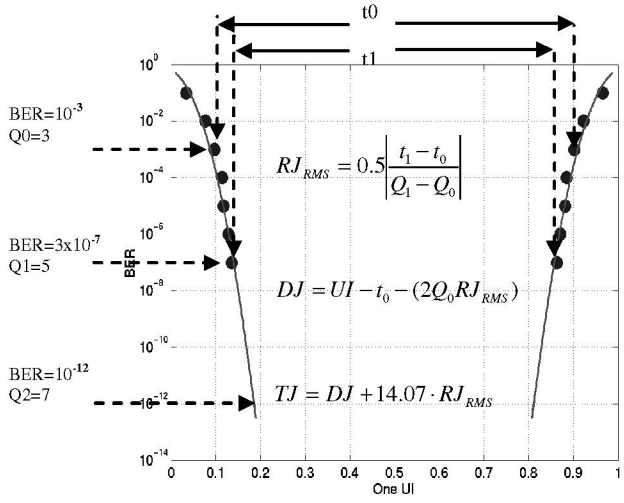


Fig.3 *Schematic diagram of a BERT scan bathtub curve, showing the simplified formula to calculate DJ, RJ and TJ with only two data points recommended by the XAUI standard.*

In the presence of both DJ and RJ, the DJ component has its own PDF, and the combined Total Jitter (TJ) PDF is a convolution of the DJ and RJ PDFs. In reality, for Tx jitter measurement, all we can see is the combined jitter profile, and we need to work backwards to derive its DJ and RJ components. That is nearly impossible if we do NOT make some assumptions and use a simple model. The most popular simple model used is the so called "double delta" model – assuming that the only DJ is DCD, whose PDF is comprised of only a pair of delta functions. In that case, the complicated convolution is reduced to a standard complimentary error function. This serves as the model for modern bathtub curve fitting. It is an overlooked fact that the bathtub curve approach is derived from a simple double delta assumption. However, this seemingly limited model worked reasonably well when used appropriately (i.e. proper selection of the curve fitting range). There are studies on the effect of the DJ profile when deviating from this assumption [10].

Even though there have been many arguments concerning this technique, citing that it is theoretically NOT the most flexible model for arbitrary jitter PDF [10], this technique is still favored by many people because it directly links to system level performance at 10^{-12} BER. Although the double-delta DJ assumption is arguably limited, the level of accuracy can be easily verified by performing the test at a lower BER range, and comparing it to the curve from extrapolation. Therefore we identified the BER-scan as the potential candidate from which to borrow ideas for our jitter tolerance test acceleration.

The Rx-BER-scan for jitter tolerance has several similarities to the Tx-BER-scan:

(1) Collect data points at higher error rates, which take much less time to do. Then extrapolate performance to the lower error rate.

(2) It is based on a simple model for curve fitting. A black box approach is used, without detailed knowledge of the shape of the ISI jitter distribution.

(3) Just like Tx jitter extrapolation, the level of accuracy can be easily verified by performing the test at the lower BER range, and comparing it to the curve from extrapolation. If the error is small, then the new method is self validated, because the BER rate is the most direct measure of device performance as defined in the specifications.

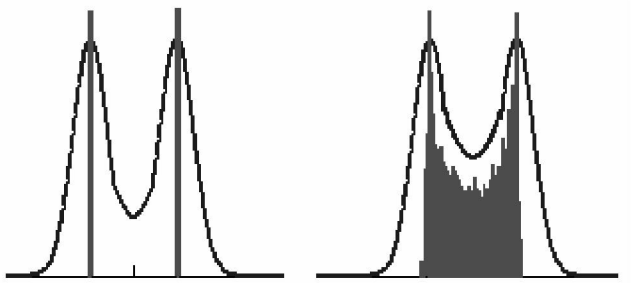
The Rx-BER-scan for jitter tolerance has several significant differences from the Tx-BER-scan:

(1) It is no longer going to be a bathtub curve-fit. If we look at a high BER point with a lot of jitter, the bathtub curve will bottom out (lines crossed) at that error rate. What we will get is a series of crossed curves, as shown in Fig.1. Therefore, a new extrapolation needs to be developed for the jitter tolerance test.

(2) Unlike the Tx jitter test, where the signal under test is exposed, for Rx there is no visibility on the jitter types inside device CDR. Therefore, any extrapolation needs to be done with some assumptions and a behavior model.

(3) Unlike the Tx jitter test, where we have no control over the jitter probability distribution, Rx jitter tolerance specs do NOT define the shape of the jitter probability distribution. Normally the standards spell out separate DJ, PJ and RJ limits in the TJ composition. In the jitter tolerance test we do have some flexibility to shape the probability distribution function. This is an important property, because there is evidence that some types of jitter probability distribution function will affect the curve-fit accuracy when we use the complimentary error function to model the curve [10].

In light of the assumption that a double delta PDF profile will give a better curve fit, we realize that the PDF for a single tone sine wave modulation is very much like a double delta distribution as shown in Fig.4. This means that there is a strong tendency to favor the two edges and have not much in the center in the PDF curve. Therefore, we elected to start with a single tone PJ injection, which we would expect to be a closer fit to the complimentary error function.



(a) Double delta PDF *(b) Sine wave modulation PDF*

***Fig.4** Jitter PDFs*

On the other hand, we are NOT very concerned with other DJ PDF profiles that have ISI presence, because in the Tx jitter bathtub curve technique, we can still achieve good accuracy with many kinds of jitter distribution.

3. TEST IMPLEMENTATION

3.1 Generate controlled amount of jitter mix

In this section, we will illustrate how we are maximizing the signal integrity and then injecting a precise amount of jitter and calibrate the accuracy of the resulting signal.

3.1.1 Signal integrity improvement with equalization

To perform a jitter tolerance test, we need source signals with controllable jitter. In our implementation, the source signals are generated by modulating ideal AWG binary signals with any user defined jitter profile. Generally speaking, we can

source any waveform with spectral content limited to the Nyquist band. Jitter injection does not require additional instruments as in some other setups [1,5,6]. To minimize the baseline deterministic jitter of the test signal, equalization is used to reduce the distortion due to the non-ideal AWG channel response.

With the state of the art AWG on ATE – 6Gs/s and 1.9GHz analog bandwidth, we barely have 2 samples per bit for a 3Gb/s SATA/SAS with attenuated 3rd harmonic. In order to maximize the analog bandwidth of the AWG, the instrument manufacturer (Teradyne) added a filter bypass path on their AWG6000 upon our request. That alone has improved signal integrity significantly for 3Gb/s data generation. Although the filter bypass mode enabled us to perform a 3Gb/s functional test, the bandwidth is still slightly limited for conducting an at-speed input sensitivity test and a jitter tolerance test. We overcame the problem with equalization even with the very limited 2 samples per bit.

No external instrument is used for our equalization process. We are taking advantage of the significantly higher analog bandwidth of the sampler (9GHz) integrated on the ATE to equalize the AWG signal and signal delivery path. A chirped signal with broad frequency content is generated by the AWG, and routed to the sampler (Teradyne GigaDig) through the DUT interface board (DIB/loadboard), as illustrated in Fig.5.

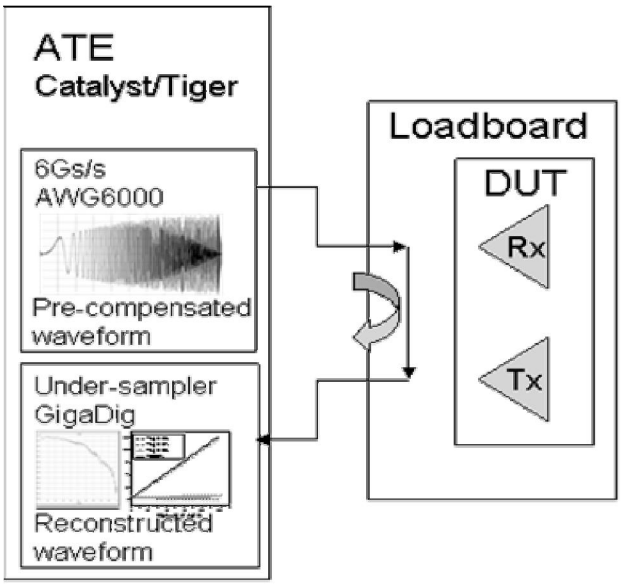
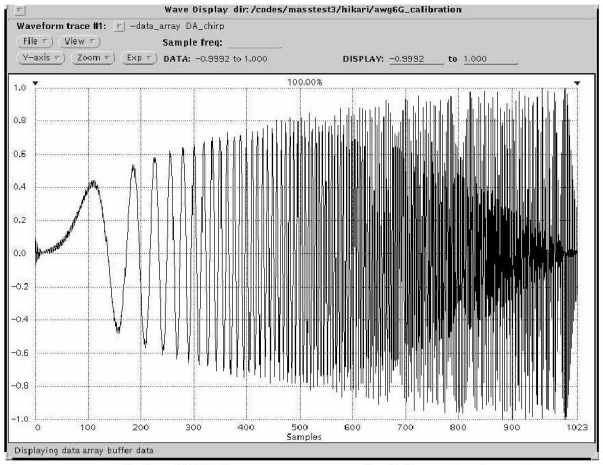
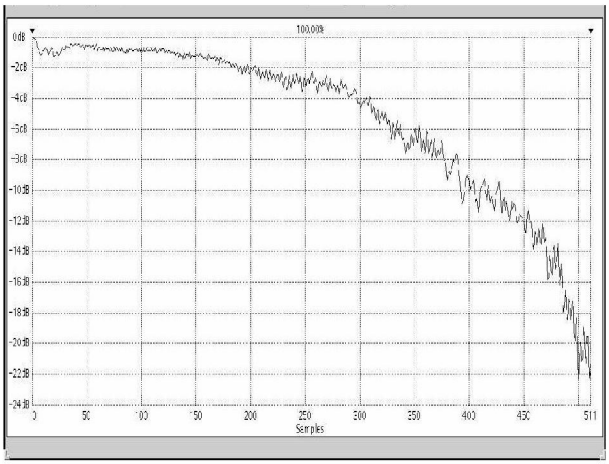


Fig.5 *Equalization/calibration setup for AWG*

A chirped signal sampled at 6GS/s containing spectral content up to 3GHz is sourced from the AWG and captured with the high-bandwidth, undersampling digitizer. By comparing the spectral content of the chirped signal with the captured signal, the transfer function of the AWG channel is obtained, as shown in Fig.6. The inverse transfer function is used to pre-compensate the 128-bit serial-ATA PRBS pattern. This pre-compensation accounts for both the AWG's frequency response and signal delivery path limitations.



(a) AWG programmed chirped signal

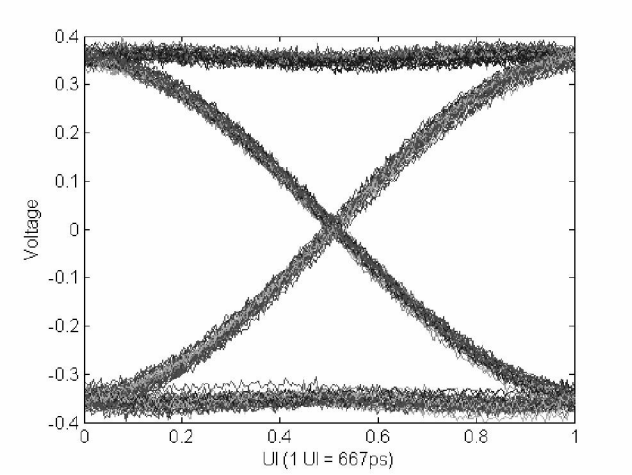


(b) Calculated frequency response of the AWG path

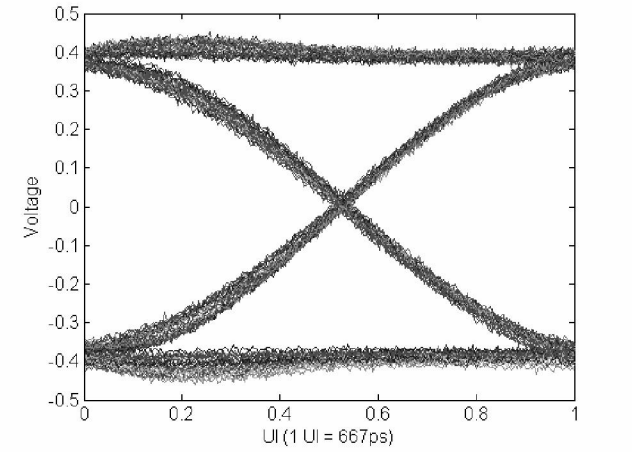
Fig.6 *AWG equalization/calibration waveforms*

When creating analog waveforms with an AWG, the spectral content can be arbitrarily programmed by software filtering up to half of the sampling frequency. If the frequency response of the AWG channel is known, it can be equalized. Constant 0dB magnitude and linear-phase responses can be achieved up to the Nyquist frequency by equalization.

The effect of this equalization is very significant for a 3Gb/s signal - we can achieve 0.07UI TJ noise floor. For a 1.5Gb/s signal, the equalization only makes a small improvement, because there is NOT much ISI (much less bandwidth limitation) as shown in Fig.7. This equalization is of course only effective for reducing ISI, instead of RJ. As the eye-diagrams have shown, the remaining jitter is mostly RJ.



(a) Waveform with equalization



(b) Waveform without equalization

***Fig.7** Captured waveforms (no injected jitter)*

3.1.2 Jitter injection

The injected jitter signal can have any arbitrary profile. In this section, we demonstrate how different amounts of sinusoidal jitter signals are generated. In our implementation, AWG6000 is used and its sample rate is set at 6G samples/s. Each data bit has two samples for 3G SATA tests and four samples for 1.5G SATA tests. The following discussion is based on 1.5G tests, but it also applies to 3G tests.

As the length of an AWG sequence is finite, the injected jitter signal needs to be coherent with the AWG signal. When 128-bit PRBS test signals are used, the frequency resolution is given by

$$\frac{6000MHz/Sample}{128bitsX\,4Samples/bit}=11.71875MHz$$

We can inject any sinusoidal jitter signal that is a multiple of 11.71875MHz. If lower frequency jitter injection is needed, we can obviously double the AWG pattern length by repeating the 128-bit PRBS, and make PJ coherent to half of the frequency resolution. In our experiments, we investigated the jitter tolerance characteristic at different frequencies while concentrating on the jitter frequency of 93.75MHz (= 8 x 11.71875MHz) for most of our work, because this frequency is considered out of band jitter for the CDR PLL tracking band.

Basically, jitter injection involves the following five steps:

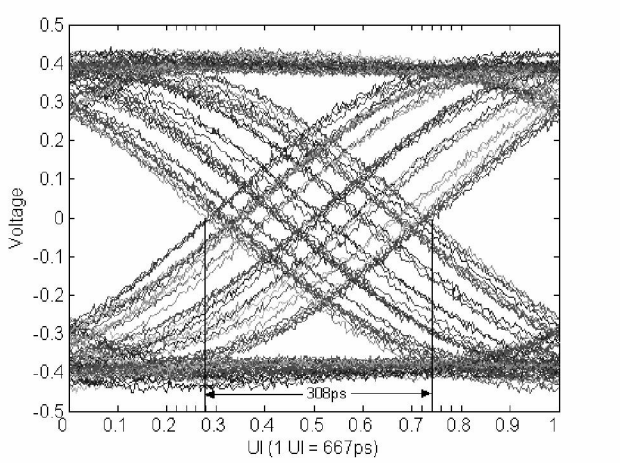
(1) Create a digitized sinusoidal signal of proper amplitude and frequency representing the PJ we intend to inject

(2) Oversample the ideal binary data stream and add widowing

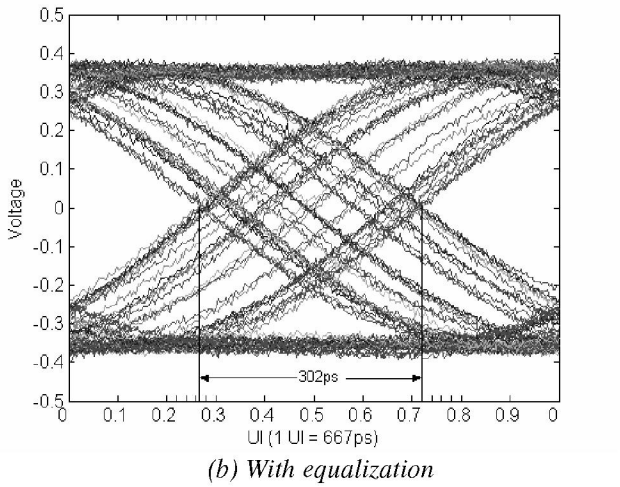
(3) Modulate data edges, converting jitter amplitude information to timing information by moving the edge of the data signal based on the jitter amplitude information

(4) Apply the inverse equalization transfer function as discussed in section 3.1.1

(5) Filter out components above the Nyquist frequency and decimate the waveform to get the desired AWG samples.



(a) Without equalization



(b) With equalization

***Fig.8** Eye diagrams of the AWG generated 1.5Gbs with 300ps PJ injected captured by the sampler in calibration.*

Fig.8 captures the eye diagrams of the AWG output signal with injected jitter. As can be seen, our jitter injection technique is very accurate. It also once again demonstrates that the equalization process reduced the baseline jitter. The whole scheme generates the exact test signals, which are essential for the jitter tolerance test.

We use the same calibration setup as the AWG equalization - shorting the AWG to the digitizer, to calibrate the amount of jitter injected. Fig.9 shows a good linear control on our jitter injection. However, as described in our previous publication [1], the RJ measurement noise floor is higher on ATE because of the sampler clock jitter. That led us to report an exaggerated RJ (and hence TJ) number. In order to report a more accurate jitter tolerance number, we used a Wavecrest SIA-3000 to calibrate the TJ, RJ and DJ number that we injected. As shown in Fig.10, the Wavecrest measured TJ and DJ (ISI+DCD+PJ) have a constant slope and a vertical offset to our programmed PJ. The difference in RJ measurement between ATE and the Wavecrest explains the offset. The final jitter tolerance number we report will be derived from this calibration curve.

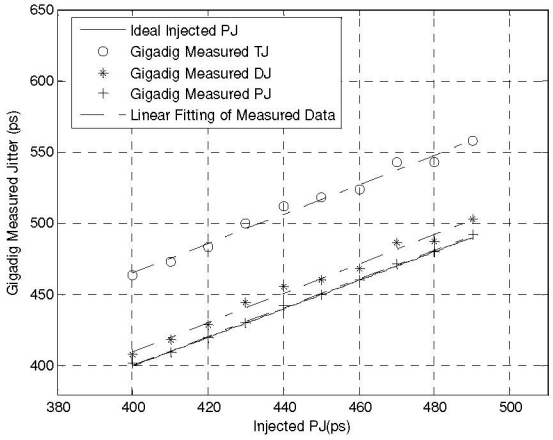


Fig.9 *Jitter injection calibration curves with digitizer*

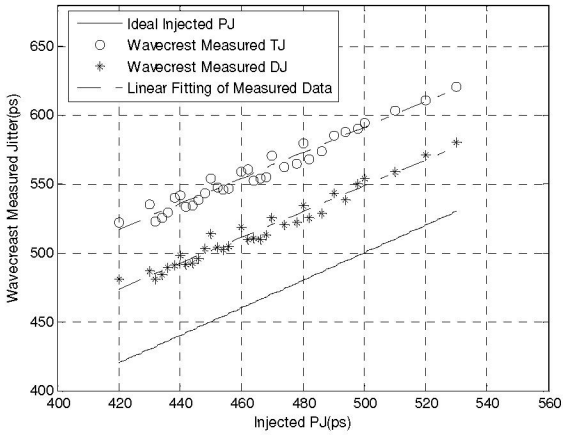


Fig.10 *Jitter injection calibration curves with Wavecrest SIA-3000*

3.2 Rx error rate monitoring

The receiver error rate can be monitored with several methods. One approach is to use its internal error checker, which is available as a DFT feature in our devices. However, the range of the counter is small and it might get saturated quickly when the error rate is high. Another approach is to loop back the received parallel data signals to the transmitter and then check the bit errors from the output of the transmitter. This approach usually needs a high speed BERT, which is not available in ATE. The undersamper on the Tx side is not a good candidate for bit error rate measurement. In addition, the transmitter itself might introduce errors, which makes it hard to justify the jitter tolerance test results. Our solution is to bring out received parallel data signals to device pins, and then compare the output of these pins with expected values in a digital pattern. The number of errors on each pin can be accessed by reading the failure counter for each parallel output pin. The failure counter is essentially a byproduct of the failure capture memory. It keeps track of how many pattern cycles the fail flag has been asserted from the last HSD reset or counter clear to the time the counter is read. Each failure counter is 16 bits. In our devices, the parallel data is 20 bits in width. The maximum number of errors the ATE can track is more than 1 million, which is enough to suppress the statistical variation and allow a generous step size on incrementing the jitter injection amount. This is important because we need to obtain multiple data points for extrapolation, but the performance (and hence the error rate) can vary from device to device at a certain jitter injection level.

One challenging issue for using the parallel data bus for error counting is synchronization. As we know for de-multiplexer testing, we have to achieve clock synchronization, bit phase alignment, byte (word boundary) alignment, and pattern alignment requirements. As the AWG and digital channels on ATE are in two clock domains, clock synchronization is a priority. As the two domains are generated from the same clock source on ATE, clock synchronization can be achieved by properly setting the two clock dividers such that the two frequencies are coherent and the receiver can work correctly. The receiver reference clock uses the same clock signal as the digital channels on ATE. The digital channel strobe is set centered on the parallel data of the receiver output. The byte (i.e. word boundary) alignment is becoming a norm for almost all devices. The frame alignment character is detected by the receiver to ensure that the right sequence of a word (from LSB to MSB) comes out of the parallel bus in a consistent way, from run to run, and from device to device. For pattern alignment, as shown in Fig.11, the widely used "SYNC" and "ALIGN" microcode on Teradyne ATE does not work accurate enough as the delay from the input of the receiver to the output of the receiver varies from device to device. We need to use a match loop to line up to the repeating PRBS pattern. The match loop continuously monitors the receiver output parallel sequence. Once an expected parallel data

sequence is detected, it jumps out of the loop and starts checking errors.

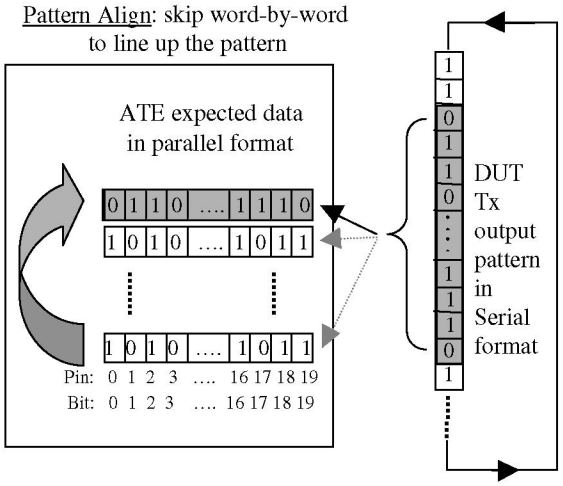


Fig.11 *The requirement for pattern alignment: skip byte-by-byte in parallel to line up the pattern generated by AWG (serial) and ATE (parallel).*

We can vary the length of the error checking pattern to get BER with expected confidence levels. A longer length pattern provides a higher confidence level, but takes more test time. By sourcing test signals with different amounts of jitter, we can get different bit error rates. For example, one needs to check at least 10^{10} bits to cover a stable BER of 10^{-9}, which takes more than a few seconds. The lower the error rate that we need to detect, the longer the test time it takes to complete. That is why no one can afford to directly test BER spec at 10^{-12} on a million dollar ATE, which will take nearly 2 hours at a data rate of 1.5Gb/S. Instead, we develop a much faster jitter tolerance test technique that can be done in seconds instead of hours, by measuring above the 10^{-9} BER and extrapolate down to the 10^{-12}.

3.3 Extrapolating to lower error rate

To illustrate how our new jitter tolerance BER scan technique works, we sweep injected PJ in small increments to get various BER levels. We measure the BER and calculate the BER distribution for different amounts of PJ injected. The theory is based on the traditional Q-factor equation:

$$Q = \sqrt{2} \times erfc^{-1}(2 \times BER)$$

Q factor can be calculated for different BER values, which directly links to the amount of jitter injected. At the point where a certain error rate shows up - illustrated as the bathtub curve crossing, we have:

$$DJ + 2Q \times RJ = UI$$

Under the black box assumption, we have no knowledge about the DCD and ISI jitter distribution. We assume that the intrinsic RJ is fixed when we sweep through the different amounts of PJ at a particular frequency. This is a reasonable assumption because the RJ in the AWG comes mostly from the sampling clock, which is fixed when we change the programmed samples for PJ injection. The ISI and DCD are assumed to be constant when the PJ is incremented. This is also a reasonable assumption, because they are mainly determined by the group delay caused by the bandwidth limited media. Of course, how the PJ combines with the ISI depends on the relative phase relationship, which is unknown inside the DUT. The internal DJ seen by the CDR (DJ_{int}) can be expressed as:

$$DJ_{\text{int}} = (PJ_{awg} + ISI_{awg} + DCD_{awg}) + DJ'_{cdr}$$

Where PJ_{awg}, ISI_{awg} and DCD_{awg} are the jitters in AWG waveform, DJ'_{cdr} is the additional DJ added by the CDR on top of the AWG jitter.

The internal RJ seen by the CDR (RJ_{int}) is defined as:

$$RJ_{\text{int}} = RJ_{awg} + RJ'_{cdr}$$

Where RJ_{awg} is the intrinsic RJ from the AWG; RJ'_{cdr} is the additional RJ added by the CDR on top of the AWG jitter, resulting from the CDR response to a certain jitter frequency. With the same device and the same injected jitter frequency, RJ_{int} should be a constant.

Under the Q-factor model at the bathtub crossing point (filling up $1UI$), we have

$$\begin{aligned} UI &= DJ_{\text{int}} + 2Q \times RJ_{\text{int}} \\ &= PJ_{awg} + (ISI_{awg} + DCD_{awg} + DJ'_{cdr}) \\ &+ 2Q \times (RJ_{awg} + RJ_{cdr}) \end{aligned}$$

Assign

$$ISI_{awg} + DCD_{awg} + DJ'_{cdr} = \Delta DJ$$

Then

$$1UI = PJ_{awg} + 2Q \times RJ_{\text{int}} + \Delta DJ$$

$$Q = (-\frac{1}{2RJ_{\text{int}}}) \times PJ + (\frac{UI - \Delta DJ}{2RJ_{\text{int}}})$$

Given Q factor as a linear function of injected PJ, based on the Q values we collected over a certain number of points, we are able to do a simple linear regression to calculate the Q as a function of PJ_{awg}

$$Q = C \times PJ_{awg} + S$$

where C and S are the linear fitting factors that can be easily acquired from data analysis.

$$C = -\frac{1}{2RJ_{\text{int}}} \xrightarrow{derived} RJ_{\text{int}} = -\frac{1}{2C}$$

$$S = \frac{1UI - \Delta DJ}{2RJ} \xrightarrow{derived} \Delta DJ = 1UI - 2S \times RJ_{\text{int}}$$

$$DJ = PJ_{awg} + \Delta DJ = PJ_{awg} + 1UI - 2S \times RJ_{\text{int}}$$

For BER=10^{-12}, substitute Q with

$$Q = \sqrt{2} \times erfc^{-1}(2 \times 10^{-12})$$

We can predict the injected PJ at this error rate

$$PJ_{awg} = \frac{\sqrt{2} \times erfc^{-1}(2 \times 10^{-12}) - S}{C}$$

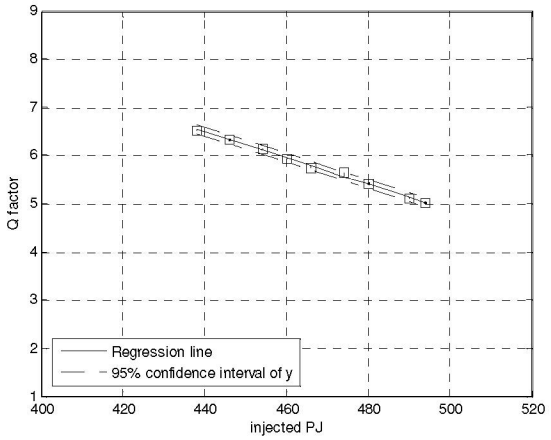
Therefore, we linked BER to a single variable – PJ_{awg} following the classical complimentary error function – *erfc*. The curve fitting for this function has matured for decades when applied to transmitter bath tub curve fit.

In summary, we found that even though we can NOT use bathtub curves to represent jitter tolerance extrapolation, the mathematics needed for conducting the jitter tolerance curve fit is still the complimentary error function.

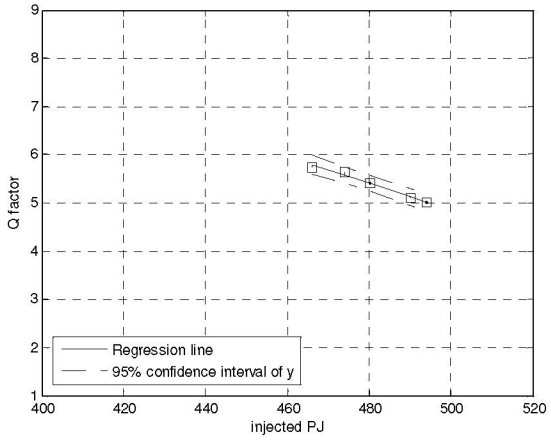
Next we will demonstrate how accurate our method is to predict the lower BER jitter tolerance. In the example below, jitter tolerance BER scan was performed and BER data was collected in the range of 10^{-6} to 10^{-11}; the data between 10^{-6} to $5x10^{-9}$ was considered as high BER and was used to predict the remainder of the points, which are considered as low error rate points.

Fig.12 is a linear regression fitting of the Q factor versus injected PJ. Figure (a) is the fitting result based on all measured points collected between 10^{-6} and 10^{-11}, while Figure (b) is the fitting result based on high BER points only. The difference between the two fitting results is shown in Fig.13.

Based on the Q factor fitting result, we can now plot the BER curve as a function of the injected PJ, and thus predict the jitter tolerance for lower error rate, e.g. 10^{-12}. Fig.14 shows the difference between the BER curve predicted based on the high BER points and the curve fitted with all measured points. The discrepancy is found to be very small; from Fig. 14, we read only 2~3ps difference at 10^{-12}. In this plot, diamond points (high BER) are used for the prediction; star points are the real measured points at lower BER. The real data in the lower BER range (star points) is very close to the predicted curve, which indicates that our prediction can successfully match the real measurement.



a) Q-factor fitting with all measured points



b) Q-factor fitting with high BER points

Fig.12 *Linear regression of Q factor as a function of the injected PJ*

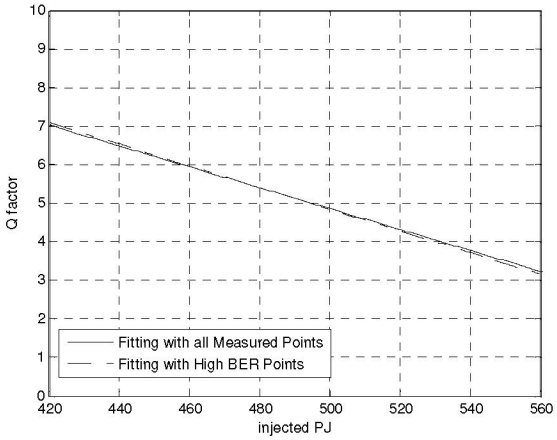


Fig.13 *A comparison between fitting results based on all measured points and high BER points only, which showed great level of accuracy of the prediction*

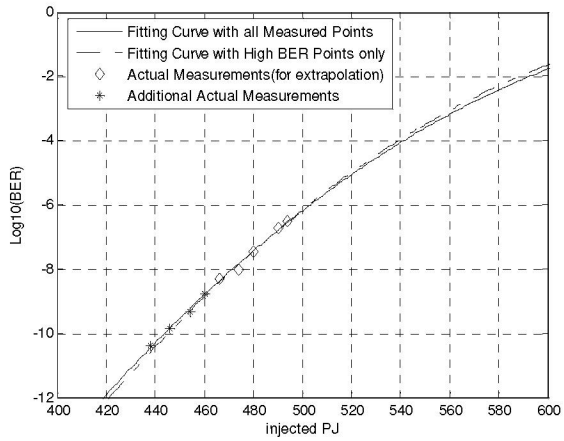
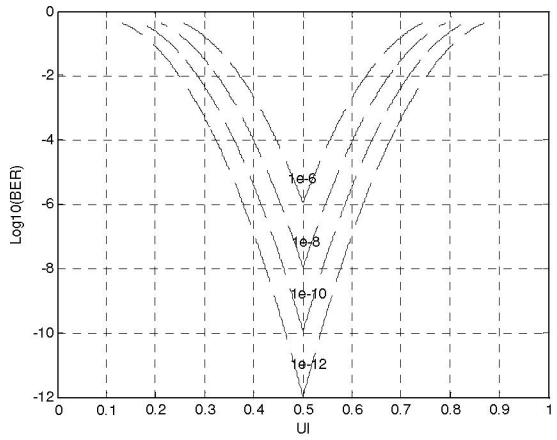
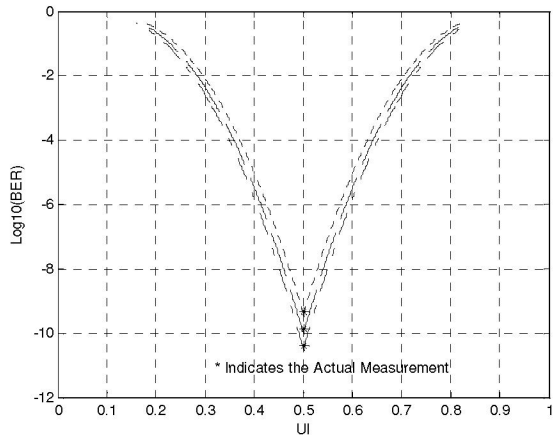


Fig.14 *A comparison of BER curve fitting results based on all measured points and high BER points*



(a) Predicted bathtub curves



(b) Predicted bathtub curves vs. actual measurements

Fig.15 *Bathtub curve prediction*

The jitter tolerance result presented in Fig.14 is in terms of PJ only. To include the ISI+DCD and RJ, we need to link this diagram with jitter injection calibration curves from the Wavecrest SIA-3000 (in Fig.10); the delta between the injected PJ and the actual TJ observed by the SIA-3000 is a constant around 92ps. For this particular device under test, the jitter (TJ) tolerance at 10^{-12} BER is 512ps or 0.76UI, while 420ps of PJ is injected through AWG.

Fig. 15 can further help understand the story from the bathtub point of view. Figure (a) conceptually shows that the depth of the bathtub curve moves with different amounts of jitter injected; Figure (b) shows the confidence level that we achieved by comparing the predicted bathtub to real measurements. This is another way to visualize the accuracy of our model.

4. EXTENDING THE TEST FOR CDR DYNAMICS ANALYSIS

In this work, we conducted intensive experiments on injecting PJ with a frequency of 93.75MHz to investigate the receiver jitter tolerance. We also extended our experiment to different frequencies to study the frequency response of the CDR jitter tolerance. Fig.16 is an example of the frequency response of a CDR.

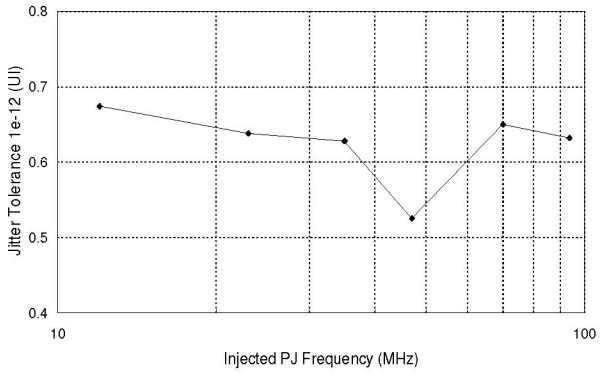


Fig.16. *Jitter tolerance frequency response*

As we can see from Fig.16, the frequency response is not flat. It has higher jitter tolerance at low frequencies. When the frequency increases, the jitter tolerance decreases and reaches a minimum around 40MHz. This type of frequency sensitivity investigation is extremely time-consuming with traditional jitter tolerance test methods. It takes days to characterize a device using traditional techniques. Our proposed accelerated jitter tolerance test scheme can significantly reduce the time needed for this type of characterization.

In addition, our proposed scheme provides a method to make a CDR model. From the testing point of view, CDR is just a black box. With our scheme, we can easily stress the CDR using test signals with different frequency and jitter profiles and check its performance. Based on this kind of experiments, it is possible to make an accurate CDR model, through which the performance of the CDR can be predicted. Future work will be done in this direction.

5. CONCLUSIONS

We have demonstrated an innovative method to make the time-consuming jitter tolerance test run faster by 1000 times, reducing the test time from >1 hour to a few seconds. The approach is straightforward, varying the amount of jitter to get receiver into several higher BER regions, and extrapolating down to the 10^{-12} BER spec for jitter tolerance. We presented assumptions we made, extrapolation models we derived, and experimental data that supports it. The model predicted the jitter tolerance performance down to 10^{-12} BER with data at 10^{-9} BER and above. The actual data down to the 10^{-11} validated the excellent accuracy of the technique in pico-seconds.

6. ACKNOWLEDGEMENTS

The authors would like to acknowledge Joe Martone and Bernhard Laschinsky for technical assistance.

REFERENCES:

[1] Y. Cai, A. Bhattacharyya, J. Martone, A. Verma, W. Burchanowski, "A Comprehensive Production Test Solution for 1.5GB/S and 3GB/S Serial-ATA", International Test Conference, Dallas, Texas, (2005).

[2] National Committee for Information Technology Standardization (NCITS) T11.2/Project 1316-DT/Rev 3.1: "Fiber Channel – Methodologies for Jitter and Signal Quality Specification", October 2001

[3] IEEE Draft P802.3ae/D3.3, "Supplement to Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method & Physical Layer Specifications", XGMII Extended Sublayer (XGXS) and 10 Gigabit Attachment Unit Interface (XAUI), October 2001

[4] T. Palkert, "SFI-5 Proposed Electrical Specifications", Optical Internetworking Forum (OIF2001.033), January 2001

[5] "High Frequency Serial Communication: Technology Requirement", Test and Test Equipment Section, ITRS: International Technology Roadmap for Semiconductors, Nov, 2004

[6] Y. Cai, T. P. Warwick, S. G. Rane, E. Masserrat, "Digital Serial Communication Device Testing and Its Implications on Automatic Test Equipment Architecture", IEEE International Test Conference, Atlantic City, New Jersey, (2000).

[7] Y. Cai, B. Laquai, K. Luehman, "Jitter Testing For Gigabit Serial Communication Transceivers", IEEE Design and Test of Computers, pp66-74 January (2002).

[8] B. Laquai, Y. Cai "Testing Multilane Gigabit SerDes Interfaces with Jitter Injection", IEEE International Test Conference, Baltimore, Maryland (2001)

[9] Y. Cai, S. Werner, G. Zhang, M. Olsen, R. Brink, "Jitter Testing for Multi-gigabit Backplane SerDes – Techniques to Decompose and Combine Various Types of Jitter", IEEE International Test Conference, p700-709, Baltimore, Maryland (2002)

[10] M. Li, J. B. Wilstrup, "On the Accuracy of Jitter Separation From Bit Error Rate Function", IEEE International Test Conference, p710-716, Baltimore, Maryland (2002)