Analysis and Estimation of Jitter Sub-Components

Student Name: Vijender Kumar Sharma

M. Tech-ECE-VLSI Design & Embedded System-12-13
June 12, 2014

Indraprastha Institute of Information Technology, New Delhi

Advisor
Dr. Sujay Deb
Mr. Rajkumar Nagpal

Submitted in Partial fulfillment of the requirements
for the degree of M. Tech in Electronics and Communication Engineering,
with Specialization in VLSI Design and Embedded System

Declaration

This is to certify that the thesis titled "Analysis and Estimation of Jitter Sub-Components" submitted by Vijender Kumar Sharma for the partial fulfillment of the requirements for the degree of Master of Technology in Electronics and Communication Engineering is carried out by him under our guidance and supervision at Indraprastha Institute of Information Technology, Delhi and ST Microelectronics, Greater Noida. This work has not been submitted anywhere else for the reward of any other degree.

Vijender Kumar Sharma **Place & Date** : …………………

Certificate

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

Dr. Sujay Deb **Place & Date**: ....
IIIT-Delhi

Mr. Rajkumar Nagpal **Place & Date**: ....
STMicroelectronics, India

Abstract

Maintaining quality of signal transmission is a major challenge with the increasing speed of data transmission in Nano-scale VLSI technology. The decreasing voltage margins at the same time, makes it even stringent to maintain Power Integrity (PI) and Signal Integrity (SI) in high speed systems. Jitter is an important phenomenon of signal integrity. It is the difference of expected transition edges to their actual transition edges. As the data rate of serial data pattern gets higher, the bit interval time gets shorter accordingly and thus the set-up time and hold-time requirements have very less margins for designers. If the jitter exceeds beyond its specified budget, the set up time and hold time can be violated in the system. To mitigate or reduce these effects, the causes of jitter in the circuit need to be identified. That is why the decomposition of jitter into its component is useful. With the help of jitter components segregation the time required to test a link for its bit error rate can be reduced.

Jitter can be classified into two major components called Deterministic Jitter (DJ) and Random Jitter (RJ). Deterministic jitter is bounded in nature while random jitter is unbounded. DJ has specific causes and is predictable while RJ is non-predictable and uncorrelated to data pattern. DJ is caused by crosstalk, Electromagnetic Interference (EMI), data pattern, etc. On the other hand, thermal noise, phase noise, process variations are the root causes of RJ. DJ can further be categorized into Periodic Jitter (PJ) and Data Dependent Jitter (DDJ). PJ is periodic, and bounded in nature. It is caused by PLL comparator, crosstalk, external noise coupling, etc., while DDJ depends on the both data being currently transmitted and the data that has been already transmitted. Inter-Symbol Interference (ISI) and Duty Cycle Distortion (DCD) are sub-components of DDJ. Reflection, discontinuities, limited bandwidth of the channel, threshold variations are the major root cause of DDJ.

In this thesis, comparisons of different types of jitter estimation techniques, their strength and limitations are discussed. Further, mathematical models of different jitter sources are implemented. Finally, we introduced a new technique for ISI estimation from total jitter using clock pattern. The results of these algorithms are extensively validated with Agilent ADS.

ACKNOWLEDGEMENTS

I take this opportunity to express my gratitude to all those who have helped me during my entire thesis period. First of all I would like to convey my heartiest gratitude and gratefulness to my guide Dr. Sujay Deb, IIITD and Mr. Rajkumar Nagpal, STMicroelectronics for providing their excellent guidance, monitoring and constant inspiration throughout the thesis without which this work would not have been a success. I also wish to express my sincere gratitude to Mr. Vivek Sharma, Vice President and Mr. Rakesh Malik, STMicroelectronics, India for allowing me to do my internship at STMicroelectronics. I am also thankful to our director Dr. Pankaj Jalote. I thank STMicroelectronics and IIIT, Delhi for providing me the opportunity to do this project.

Last but not the least, I thank the almighty, my parents, my brother, my friends and a very special person in my life, Nitika for cheering me, trusting on me and being with me throughout all the ups and downs.

List of Figures

Chapter 1 Introduction

Chapter 2 Literature Survey

Chapter 3 Jitter Generator and Segregation Techniques

List of Tables

Chapter 1 INTRODUCTION

1.1 Motivation

In the past two decades, there is a dramatic increase in the amount of information generated and in the end-user who access that information. The information available on the internet was 2000 Zeta-bytes in 2011, which has increased to 7000 Zeta-bytes in 2013, creating a tsunami of information [1] The number of internet users has increased ten times from 2000 to 2013 [2]. The demand of high-speed serial link has increased to transfer these data at higher data rates. The exponential increment in data rates is the main factor of signal distortion and higher BER at the receiving end, because of the bandwidth of the channel. In Fig.1.1, a basic demonstration of communication link as shown in [3]. The Phase Locked Loop (PLL), channel, Clock Data Recovery (CDR), etc. are responsible for deviation in the signal from their ideal position, this deviation is known as jitter at high data transmission rate.

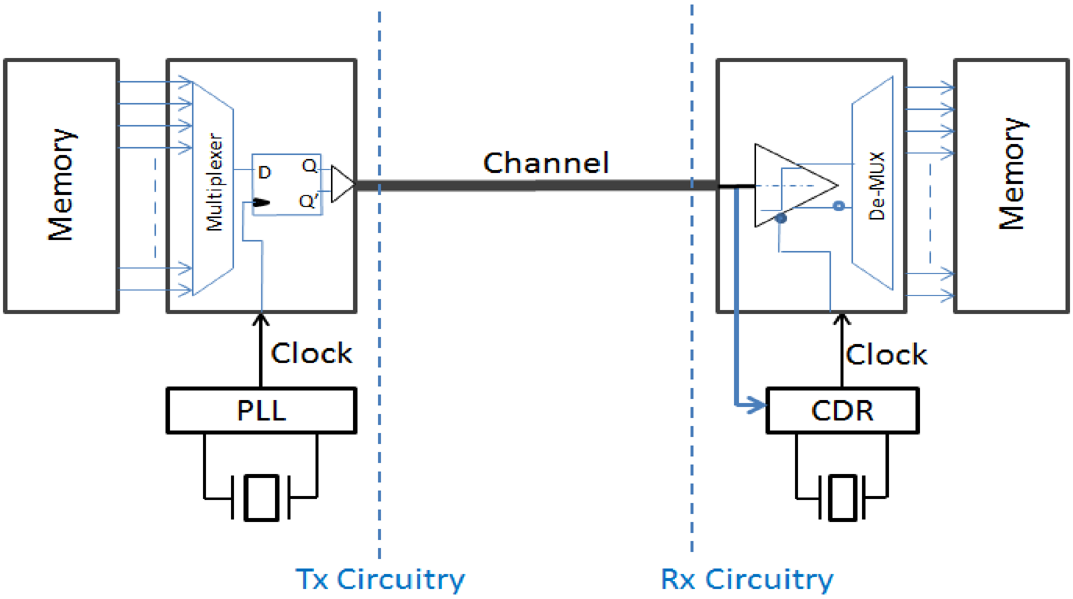


Fig.1.1 A basic demonstration of serial communication link.

In another aspect, the frequency of clock increases with the advancement in technology and higher data rate. Today's the microprocessor run at about 100 times faster than the 15 years ago as shown in Fig.1.2 [4]. Due to higher data rate and clock speed, the bit interval time of the signal continuous to decrease. This reduces the time budget that is available for jitter. This has resulted a lots of design efforts to be directed towards reducing jitter. To reduce jitter, the causes of jitter in a circuit need to be identified. Once the type and amount of jitter present in the circuit are identified, the cause of these components can be looked up and tried to minimize or mitigate.

In other words, the precise segregation of jitter into its subcomponent is significant due to following reasons [5] :

- For acceptance to technology standards, and

- Provides detailed information for refining design.

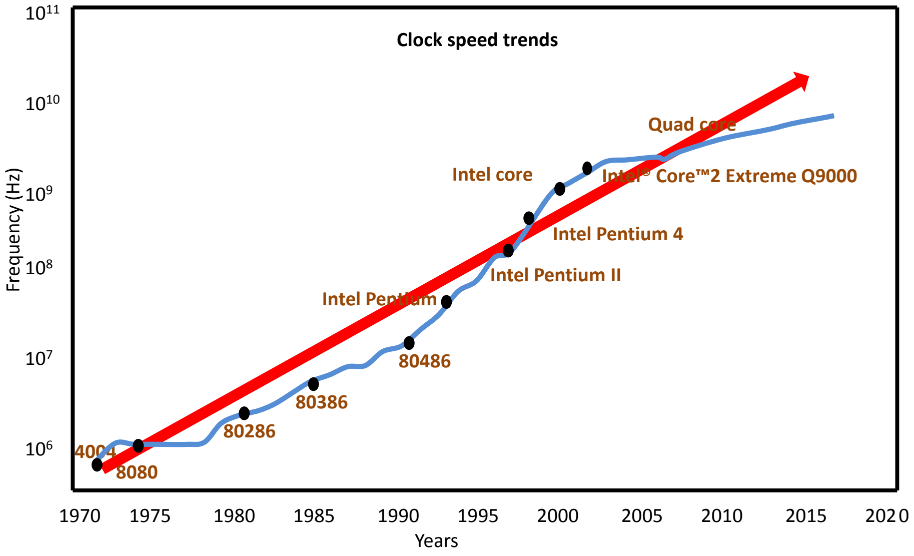


Fig.1.2 Processor clock speed trends [4].

Due to increase in operational speed of data, to maintain Signal Integrity (SI) in the high speed serial links is the most critical issues in today's semiconductor industries. Ringing, overshoot, undershoot, wander, noise, jitter are the key factor of SI.

1.2 What is Signal Integrity?

Any deviation in actual waveform from ideal waveform is called signal integrity [6] [7] [8] [9] [10]. In other words, the term of SI defines as the deviation from intact digital signals. The SI addresses the problem in the signal at receiver end. Whether the signal reaches its destination, when it is supposed to? And what is the quality of the signal?

The Fig.1.3 gives the clear idea of SI. In the given figure, the black bold response (zero to one transition) is the ideal waveform while the blue dashed response is the waveform at the receivers end. The signal deviates from the amplitude is called amplitude noise (or simply noise) while the signal deviates from transition edge is called timing jitter (or simply jitter) [7].

In this thesis, the main focus is on jitter. Jitter is the most important concern in today's high speed serial links. There are two ways to solve this problem-

- Either to control this jitter, or
- To separate the jitter

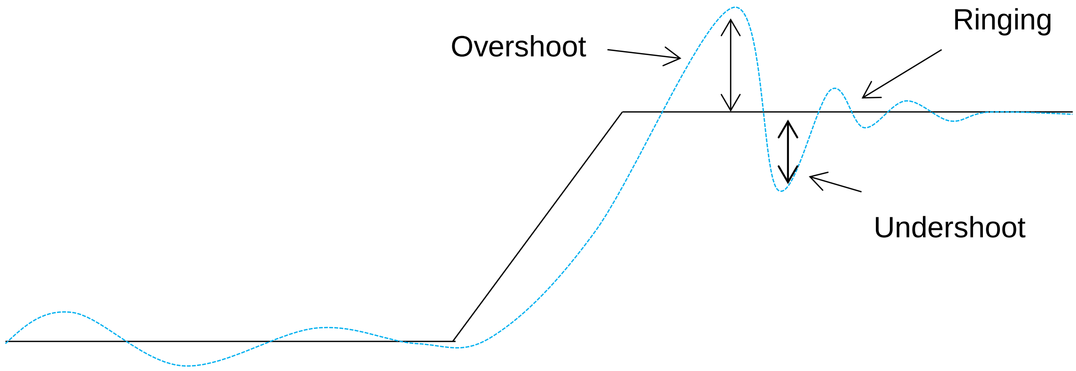


Fig.1.3 Example of an ideal signal versus distorted signal.

1.3 Basics of Jitter

The term jitter defines as the deviation of the signal from its ideal transition edge. Sometimes these transition edges come early and sometimes they come late [11]. Due to this behavior of the signal, the difference of the time from actual and expected transition edge is called jitter. Fig.1.4 shows an ideal signal versus a jittered signal for a digital waveform.

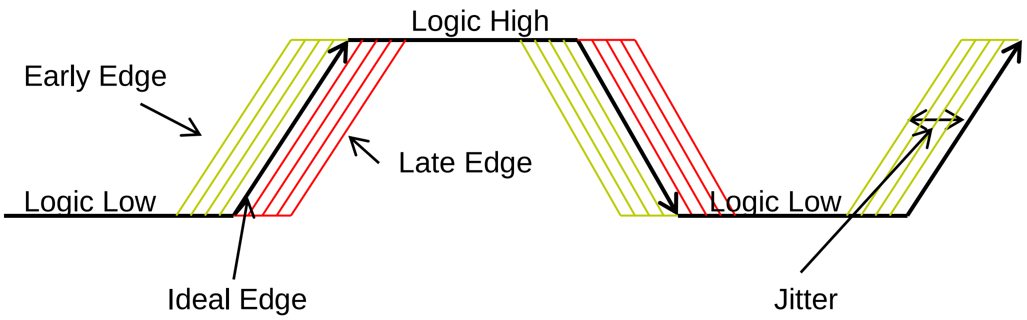


Fig.1.4 Shows an ideal transition edge versus jittered edges.

1.4 Effects of Jitter

Effects of jitter on the signals includes the setup time and hold time violation in the digital circuits, distorted signal at the receiver ends, opening of eye becomes shorter, etc. Due to these effects, BER at the receiver ends is higher.

Effect of jitter on the signal was not big problem in the past, when designers were dealing with slow link speeds. The bit interval was quite large at relatively slow speeds and bit interval could accommodate relatively high amounts of jitters.

As the increase in the link speeds, the width of bit becomes shorter and shorter and jitter has now started to occupy a major portion of the bit interval and its effect on the signal is more hazardous.

1.5 Ways to measuring Jitter

Jitter measurement can be classified into five categories [12] [13] [14].

- Periodic Jitter

- Cycle to Cycle Jitter
- Phase Jitter
- Long-Term Jitter
- Time Interval Error (TIE)

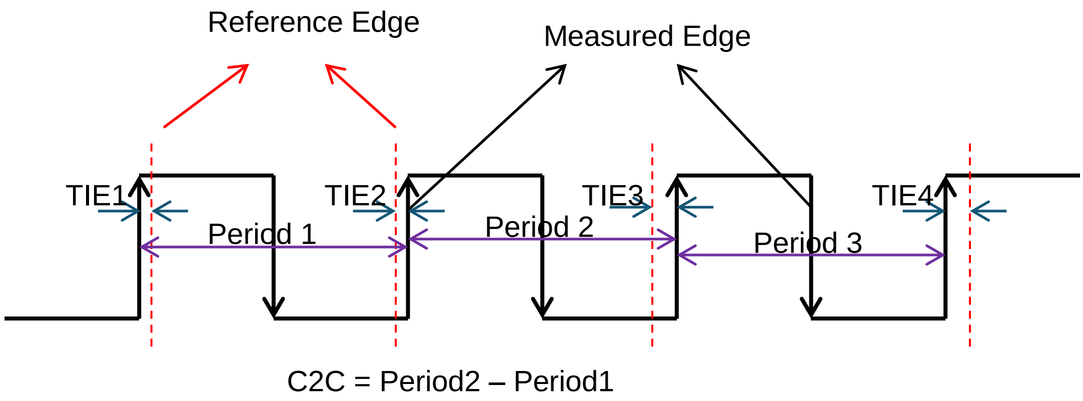


Fig.1.5 Shows different ways of jitter measurement.

1.5.1 Period Jitter

Period Jitter (PJ) is the difference of ideal clock cycle versus measured clock cycle. For a given number of individual clock cycles, we can measure each cycles of the clock and calculate the peak to peak magnitude, Root Mean Square (RMS) and the average clock period. Why average clock period? This is because, in real time signal, it is difficult to identify the ideal clock edge due to oscilloscope's output error. Storage oscilloscope is useful for measuring for period jitter. Timing margins of a system are calculated with the help of period jitter measurements.

1.5.2 Cycle to Cycle Jitter

Cycle to Cycle (C2C) is the difference of two consecutive adjacent cycles of output transition. The value of such measured difference of adjacent cycles is the maximum C2C. Peak value of C2C is defines the maximum deviation in rising edge of two adjacent cycles. Further, C2C is used to illustrate the stability of spread spectrum pattern. Timing Interval Analyzer (TIA) and Time Interval software are available for measuring the C2C jitter. A large C2C can cause a system failure.

1.5.3 Phase Jitter

Phase jitter is the integration of phase noise. Phase noise is the instabilities of a frequency in the frequency domain. Fig.1.6 shows phase noise versus frequency plot. The horizontal axis denotes the offset frequency and vertical axis denotes the phase noise. The integration of the shaded area (phase noise) is phase jitter.

1.5.4 Long-Term Jitter

Long term jitter measures the deviation in a clock's output from its ideal position over a long sequence of clock cycles. The number of clock cycles depends on the application and the clock frequency. Differential phase measurement technique is useful to measure long term jitter.

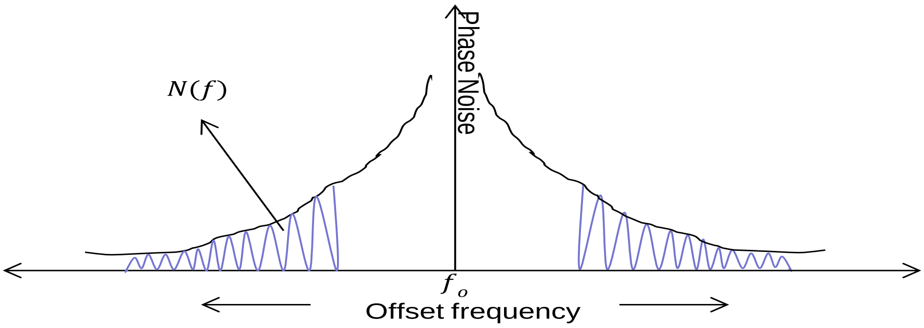


Fig.1.6 Frequency versus phase noise plot [13].

1.5.5 Time Interval Error

TIE is difference of reference transition edge and deviated transition edge at mid-crossover points. It is discrete time domain representation of phase noise. In Fig.1.7, the basic concept of TIE is shown.

In the below figure, a reference clock is represented by black waveform and output or deviated clock is represented by red waveform. A threshold voltage level or zero voltage level is defined by horizontal dashed line which helps in calculation of mid-crossing points of a signal. The intersection points of vertical dashed line (black and red line) with threshold voltage line are called mid-crossover points. The difference of reference mid-crossover points and output clock mid-crossover points gives TIE. TIE value is in terms of Pico-second and it should be negative or positive depending on arrival of transition edges. TIE plot gives the leading and lagging time instants of output clock from reference clock.

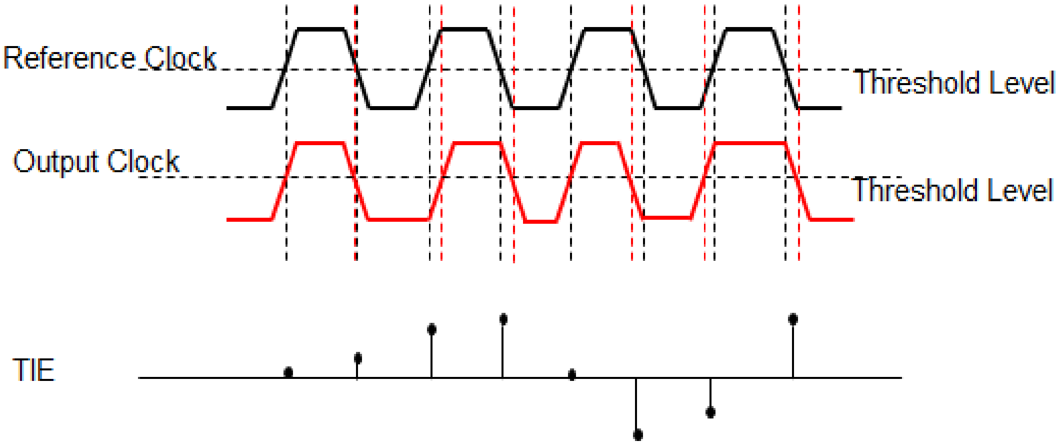


Fig.1.7 Calculation of TIE

1.6 Types of Jitter

Total jitter is composed of a complex mixture of various types of jitter, shown in (1). It can be classified into deterministic jitter (DJ) and random jitter (RJ). Periodic jitter (PJ) and data dependent jitter (DDJ) are the subcomponent of DJ. DDJ further classified into inter-symbol interference (ISI) and duty cycle distortion (DCD) [7]. Fig.1.8 shows classification of total jitter.

$$TJ = RJ * PJ * DCD * ISI \qquad (1)$$

Here “$*$” denotes the convolution operator.

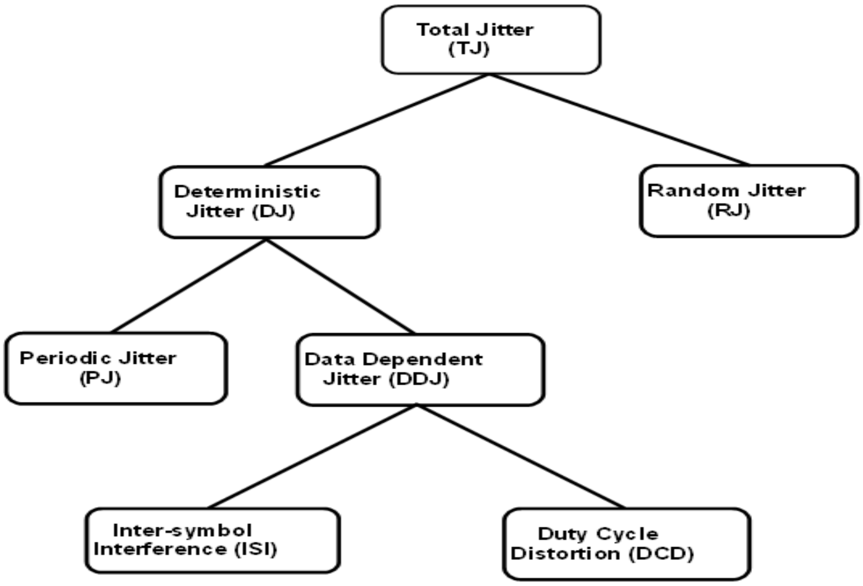

Total Jitter (TJ)
Deterministic Jitter (DJ)
Random Jitter (RJ)
Periodic Jitter (PJ)
Data Dependent Jitter (DDJ)
Inter-symbol Interference (ISI)
Duty Cycle Distortion (DCD)

Fig.1.8 Classification of total jitter

1.6.1 Random Jitter (RJ)

Random jitter is also known as unbounded jitter or Gaussian jitter. It is non-predictable, uncorrelated, aperiodic and independent of the source of jitter. According to central limit theorem, the pdf of RJ should follow normal Gaussian distribution [15], shown in Fig.1.9. It is characterize in root mean square terms. Standard deviation (σ) is used to predict the occurrence of outlying measurements from the mean [16]. Peak to peak value of RJ given is only defined at specific bit error rate (BER) as shown in (2) due to its unbounded nature. Definition of BER is defined in section 1.7.3

$$RJ_{pk-pk} = \alpha * RJ_{rms} \qquad (2)$$

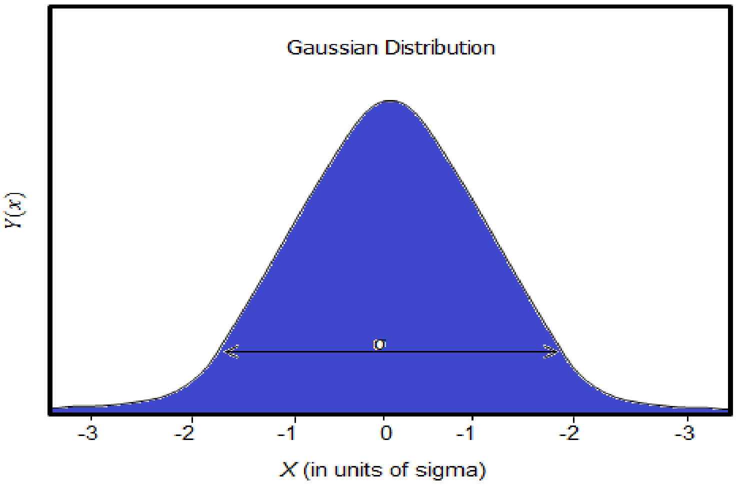


Fig.1.9 Random jitter histogram [15].

Thermal noise, shot noise, flicker noise, process variations, and microscopic variations in the resistance and the impedance of the circuit traces which can be caused by the inevitable small variations of traces width and many other minor contributors such as cosmic radiation are the primary cause of random jitter.

1.6.2 Deterministic Jitter (DJ)

Deterministic jitter is bounded in nature and has non-Gaussian probability density function. It is predictable and has specific cause. It is correlated as well as uncorrelated to data pattern. That is, the DJ of a given logic transition may or may not affect the jitter of another transition. Peak to peak value of DJ is not dependent on number of data points as shown in Fig.1.10. Hence, sufficient numbers of data points are taken to complete at least one cycle of periodic element.

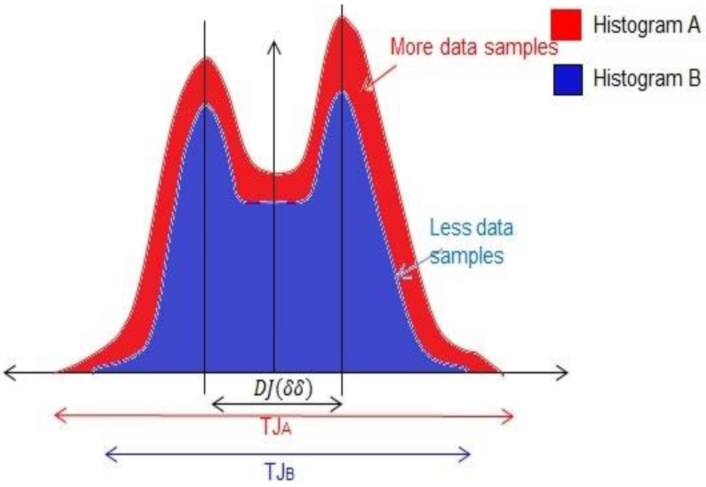


Fig.1.10 Histogram of total jitter.

In Fig.1.10, a histogram of bimodal distribution is shown. It is clearly visible from figure that DJ does not increase with the sample points because DJ is measured between the peaks [17]. The total jitter increases with increase in data sample points. It is due to the presence of extra RJ.

DJ is typically caused by circuit design, crosstalk, data dependencies, electro-magnetic induction or external environment. To reduce total jitter, it is necessary to reduce DJ.

1.6.3 Periodic Jitter (PJ)

Period is the subcomponent of DJ. It is also known as sinusoidal jitter. It can cause periodic variation in the phase of data pattern [18] [16]. It includes any jitter at fixed frequency or period. It is bounded, uncorrelated to pattern and periodic in nature. Like DJ, it is also have specific root cause but it is uncorrelated to data pattern. It is characterize by its peak to peak value and its does not depends on non-ideal behavior of channel. Probability distribution of periodic jitter is shown in Fig.1.11.

Periodic jitter exist even when no transition edges occurs. For example, consider that the data pattern is 01110, then between consecutive 1's periodic jitter can be found by interpolating the jitter values at 0 to 1 transition edges. In fact, the transition edges are only sampling points of the periodic jitter.

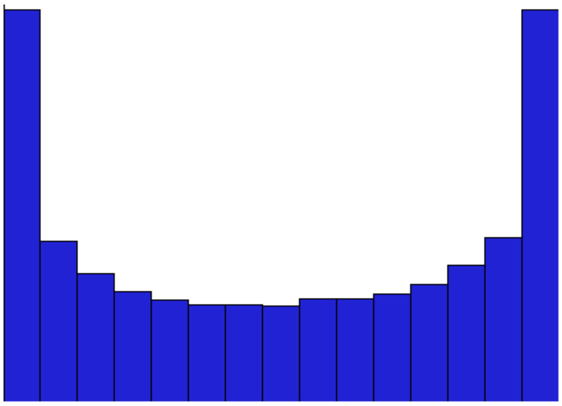


Fig.1.11 Histogram of periodic jitter [18].

Typical causes of period jitter are power supply fluctuations, PLL comparator frequency feed through, external noise coupling, crosstalk, etc. Further, PJ is easy to measure and useful in diagnosis jitter problems.

1.6.4 Data Dependent Jitter

It exists only at the occurrences of the transition edges. It is correlated to data pattern, hence it is also known as "correlated jitter". DDJ depends upon the data being transmitted. The dependency is both on the current data and the past data (already transmitted data). For example, as shown in Fig.1.12, when the jitter of a 0 to 1 transition or 1 to 0 transition that follow a pattern of alternating bits, e.g. 01010101, differs from 0 to 1 transition that follows a long string of 1's or 0's, e.g. 01000001, that jitter is called DDJ.

Typical cause of DDJ is impedance mismatches, the resistance and frequency response of the transmission path. DDJ is further broken into ISI and DCD.

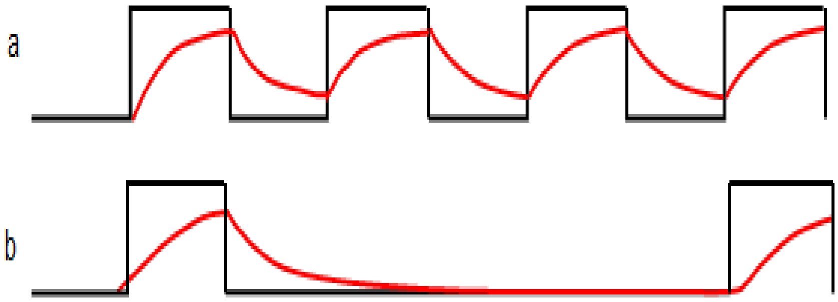


Fig.1.12 Dependency of jitter on different pattern.

1.6.5 Duty Cycle Distortion (DCD)

DCD refers a distortion or change in the duty cycle of a pattern. For example, the 0 in the sequence 1110111 is of a different width than the one in 0001000, shown in Fig.1.13, this change in the width is called DCD. Alternatively, DCD is the phenomenon in which every rising/falling edge is always delayed or advanced by the same amount. The amount of DCD at the receiver end can be calculated by (3).

$$DCD = d_1 - d_2 \tag{3}$$

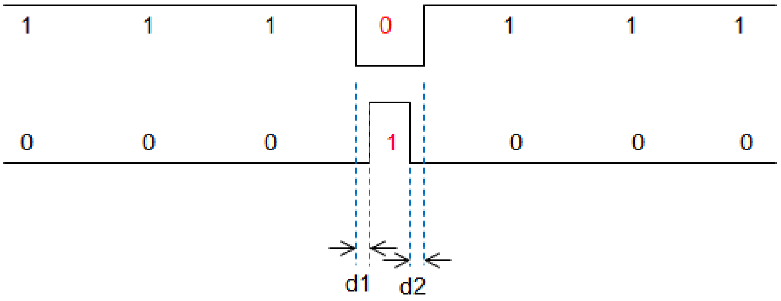


Fig.1.13 Example of Duty cycle distortion.

DCD is bounded and correlated to data pattern. Unlike PJ and RJ, it depends on non-ideal behaviour of channel. It follows a simple dual-dirac delta distribution, shown in Fig.1.14.

DCD occurs due to slight variation in refernce level during jitter analysis. In addition to reference level variation, another common cause for DCD are amplitude offset, turn on delays and saturation effect.

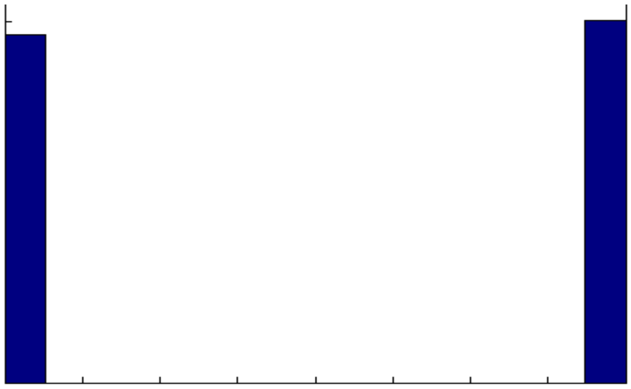


Fig.1.14 Histogram of duty cycle distortion [18].

1.6.6 Inter-symbol Interference (ISI)

The asymmetrical degradation in amplitude and in the rise time and fall time of a transmitted signal represents the effect of ISI in the signal, as shown in Fig.1.15. The effect of ISI is very less in the lower frequency transmitted signal. A clock signal does not have any ISI [7] . ISI is only related to the data pattern and if signal is repeating pattern then ISI is only periodic. It is bounded and correlated to data pattern and it does not have any defined mathematical model.

ISI is caused by bandwidth limitation and signal reflection. Channels have slow step response due to the resistive components present in the channel. This slows step response prevents a signal from switching quickly. Thus, if a long string of 1’s is followed by a single 0 which is turn followed by a string of 1’s again, the signal would have reached a full steady state high value before undergoing a transition to 0. In this case, the signal would take a longer time to reach the threshold value giving a positive timing error. Also, logic low would not achieve the minimum value, because even before the steady state negative level is reached the following high going transition makes it go back to high logic level. Thus a high going transition will undergo a negative jitter, i.e. the edge will come early than the expected [19]. This can be seen in Fig.1.16.

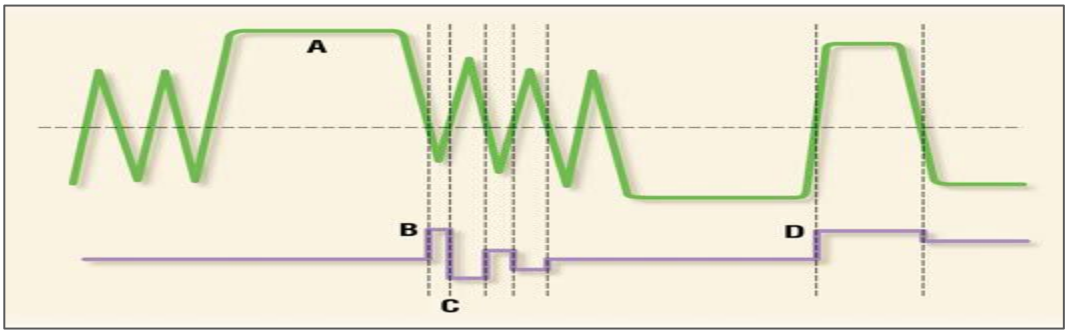


Fig.1.15 Example of ISI due to bandwidth limitations [19].

Further, ISI can be caused by improper impedance termination and due to the combination of the design of the traces and circuit geometry.

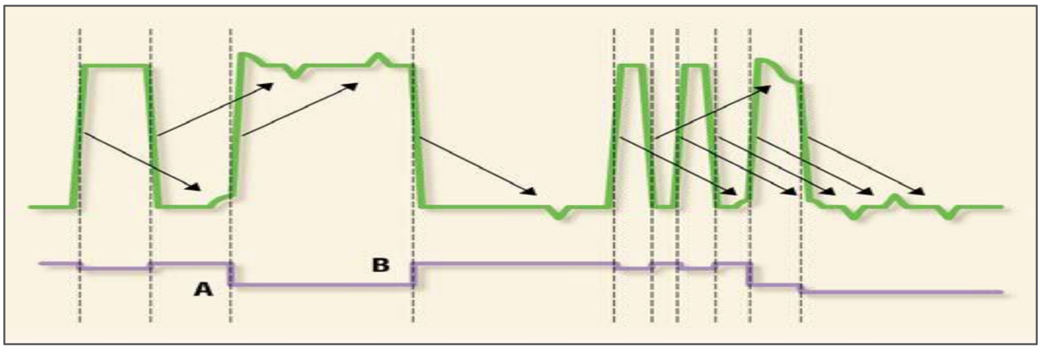


Fig.1.16 Example of ISI due to reflection [19].

1.7 Jitter analysis

Jitter analysis is important for determining the source of jitter. Some measurement techniques for jitter analysis, including the separation of RJ and DJ are discussed in this section. Dedicated jitter measurement instruments such as BERTs, oscilloscope can have capability to analyze jitter using the below mentioned techniques.

1.7.1 Eye Diagram

The eye diagram is a methodology to characterize the influence of both the timing jitter and amplitude noise on the data. The electrical quality of the signal can be visualized and determined through eye-diagram [20]. The eye diagram is constructed by superposition of transition edge (rising and falling edge) of several data pattern over one unit interval (UI) range. In Fig.1.17, the eye diagram analysis is shown.

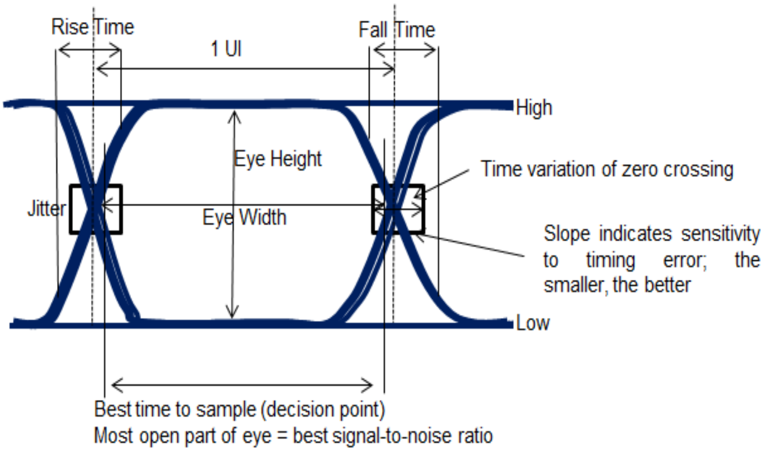

Rise Time
1 UI
Fall Time
High
Eye Height
Time variation of zero crossing
Jitter
Eye Width
Slope indicates sensitivity to timing error; the smaller, the better
Low
Best time to sample (decision point)
Most open part of eye = best signal-to-noise ratio

Fig.1.17 Eye Diagram Analysis.

An eye diagram can provide us several parameters that are useful for jitter analysis, such as rise and fall time of the signal, total jitter information, BER, etc. The larger the opening of eye, the better tolerance the system has and probability of BER is very less. For the correct functioning of a system, eye should not be closed.

1.7.2 Histogram

Histogram is another jitter measurement viewpoint. It is a bar graph that shows the probability of jitter at different time instance. Alternatively, Histogram represents the probability distribution function of a signal, shown in Fig.1.18. It is use to view the distribution of signal.

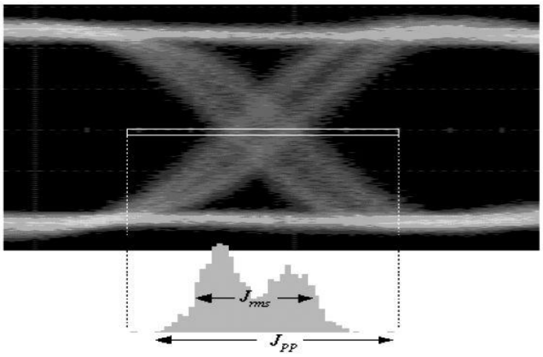


Fig.1.18 Histogram analysis [21].

The histogram may be bounded or unbounded, depends upon presence of RJ. The histogram shown in figure with an approximation of RMS and peak to peak jitter indicated. The peak to peak measurement spans from minimum and maximum point collected.

1.7.3 Bathtub Plot

It is plot between the BER on y-axis and sampling time though a bit interval on x-axis, shown Fig.1.19. BER depends on the sampling window, similar to the eye diagram. As BER increases the window becomes narrower. The distance between the curves at a particular BER, represents the available time for sampling without a bit error.

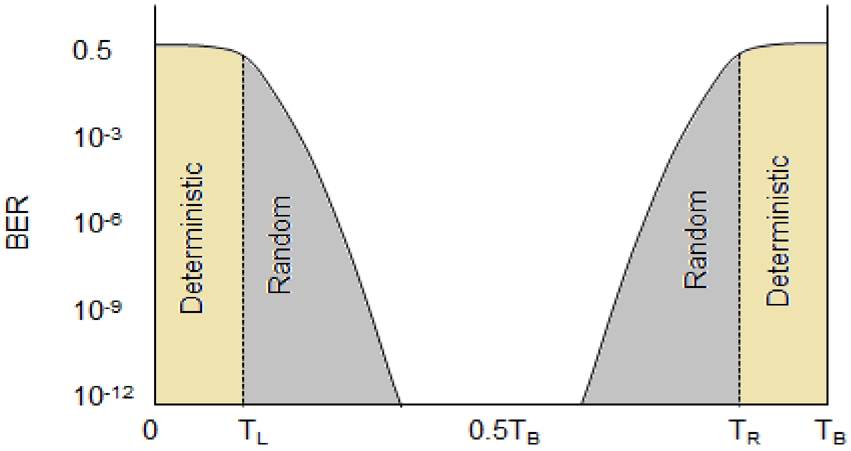


Fig.1.19 Bathtub Curve.

As shown in figure, near the left and right edges, jitter is dominated by deterministic jitter and the value of BER is constant and at 0.5. This indicates that the probability of an error occurring when data is sampled near the edges is 50%. As we move into the bit interval, the BER begins to decrease because jitter now is dominated by random jitter and due to Gaussian nature of random jitter the probability of jitter decreases as the value of jitter increases.

The curve can be obtained from jitter PDF curves by integrating this curve from x to infinity to obtain the BER_L. This integration indicated the probability that a transition might occur after the sampling time, meaning an old data was sampled.

To obtain BER_R the curve is integrated from minus infinity to x to indicate that transition happened before the data was sampled and so data is lost. The mathematical equation for total BER is shown below.

$$\mathrm{BER_L(x)} = \rho_\mathrm{T} \int_\mathrm{x}^{\infty} \mathrm{J(x')dx'} \tag{4}$$

$$BER_R(x) = \rho_T \int_{-\infty}^{x} J(x')dx' \tag{5}$$

$$BER(x) = BER_L(x) + BER_R(x) \tag{6}$$

This thesis explores the roots cause of different types of jitter, their mathematical module and tried to obtain the different values of jitter components as accurately as possible. Further, the modification can be done in existing algorithm. Finally we design a new algorithm for ISI estimation.

1.8 Thesis Organization

An introduction to jitter analysis in high-speed serial links is given in the subsequent Chapter 1. Fundamentals of signal integrity, jitter, and noise and BER analysis in serial interfaces are described, together with an overview to the ways of measuring jitter. Further, the overview of jitter analysis approach such as eye diagram analysis, histogram and bath-tub curve analysis are discussed.

Chapter 2 contains the literature review of jitter segregation techniques.

Chapter 3, the jitter generator and insertion methodologies, their mathematical models are discussed. Further, the jitter analysis in time domain and in frequency domain and improvement in jitter segregation techniques are discussed.

Finally, Chapter 4 summarizes the results of jitter segregation algorithms, their strength and limitation and concludes a brief outlook to future research direction.

Chapter 2 Literature Study

Separation of conventional DJ from TJ was discussed in the IEEE 802.3ba D1.1 editor's note. In this note, a 99% jitter estimation method is introduced. Total jitter at higher probability (approx 10^{-4}) that contains both DJ and RJ, 99% estimation seperates DJ and RJ from TJ but at lower probability, this method will not able to separate the DJ and RJ accuratly [22]. In IEEE 802.3 ba, the problems and measurement error during DJ and RJ estimation are discussed. The 99% jitter is defined as the time difference from 0.5% to 99.5% of the histogram, as shown in Fig.2.1 . This method is based on specific histogram that contains atleast 10,000 bins and the bins having amplitude greater than 1 are considered.

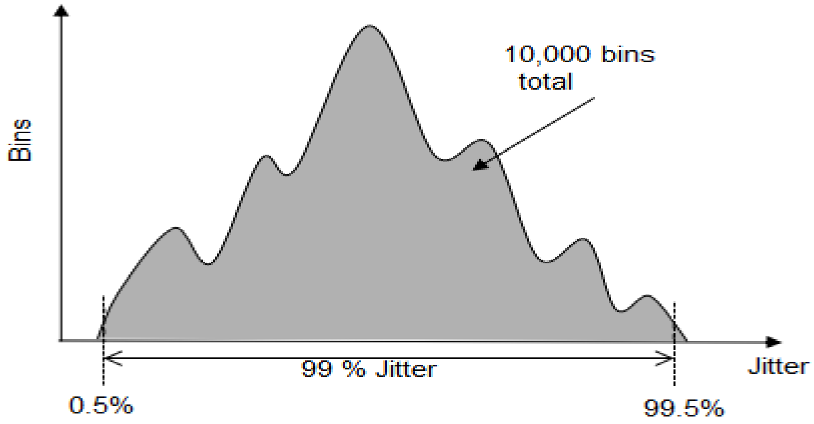


Fig.2.1 99% jitter estimation [22].

Further, this method is not generic and scalable for multiple measurement platforms. Another limitation of this method is if the number of bins in the histogram is beyond 10,000, then DJ will be underestimated and if RJ dominants over DJ, the method estimates extra value of DJ. The above discussed limitations are shown in Fig.2.2 .

In short, this method is wrong metric to quantify or bound DJ.

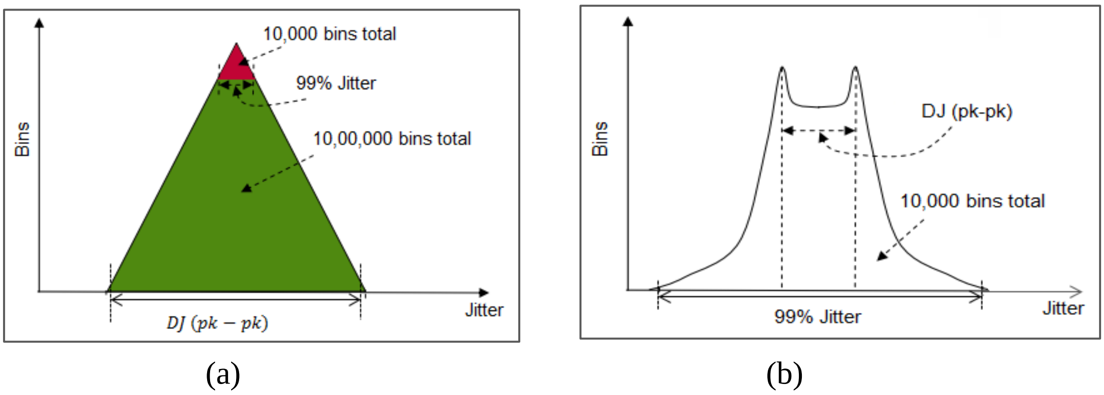


Fig.2.2 Examples for 99% Jitter [22] (a) An example of 99% jitter under-estimates DJ. (b) An example of 99% jitter over-estimates DJ.

Different ways to obtain RJ and DJ were discussed by Dr. Martin Miller in [23]. In addition to this, he characterizes three kinds of clients, who are interested in jitter analysis. First type of clients is system designers and integrators who are only concerned with estimating overall error-rates and they doesn’t care about the root cause of timing jitter. Second type of clients is component and subsystem designers, who are exactly concerned about which source of jitter are contributing to a subsystem jitter measurements. Finally, the remaining whose main concern is to obey some suggested specification or reference.

The four kinds of methodologies for jitter measurements and their limitation are also discussed in this paper. The time interval analyzer (TIA) estimates the jitter on the basis of “edge to edge” measurements. It can also perform “edge to reference” measurements if system clock reference is provided. Real time oscilloscope signal analysis can execute both kinds of analysis. Sampling oscilloscope is used to obtain timing distribution and bathtub curve can be determined directly via bit error rate tester (BERT).

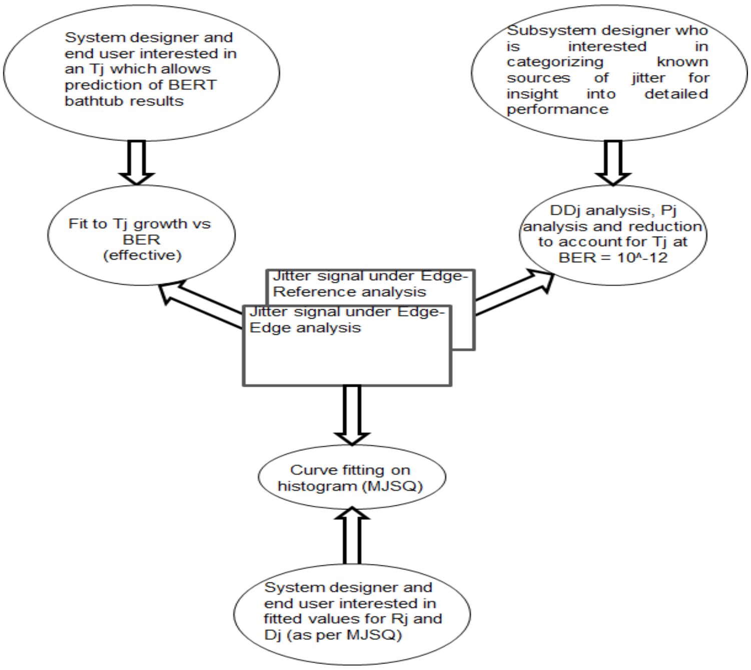

System designer and end user interested in an Tj which allows prediction of BERT bathtub results
Subsystem designer who is interested in categorizing known sources of jitter for insight into detailed performance
Fit to Tj growth vs BER (effective)
DDj analysis, Pj analysis and reduction to account for Tj at BER = 10^-12
Jitter signal under Edge-Reference analysis
Jitter signal under Edge-Edge analysis
Curve fitting on histogram (MJSQ)
System designer and end user interested in fitted values for Rj and Dj (as per MJSQ)

Fig.2.3 Six tales of RJ and DJ [23].

In his research work Buckwalter has introduced an analysis of DDJ calculation for first order system as well as an approximation for DDJ in second order system is described in terms of damping factor and natural frequency with the help of Taylor’s series expansion [24]. The predicted DDJ in this paper are verified with jitter measurements in a broadband amplifier and in a bandwidth-limited amplifier.

This paper concludes that response of causal system is not only dependent on current bit but it also depends on the previous bits. For example, the amount of DDJ for 001 and 101 bits are different while in case of 010 and 101, the value of DDJ is same. Next according to equation (8), the a_{-2} has main impact on the DDJ and the prior bits have lesser effect on the current threshold crossing time. This is because of the value of α is less than one and hence, α^k is exponentially decreasing.

$$t_c = \tau.\ln[\frac{-a_o + \sum_{n=-\infty}^{-1} a_n[\alpha^{-(n+1)} - \alpha^{-n}]}{v_{th} - a_o}] \tag{7}$$

Here t_c is the threshold crossing for arbitrary value of previous bits, a_n is the binary value, v_{th} is the threshold voltage and α represents the ratio between bandwidth and the bit rate of the system.

Further, DDJ can be calculated for higher order response by equation (9)

$$t_{c,DDJ} = -\frac{v_{th} - g(t_o + T)}{g^{(1)}(t_o + T)} \tag{8}$$

Here $g(t_o)$ is the channel output response and t_o is the threshold crossing time for the step response. The denominator part of equation (9) denotes the first order Taylor’s approximation of step response. In another research work, estimation of DDJ in high speed digital system using single pulse response is proposed and obtained DDJ is experimentally verified with timing jitter measurement [25]. Mostly, commercial simulation tools such as BERT, TIA, etc. are used to predict DDJ. But this takes ample amount of time for adequate timing resolution and does not have any other choice of the channel model except the ones which are provided by the tool. The bandwidth of the channel should be known in advance for the estimation of DDJ in Buckwalter’s work [24]. This problem can be overcome by physically characterized channel based on the transmission line theory and then use the single pulse analysis method.

The transfer function of the transmission line is

$$H(w) = e^{-l.\gamma(w)} \tag{9}$$

Where $\gamma(w)$ is propagation constant and it depends on R, L, G, C component of transmission line.

Maximum DDJ value can be predicted by using single pulse method because the response of a single pulse after consecutive 0’ or 1’ is considered as the worst pattern on lossy transmission line. The single pulse is composed of two step function which have opposite polarity and delay of a bit period. The resultant pulse is deviated from original one after passing through transmission line as shown in Fig.2.4

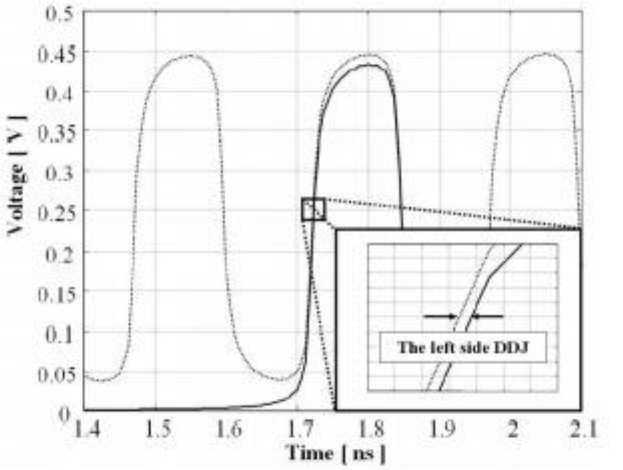


Fig.2.4 Comparison of a clock signal and the single pulse response [25].

Finally, DDJ is the sum of the difference of single pulse and clock signal during rising and falling edge at reference crossing time.

$$t_{DDJ} = t_{DDJ\_rising} + t_{DDJ\_falling} \tag{10}$$

The authors of [25] further elaborate the transmission line and DDJ estimation in [26]. In this work, the modeling of transmission line, modeling of single bit pulse, single pulse analysis methods were explained in more elegant way. Finally, the work of this paper was verified by the time domain jitter measurement in several conditions and found to be quite accurate.

DDJ model and estimation methods, in presence of capacitive and inductive discontinuities that results due to vias and right angular bends are discussed in [27]. SPICE tool is used for verification of the proposed model.

In [24], channel response has been modeled as first and second order filters. However, in presence of discontinuities, such a simple model becomes invalid. This is because of reflections from discontinuities and for the bandwidth limited channel that keeps the value of previous bits. This increases the chances of getting more error while estimating DDJ. To avoid this problem, use a transmission line having capacitive and inductive discontinuities. The model of transmission line with an impedance discontinuity is shown in Fig.2.5.

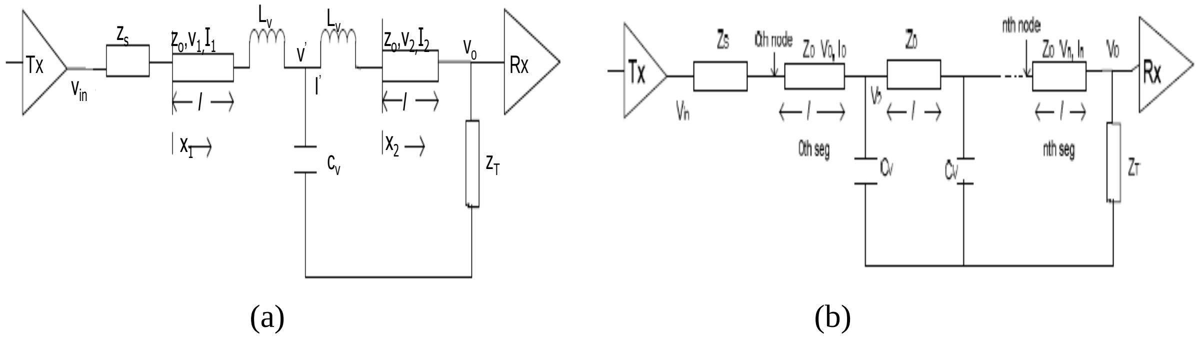


Fig.2.5 Transmission line model (a) transmission line with T mode for vias (b) T line with equally spaced capacitive discontinuity [27].

He further explains the several reasons for discontinuities in transmission line such as a right angle bending, vias and change of width along the line. The analytical approach was used for estimation of DDJ in presence of capacitive and inductive discontinuities. Finally, the paper concludes that the impact of transmission line discontinuities is significant on the jitter.

The relationship among different jitter sub-components are presented in [28]. In this paper, the two experiments are performed for jitter analysis based on convolution and superposition techniques. In first experiment, the TJ is calculated from the convolution of different combinations of jitter components, observed from serial data system (refer equation (1)). The RMS and peak-peak value of TJ is almost same for all the combination of jitter components. But this scenario is somewhat different when the superposition of jitter components is used instead of convolution for rms value of jitter. The superposition techniques can work best in case of peak-peak jitter. The TJ_{rms} can be calculated from superposition technique as shown in equation (11).

$$\sigma_{total} = \sqrt{\sigma_{RJ}^2 + \sigma_{PJ}^2 + \sigma_{DCD}^2 + \sigma_{ISI}^2} \qquad (11)$$

In second experiments, the amplitudes of jitter components are changed. The superposition of jitter subcomponents is valid only in the case of peak-to-peak jitter calculation while the equation (1) is valid for both rms and peak-to-peak jitter.

In paper ‘Data Dependent Jitter Characterization Methodologies’, the effect of DDJ on the other jitter sub component which is essential for analysis and characterization of DDJ is discussed [29]. This research work also explains how the increase in jitter can be predicted as a function of different transient response, adjacent line injection and spurious signal coupling. In this paper, MATLAB and HSPICE simulation tools are used for jitter analysis. The effects of location of real and imaginary poles, settling time of response, length of data pattern, bandwidth of low pass filter, high frequency loss such as skin effect, dielectric loss, etc. on ISI and relationship between damping ratio and settling time are also discussed in this paper.

Further, the experimental result shows that ISI has a little effect on RJ (within 10% difference), when RMS value of RJ is fixed to 2.44ps and peak-to-peak value of ISI varies from 5ps to 25ps. Total jitter is same for this case because of measured RJ is greater than the injected RJ while measured ISI is less than the injected and superposition of RJ and ISI gives same TJ as it was injected. In second case, there is no effect of ISI on PJ whether ISI is fixed or variable. And TJ is the same as the summation of PJ and ISI although PJ varies. Finally, the third experiment shows the dependency of ISI on DCD. In this case TJ is about 2ps smaller than the superposition of ISI and DCD, if peak-to-peak value of DCD is fixed while ISI varies. The difference between the measured TJ and the superposition of DCD and ISI is 5%. And when ISI is fixed and DCD varies, TJ increases as DCD increases, i.e. the superposition of ISI and DCD increases as DCD increases.

Jitter components are segregated using tail-fit algorithm, discussed in [30]. The strength and limitation of tail-fit algorithm and there complete flow is discussed in section 3.2.1

Estimation of DJ using de-convolution technique is proposed in [31]. In this paper, the demonstration of DJ is given for real distribution model (not only the simple dual-distribution model). The convolution, de-convolution, linear matrix formulation and Least Square (LS) solution techniques for estimation of DJ are discussed. Further, the strength and limitation of these techniques are discussed in this paper. The approach discussed in this paper is not useful if other jitter components present in the signal whereas only RJ and DJ components can be segregated using this approach.

The research is going on for the improvement in tail-fit algorithm. In [32], the author claims a better method than tail-fitting algorithm for jitter separation based on Gaussian Mixture Model (GMM). In this paper, the methods directly targets on the original TIE vector instead of histogram approach. For this, concept of kurtosis is introduced to determine the order of GMM. Meanwhile, the mathematical relationship between GMM and the quantities of DJ and RJ have been discussed. The authors further privileges that this method bypasses the problem of initial value selection in EM Algorithm and hence more traceable and stable than tail-fit algorithm. Description of EM algorithm is given in [33].

The problems in tail-fit algorithm in presence of sinusoidal jitter and DCD and mitigating the impact of these on RJ were discussed in [34]. In this paper, a new methodology has been introduced for RJ estimation in presence of sinusoidal jitter and duty cycle distortion. The methodology involves calculation of mathematical correction factor, which are used to estimate the accurate value of RJ. The limitation of Dual-Dirac for DJ and RJ measurement are also discussed. Finally, the simulation result shows that percentage error is below 3% for both RJ with sinusoidal jitter as well RJ with duty cycle distortion.

The above discussed paper [30] [31] [32] [34] are only focused on RJ and DJ separations. The other jitter sub-components are not considered in these papers while worst case considered for DDJ estimation in [24] [25] [26] [27]. In this thesis, we considered all the jitter sub-components and analyze the jitter results from tail-fit algorithm and frequency domain algorithm. Further, improvement can be done in these algorithms if needed. Finally, design a new method for ISI estimation using clock pattern.

Chapter 3 Jitter Generator and Segregation Techniques

As described in Chapter 1 jitter can be classified into following categories

1. Periodic jitter
2. Random jitter
3. ISI
4. DCD

This chapter includes the generation and insertion of different types of jitters with the help of their known root causes (as described in Chapter 1) and their mathematical models using MATLAB [28]. Section 3.1 describes the jitter generation and insertion methodology, section 3.2 covers the different types of jitter segregation techniques and problems associated with them.

3.1 Jitter Generator

Jitter in the signal can be primarily inserted through transmitter, channel and receiver. A basic demonstration of jitter source is shown in Fig.3.1

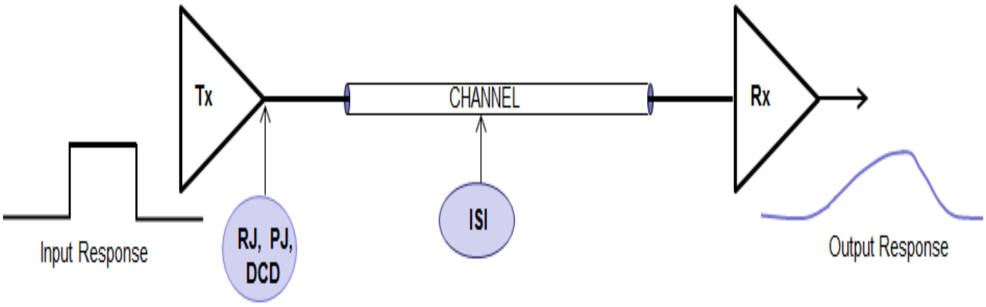


Fig.3.1 A basic demonstration of jitter source.

3.1.1 Periodic Jitter

1. PJ Generator

To generate periodic jitter, a sinusoidal signal is generated. The peak to peak amplitude of the sinusoidal signal defines the total periodic jitter. The mathematical model of PJ is

$$Sinusoidal\,Jitter = \sum_i \text{A}\sin(2\pi f t_i) \tag{12}$$

Here A denotes the amplitude of sinusoidal signal, f denotes frequency and t denotes time. Power supply variations in the circuit are the main source of PJ, which have very low frequency than the input pattern. That’s why the frequency of periodic jitter is very low as compared to input signal.

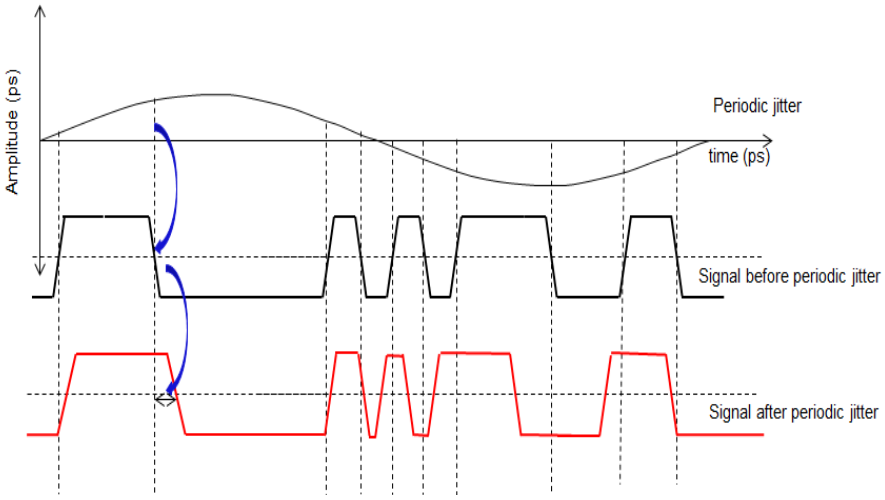


Fig.3.2 Periodic jitter generation.

2. PJ Insertion

In order to insert Periodic Jitter in the input pattern, amplitude of sinusoidal signal at each time instant of zero-crossover points (of PRBS input pattern) is calculated and these values are saved in lookup table. These values are than added to the actual zero-crossover points of the input signal. The resultant signal will be affected from the periodic jitter as shown in Fig.3.2 . To check whether PJ is inserted or not, we calculated the difference of actual signal crossover points and resultant signal crossover points and then we plotted this difference, the resultant graph looks like an exact sinusoidal wave and the histogram of these values is perfectly a PJ histogram. Fig.3.3 shows the complete PJ insertion mechanism. Fig.3.4 shows the histogram and TIE plot of PJ.

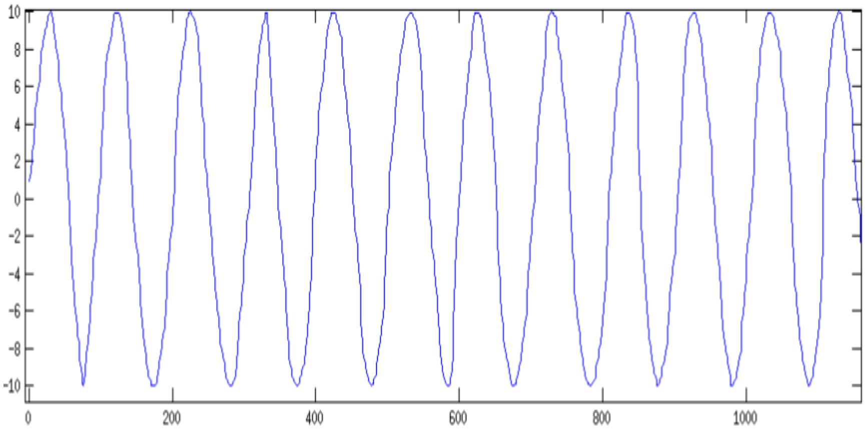


Fig.3.3 Sinusoidal waveform from PJ TIE.

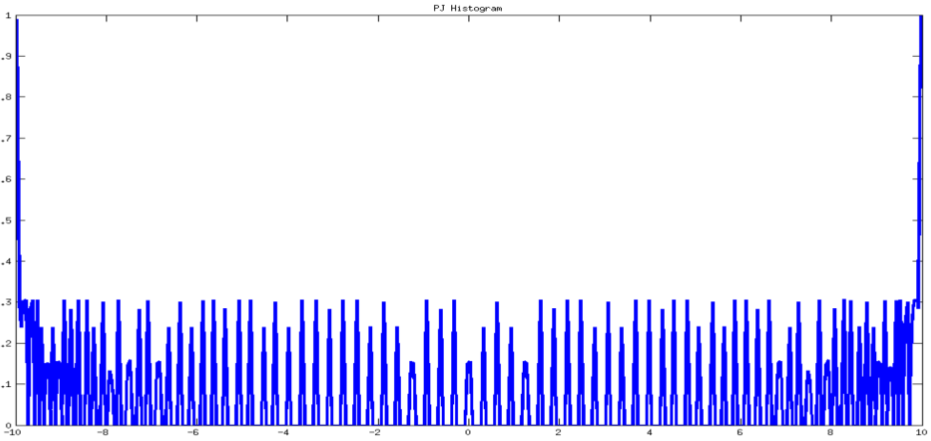


Fig.3.4 PDF of Periodic Jitter.

3.1.2 Random Jitter

1. Random Jitter Generator

To generate random jitter, we first generate a normally distributed random numbers. Mean (μ) and standard deviation (σ) are the primary parameters of a Gaussian distribution function. Sigma value defines the root mean square value of RJ because RJ is always defines in terms of rms value (except if BER is given). Mathematically,

$$Random\,Jitter = \frac{1}{\sqrt{2\pi}\sigma} exp^{\frac{-(t-\mu)^2}{2\sigma^2}} \qquad (13)$$

In MATLAB,

$$Random\,Jitter = normrnd(\sigma, \mu, m, n) \qquad (14)$$

Here m, n are defined as $m \times n$ output vector of random jitter. σ and μ are the standard deviation and mean of the normal distribution.

2. Random Jitter Insertion

To insert RJ in the input signal, the randomly generated numbers are added to the zero-crossover points of input signal. The resultant signal is affected by random jitter.

To check whether RJ is inserted or not, we plot a PDF of these randomly distributed number and it looks like the Gaussian distribution curve shown in Fig.3.5

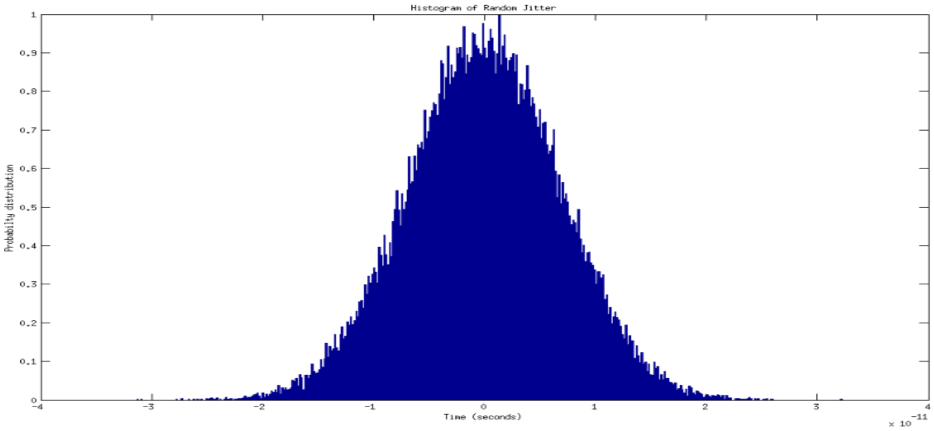


Fig.3.5 PDF of Random Jitter

3.1.3 Duty Cycle Distortion (DCD)

1. DCD Generator

DCD is caused due to difference in propagation delay between low to high transition and high to low transition and due to variation in threshold voltage. Fig.3.6 shows the DCD insertion mechanism.

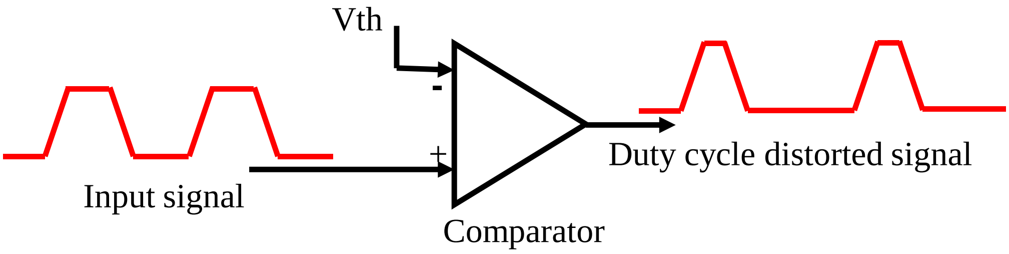


Fig.3.6 DCD Generation and insertion mechanism.

2. DCD Insertion

By shifting the threshold voltage level to some other level, DCD can be inserted in the input signal. The crossover points at newly defined threshold voltage are calculated and the difference of new crossover points and input signal crossover points are TIE of DCD. For verification, whether the inserted jitter is DCD or not, we plot a histogram as shown in Fig.3.7

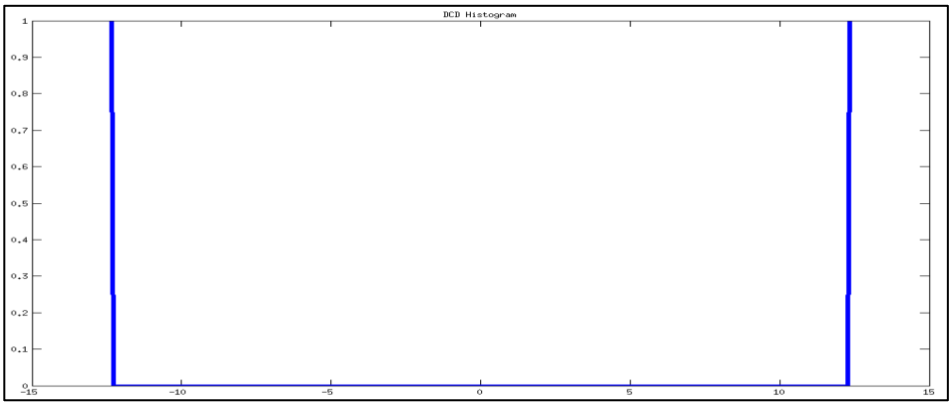


Fig.3.7 DCD Histogram.

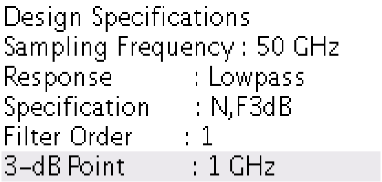
3.1.4 Inter-symbol Interference (ISI)

1. ISI Generator

The signal affected with ISI is distorted in terms of their rise and fall time due to limited channel bandwidth. This behavior of a channel can be modeled as a low pass filter as shown in Fig.3.8 . To introduce ISI effect in the signal, first we design a first order low pass filter whose cutoff/3-dB frequency should be less than the bandwidth of the original signal [35]. The 3-dB frequency should be dependent on the rise time of the input pattern.

Mathematically,

$$Cutoff\ frequency(f_c) = \frac{0.35}{t_{r_{10\ to\ 90}}} \tag{15}$$



Design Specifications
Sampling Frequency : 50 GHz
Response : Lowpass
Specification : N,F3dB
Filter Order : 1
3-dB Point : 1 GHz

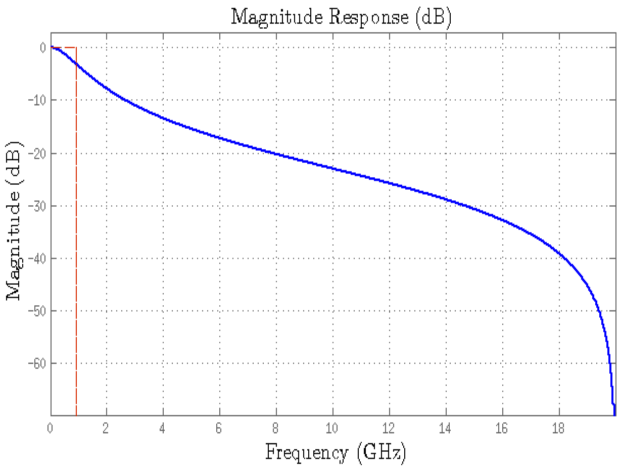


Fig.3.8 Low-pass Filter

2. ISI Insertion

Response of low-pass filter is shown in Fig.**3.8** .

Input pattern is passed through this low pass filter. The variation in rise time and fall time in each bit is responsible for ISI. The resultant signal of channel is distorted. The crossover points of the output response are calculated and save these crossover points in a file. The ISI insertion methodology and the input pattern v/s pattern with ISI effect are shown in Fig.3.9 and in Fig.3.10.

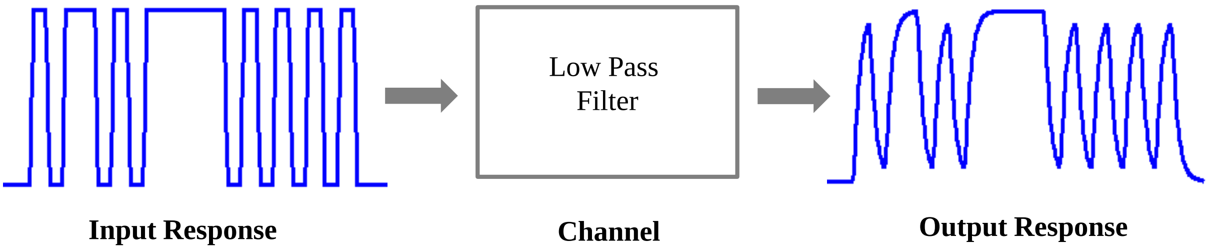


Fig.3.9 ISI insertion mechanism

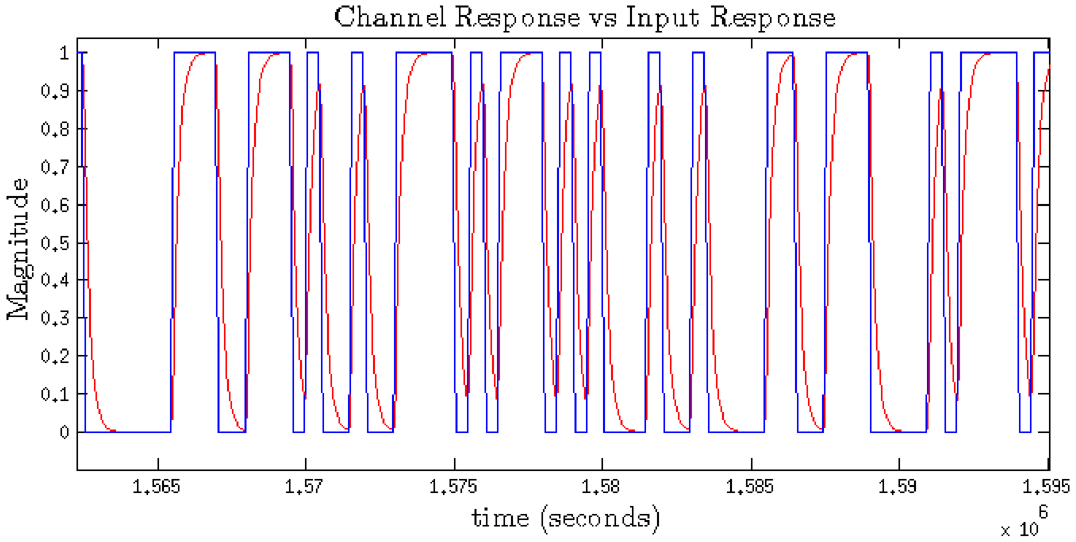


Fig.3.10 Channel Response VS Input Response.

3.1.5 Total Jitter

The TJ can be calculated by convolution of PDF of individual jitter components as well as superposition of peak-to-peak values of jitter components as shown in equation (16) and (17). To get the total value of jitter, we add the entire jitter component that have generated in the above discussion and then insert this TJ in the input signal. Finally, we get the jittered signal that includes PJ, RJ, DCD and ISI. The crossover points are saved in “.matfile” format.

$$TJ_{pk-pk} = PJ_{pk-pk} + RJ_{pk-pk} + ISI_{pk-pk} + DCD_{pk-pk} \tag{16}$$

$$TJ_{PDF} = PJ_{PDF} * RJ_{PDF} \tag{17}$$

Where $*$ denotes convolution operator.

3.2 Jitter Segregation Techniques

3.2.1 Tail Fit Algorithm

The tail fit algorithm is based on convolution technique and probability density function. The total jitter distribution is the convolution of RJ and DJ distribution, as shown in Fig.3.11 (refer equation (17)). The overview of tail fitting algorithm is shown in Fig.3.12. Some preprocessing steps are required on the histogram before tail fit algorithm for more accuracy and robustness. These steps are standardization of data, removing outliers, interpolation, normalization and smoothing the histogram to remove statistical noise. The detailed description of these steps is given below.

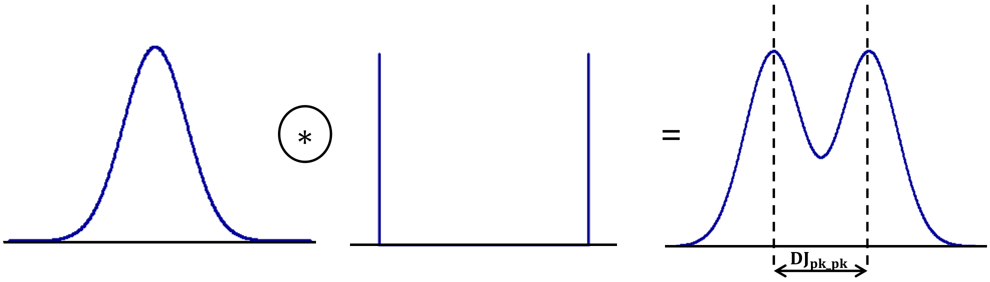


Fig.3.11 Convolution of RJ and Dual Dirac model.

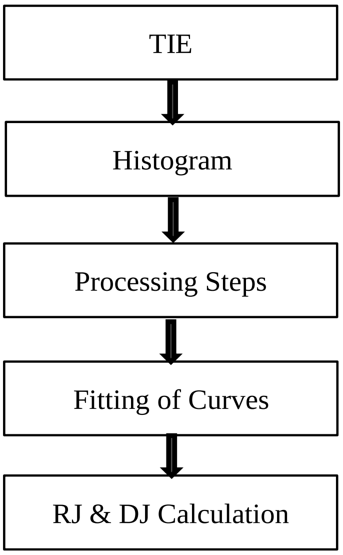


Fig.3.12 Overview of tail-fit algorithm.

Steps for Tail fit algorithm

The flow chart for the implementation of tail fit algorithm in MATLAB is shown in Fig.3.13.

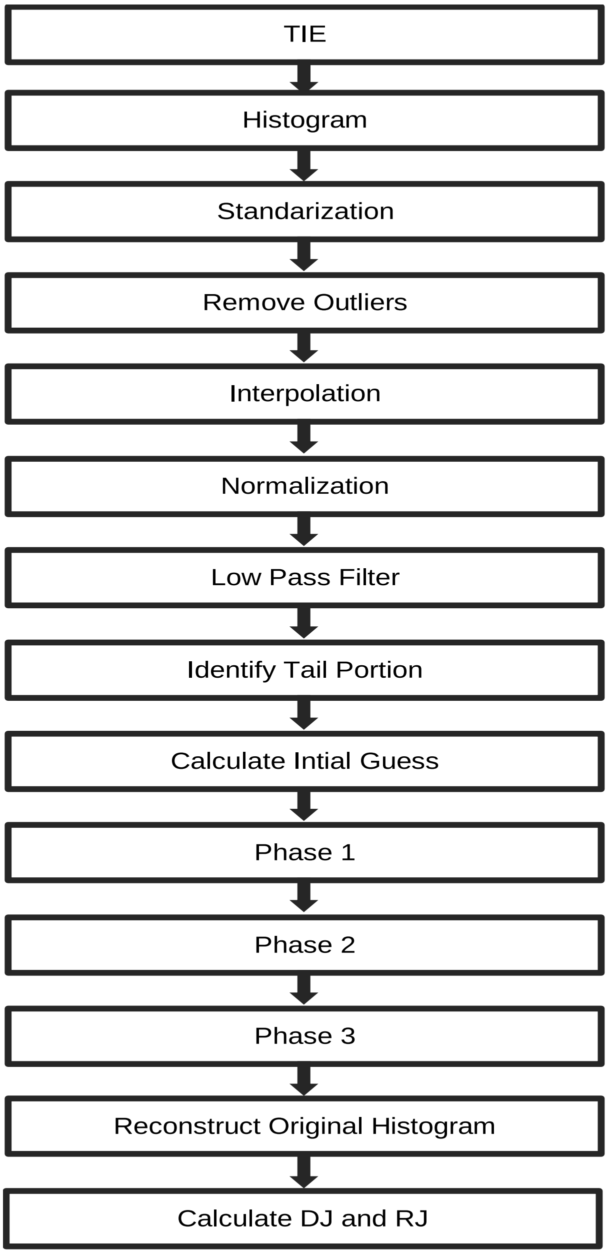


Fig.3.13 Flow Chart of Tail-fit Algorithm.

1. Load TIE

For calculating TIE value, we first calculate the crossover point of a jittered signal from Agilent Digital Software (ADS) or from jitter generator. In this report, we use a 511 binary pattern sequence, which is repeated 100000 times. The jittered signal have RJ and PJ component. After calculating the crossover point, this data is loaded in MATLAB in .mat file format or .txt format. By taking the nearest integer value of these data using *'round'* command in MATLAB, we get the reference zero-crossover points, shown in equation (18). And then subtract these reference signal from jittered zero-crossover points, finally we get TIE value (refer equation (26)).

$$Reference\ Signal_{crossoverpoints} = \frac{round\left(Jitter\ signal_{crossoverpoints}\right) * Data\ frequency}{F_s} \qquad (18)$$

2. Prepare histogram of TIE data

The histogram that is formed from original TIE data needs improvement in terms of accuracy and robustness. This is because the tail fit algorithm does not provide correct result, if using original histogram.

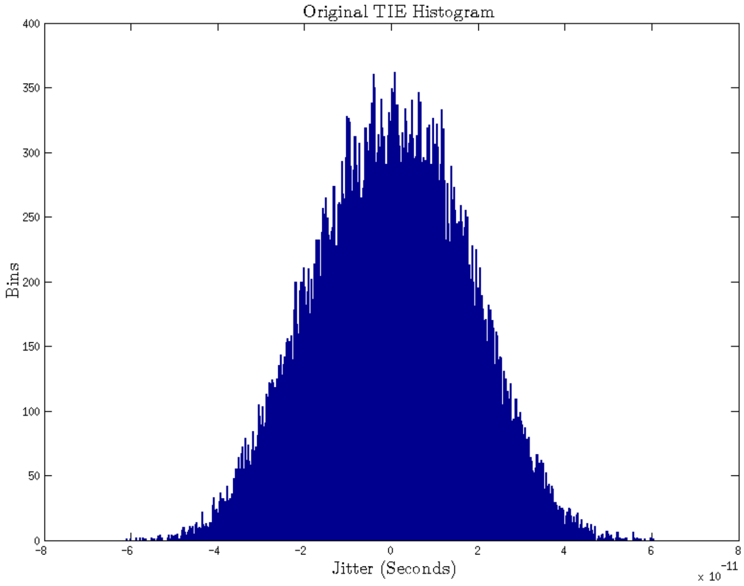


Fig.3.14 TIE Histogram.

The following steps are useful for accurate and smother histogram for tail-fit algorithm.

a) Removing data trend
b) Standardizing the data
c) Removing histogram outliers
d) Linearly interpolating the empty bins
e) Normalizing the histogram
f) Smoothing the histogram

a) Removing data trends

The clock is not very stable, so that the trends are formed and there is continuous negative slope in the jittered data. This is done by *'de-trend'* function in MATLAB. But in our case this step is excluded because our data is stable and having a constant slope. Therefore, no effect on the data before and after de-trend.

b) Standardizing the data

The resolution of original jittered data is very small and MATLAB have one disadvantage that it does not give accurate result for very low resolution. Hence, standardizing technique is used for the low resolution data that translates into a more practical scale. And after the best fit of Gaussian curve, the Gaussian parameters are translated back to the original data. Formula for Standardization is given in equation (19).

$$x' = \frac{x-\mu}{\sigma} \tag{19}$$

Here x', x, μ, σ denotes the TIE data after standardization, original TIE data, mean and standard deviation respectively.

Standard deviation of data can be calculated with the help of '*std*' command in the MATLAB.

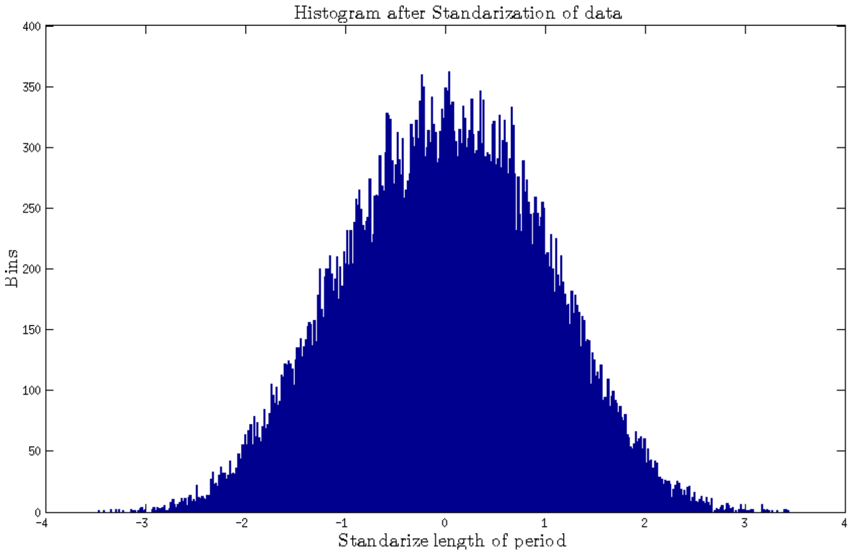


Fig.3.15 Histogram of TIE after Standardization of Data.

c) Plot histogram

For histogram of data, '*hist*' function is available in MATLAB. We consider that the 400 bins are sufficient to analyze the TIE data. By default the "*hist*" command consider only 10 bins.

d) Removing Outliers

Outliers are present on both sides of the histograms and they could adversely affect the fitting accuracy. Outliers generally defines as a value outside the 3-sigma. After removing these outliers we can achieve 99.7% accuracy. Therefore, we discard the outliers which are greater than 3σ.

The outliers are removed by looking both sides of the histogram for a bin with at least one period that is more than 5 bins away from the next occupied bin. If such a bin is found, the bin count reduced to zero. In our case we check for 5% bins on both the sides of histogram. The histogram after removal of outliers is shown in Fig.3.16.

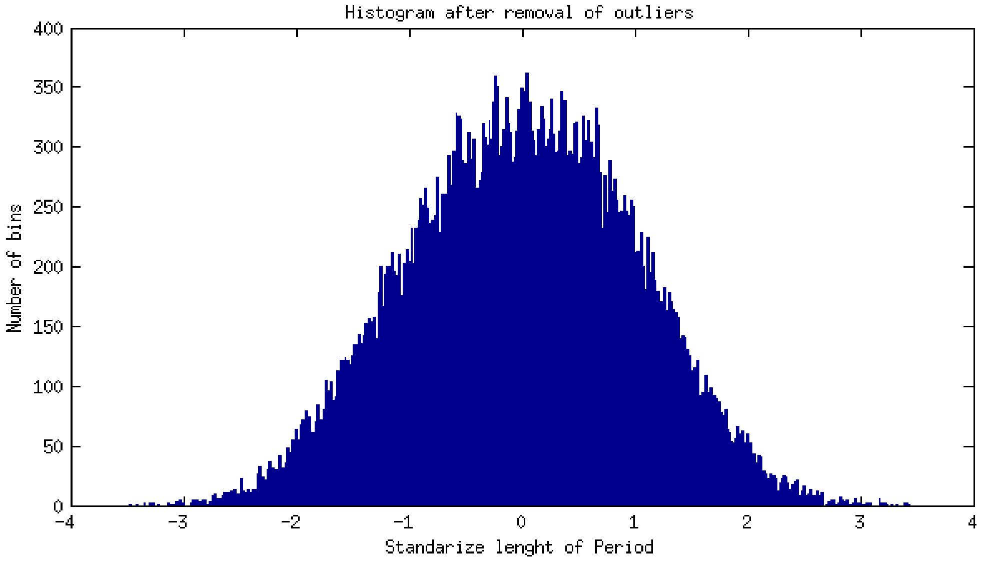


Fig.3.16 Histogram of TIE after Removal of Outliers.

e) Linearly Interpolate

If the data does not have enough significant digits, thus the histogram created above have some empty bins. For example the sequence likes 10100001. We see that the four consecutive zeros are present and hence we cannot get the crossover point for these values because crossover points only available where transition happens and in TIE data these bins are empty. Therefore, we linearly interpolate these empty bins.

f) Normalization

The histogram's bins are normalized to 1 as shown in Fig.3.17. This ensures that the Gaussian PDF and the histogram tails have same vertical scale. The Gaussian PDF that is being fit to the histogram is also normalized to 1. The advantage of normalization is when both histogram and Gaussian PDF are normalized to one, so that transferring of the tail magnitude to the PDF is simple because both are multiplied by the same number.

$$Normalization = yhist/\max(yhist) \qquad (20)$$

g) Moving through the filter

The filter removes the statistical noise from the data and smoothens the histogram. Eleven bins are used for providing adequate smoothing without distorting the histogram. Histogram is padded by 5 zero bit on either side before it passing through the filter and then taking averaging of 5 bits from right and 5 bits from left of a particular bit and this bit is replaced by the averaged value.

In our case, the first five bins of histogram are padded to zero bins and next 6th bin to 405 bins are equivalent to yhist and the remaining 406 to 410 are zero bins. In the next step, we take the average of 5bins from left and right of a particular bin and replace its value with average value and this process continuous until whole yhist bins are covered. The histogram after filtering statistical noise is shown in Fig.3.17.

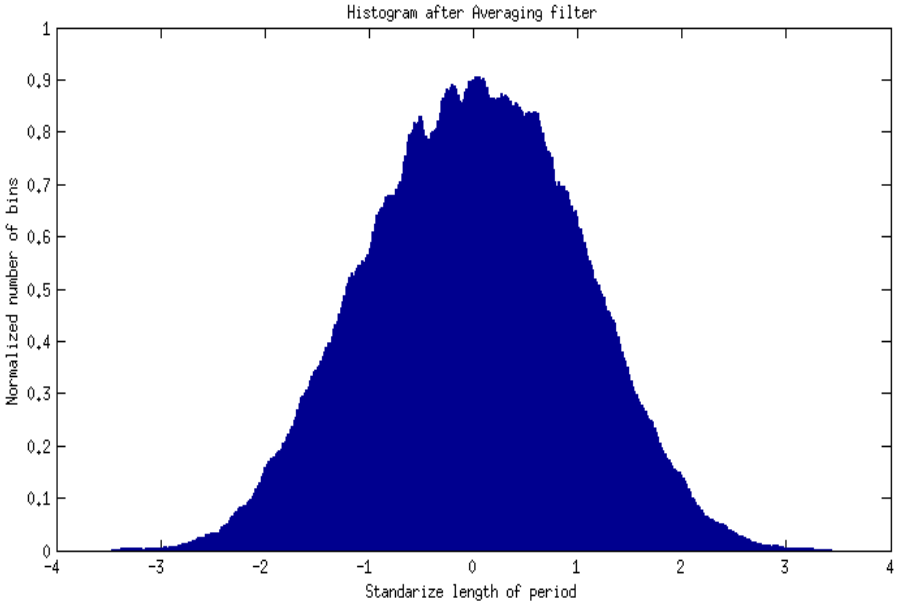


Fig.3.17 Histogram of TIE after Averaging Filter.

3. Identifying the tail portion

For finding the left and right tails, the histogram is scanned from both left and right side to the middle of histogram (i.e. nbins/2) and search for the maximum bin which is greater than the next 'n' bins and that bin is the local maxima of the tail portion. The tail portion of the histogram from left/right end of the histogram to the local maxima plus 5% to the total bins. These additional 5 bins help us to calculating the initial guess. The procedure is same for both left and right tails. For left tail, we searching bins from left to middle while for right tail identification, searching bins from right end to middle portion. The algorithm of identifying tail portion is shown in Fig.3.18.

4. Calculation of Initial Guess

The initial parameters of Gaussian distribution can be calculated from the tail portion, after cropping the histogram. These parameters are mean (μ), standard deviation (σ) and magnitude of both the tails and these parameters help us in the phase 1 of the tail-fitting.

For magnitude calculation, we take the maximum magnitude of 20 bins from the left tail and right tail and then by averaging of all these magnitude, we get magnitude of left tails and right tail. For mean calculation, we take the average of maximum bin position and for sigma value; we take the standard deviation of the histogram that have the range of from 1:190 for left tail and the range from 210:400 bins for the right portion of histogram. Finally, we plot the Gaussian curve by using these parameters for fitting the tails of original histogram as shown in Fig.3.23 and in Fig.3.26.

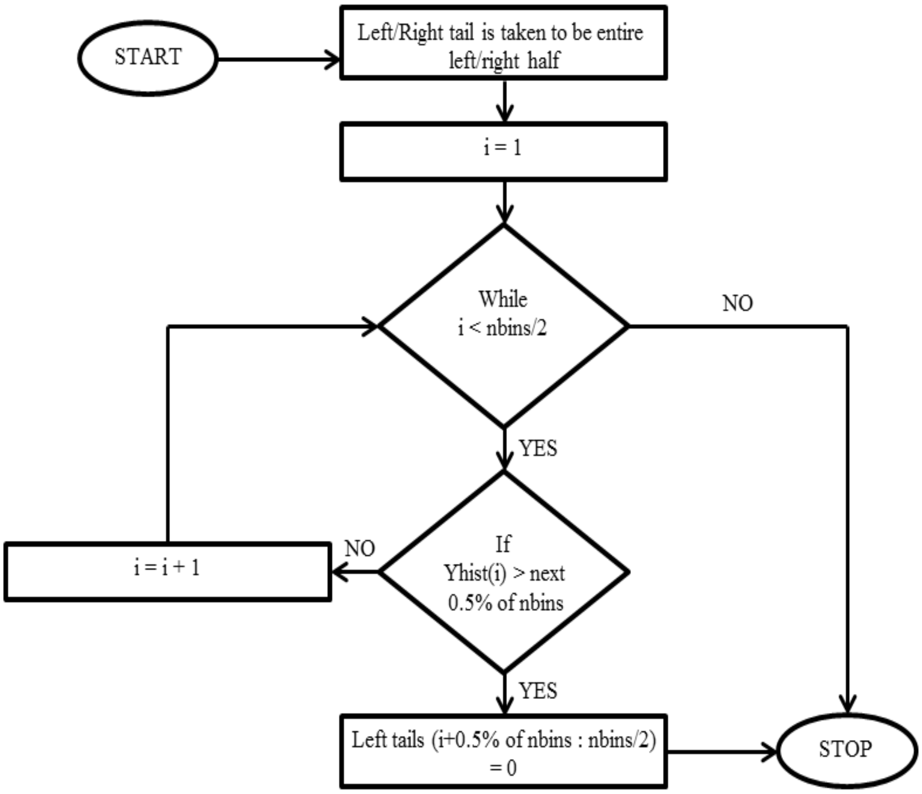

START
Left/Right tail is taken to be entire left/right half
i = 1
While i < nbins/2
NO
YES
If Yhist(i) > next 0.5% of nbins
NO
i = i + 1
YES
Left tails (i+0.5% of nbins : nbins/2) = 0
STOP

Fig.3.18 Flow Chart of Identification of Tail Portion.

5. Three Phases of Fitting

Phase 1

The phase 1 starts by setting up a range of values for both the (σ) and (μ) for each tail. These parameters of each Gaussian curve are varied from minimum limit to maximum limit by a step value. The mean in phase first is varied from ±50% from initial value and standard deviation is varied from ±40% from the initial value with a step size of 5% of (σ) and (μ). These limits provide sufficient range for all data sets and additional range for overall robustness of algorithm.

Phase 1 Algorithm

After setting the minimum and maximum limit for mean and standard deviation for both left and right tails, we find out the best curve fit using chi-square test [46]. For best fit, we first set both the σ and μ to a minimum limit. In the next step, increment in the σ value with a step size of 5% to maximum limit of σ with constant μ. And then set to σ value to its minimum limit and increment in the μ with a step size of 5% of initial μ. Each σ and μ is run through the chi-squared test subroutine to determine its best chi-squared value.

The flow chart of phase 1 and phase 2 of curve fitting is shown in Fig.3.20.

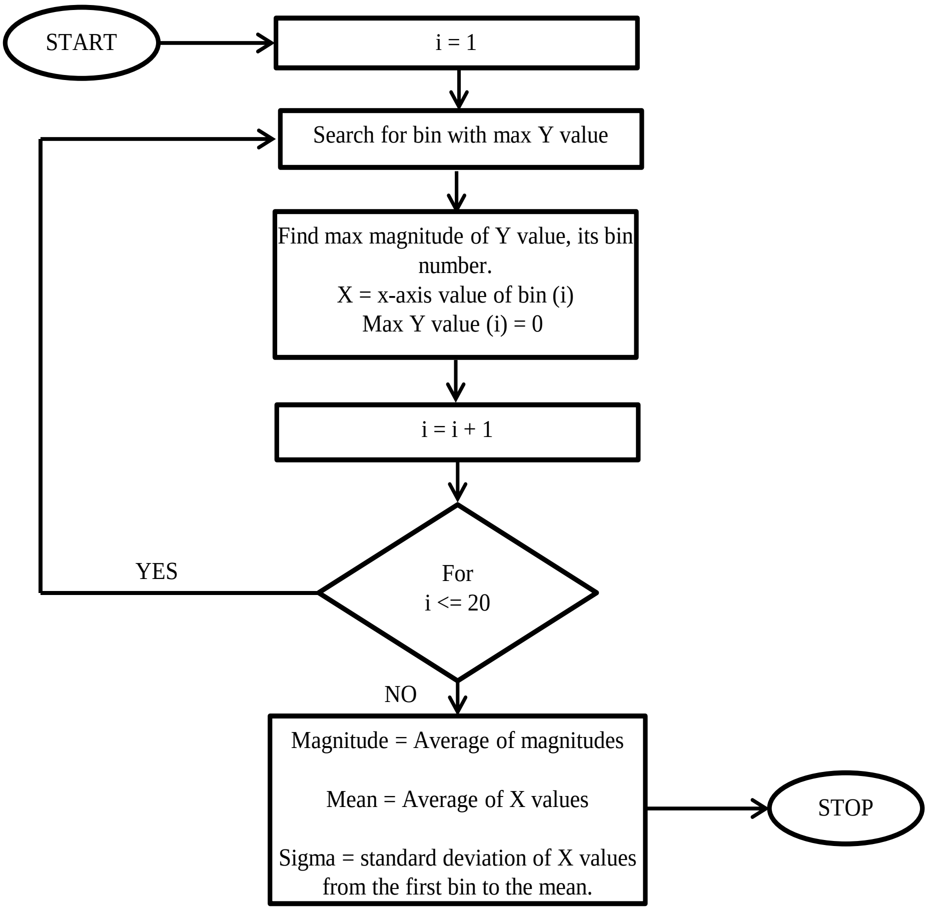

START
i = 1
Search for bin with max Y value
Find max magnitude of Y value, its bin number.
X = x-axis value of bin (i)
Max Y value (i) = 0
i = i + 1
YES
For
i <= 20
NO
Magnitude = Average of magnitudes
Mean = Average of X values
Sigma = standard deviation of X values from the first bin to the mean.
STOP

Fig.3.19 Flow chart of initial guess algorithm.

Phase 2

Second phase of curve fitting is similar to first one except the minimum and maximum limit of μ is ±6% of phase1 and for σ the range is ±12% of phase1 and the step size is 0.5 percent of Gaussian curve parameters . The remaining process is same as we follow in phase 1.

Phase 3

Third phase of curve fitting is sligtly different from phase 1 and phase 2. In this phase, the final value of σ and μ from previous phase are used as the starting value of σ and μ. The step size is 1% for both the σ and μ value. The algorithm reduces with the small increament in step size. The flow chart of phase 3 is shown in Fig.3.21.

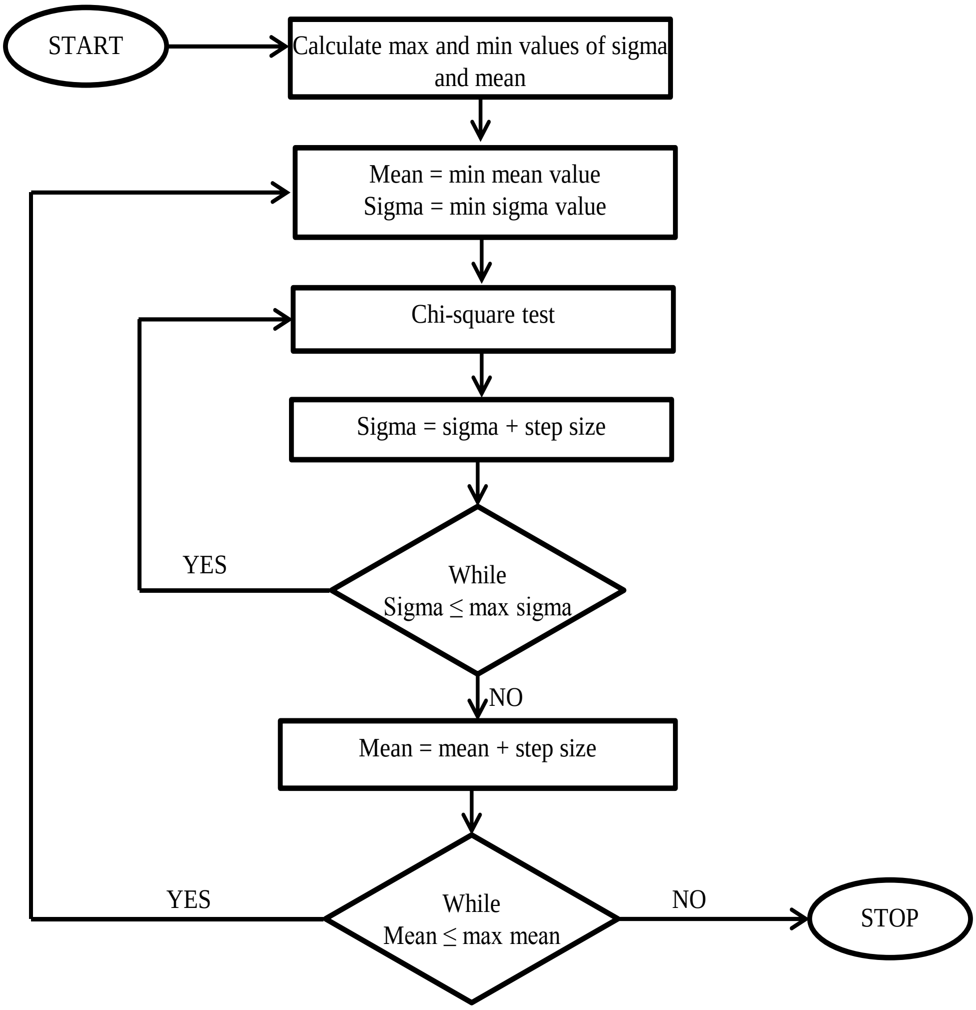


Fig.3.20 Flow Chart of Phase 1.

Phase 3 Algorithm:

The algorithm is initialized after setting the step size of σ and μ for phase 3. Initially, the values of σ and μ are the same as the final value of previous phase. Then the algorithm enters in “While” loop. The loop runs while the step > 0.001%. Initially given a step value of 1%, the previous phase of σ value is incremented and decremented by one step while the μ doesnot change. This is followed by a chi-square test. The test only saves the value if they have a better fit. If neither of the first two had a better fit, the number would stay the same. The current sigma value becomes this value.

Again the mean is adjusted in same manner as sigma adjusted, first increase and then decreases with step size and then followed by the chi-square test. At the end of each loop the step value is divided by 1.1 and loop continous. From this we get the best σ and μ value for tail fitting of gausian curve.

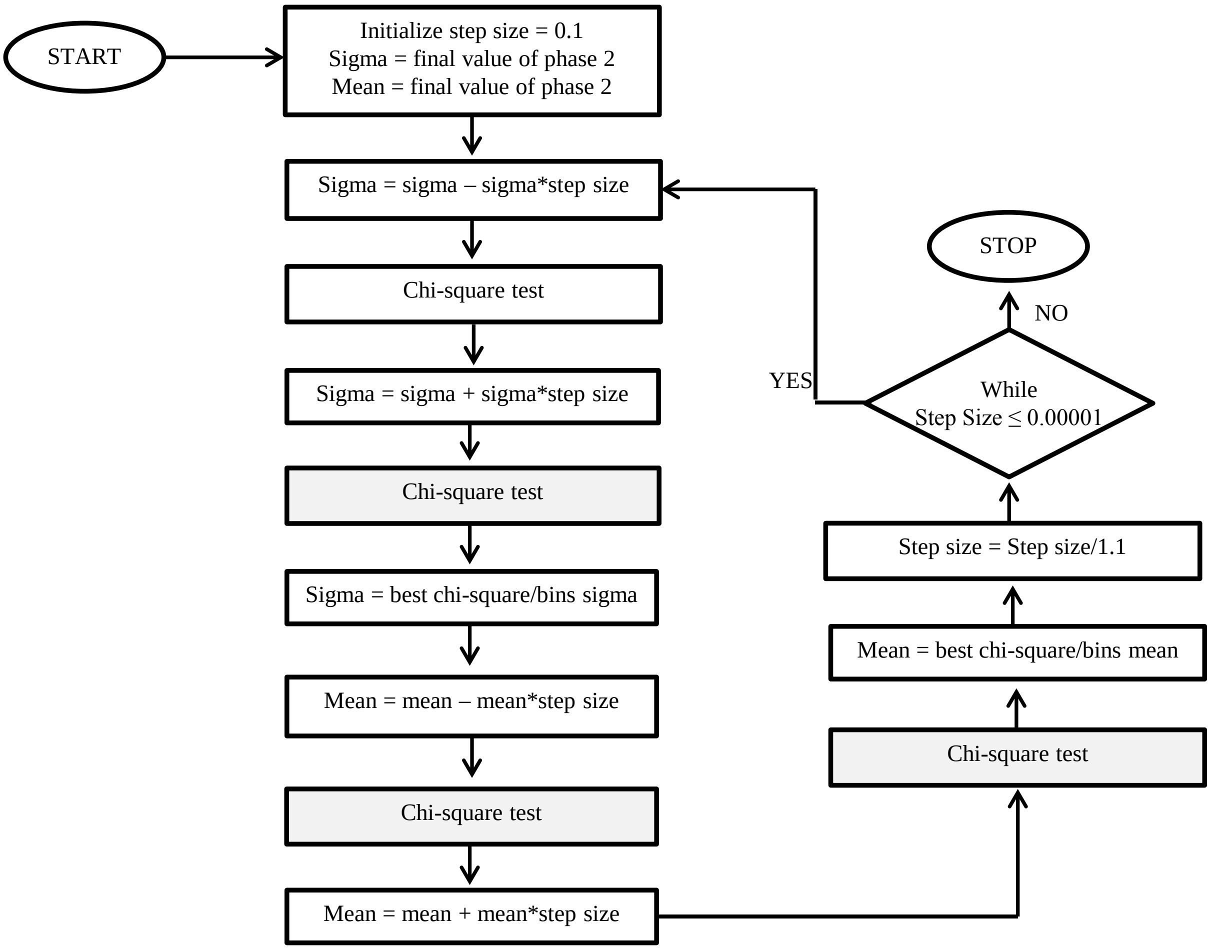

START
Initialize step size = 0.1
Sigma = final value of phase 2
Mean = final value of phase 2
Sigma = sigma – sigma*step size
Chi-square test
Sigma = sigma + sigma*step size
Chi-square test
Sigma = best chi-square/bins sigma
Mean = mean – mean*step size
Chi-square test
Mean = mean + mean*step size
Chi-square test
Mean = best chi-square/bins mean
Step size = Step size/1.1
While
Step Size ≤ 0.00001
YES
NO
STOP

Fig.3.21 Flow Chart of Phase 2 and Phase 3.

6. Calculation of final Gaussian curve parameters

After finding the best fit for the tails, we have to still find out the final σ and μ value for calculating the DJ and RJ. The formula for calculating final μ and σ are given below-

$$\sigma_{final} = \sigma_{data} * \sigma_{phase3} \tag{21}$$

$$\mu_{final} = \mu_{phase3} * \sigma_{data} + \mu_{data} \tag{22}$$

This algorithm is not designed in presence of DDJ. When ISI convolves with PJ and RJ, the TJ curve cannot simply be considered to be composed of DJ which has a dual Dirac distribution and RJ which has a gaussian distribution.

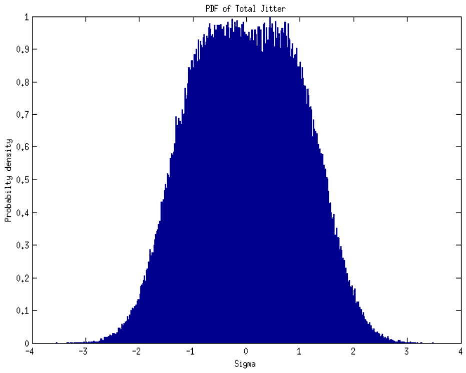


Fig.3.22 Histogram of TJ

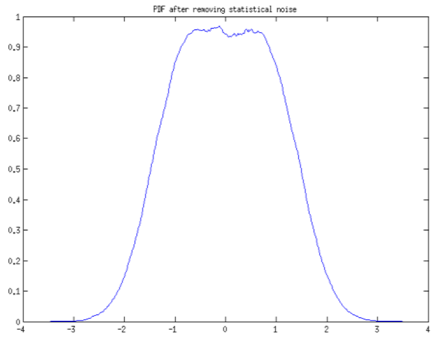


Fig.3.25 Histogram after pre-processing steps

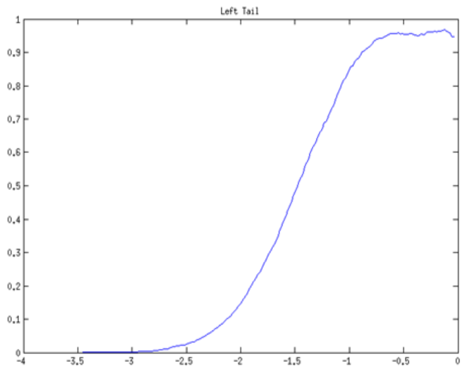


Fig.3.23 Left Tail extraction

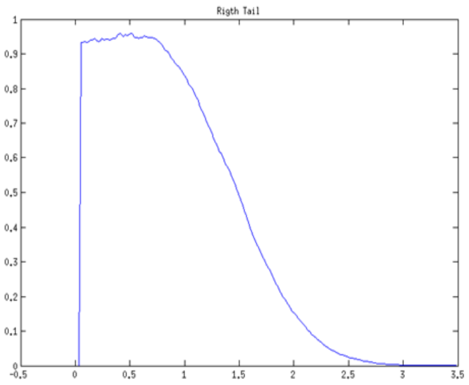


Fig.3.26 Right Tail Extraction

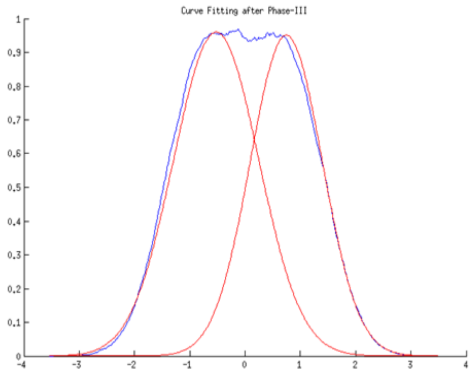


Fig.3.24 Fitting of tail after Phase 3

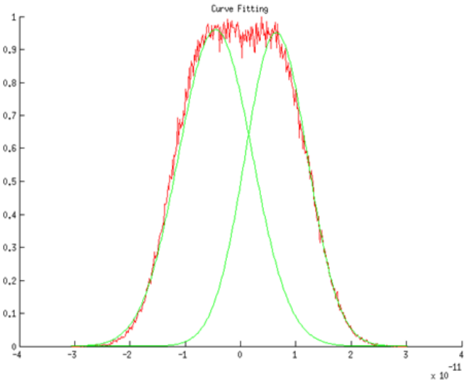


Fig.3.27 Conversion into original histogram

3.2.1.1 Problems in Tail fitting algorithm

The tail fitting algorithm gives best result when TJ is composed of Gaussian distribution function and Dual Dirac distribution, shown in Fig.3.28. But in practical case, the probability distribution of DJ is Dual Dirac distribution is very less. Therefore, in presence of small amount of ISI in the TJ gives an error, as seen in case 1, 5, 14 in Table 1.

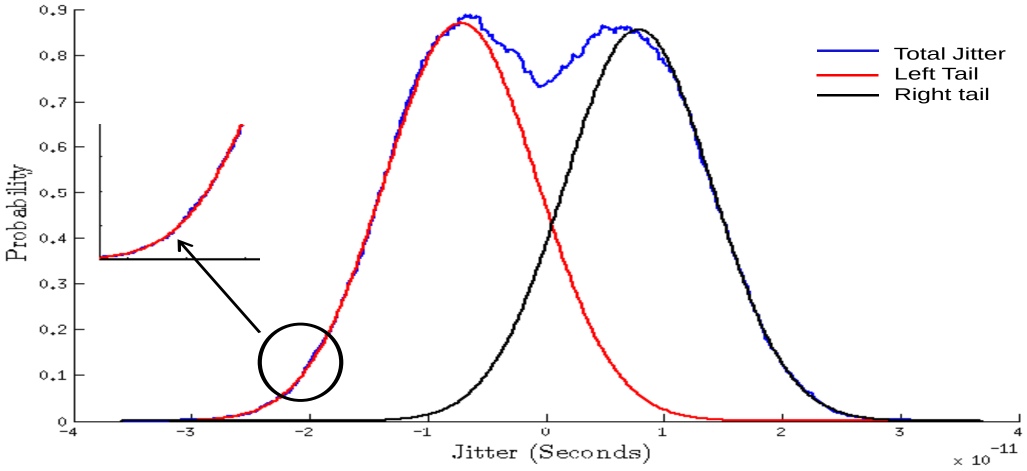


Fig.3.28 Tail-fit Result in Presence of RJ and DCD.

Another problem in original tail fitting algorithm is that there are some outliers are found in a very low probability region. Due to this the total histogram shifts to one side and the tail of the histogram misidentified and misfit by tail fitting algorithm, shown in Fig.3.29.

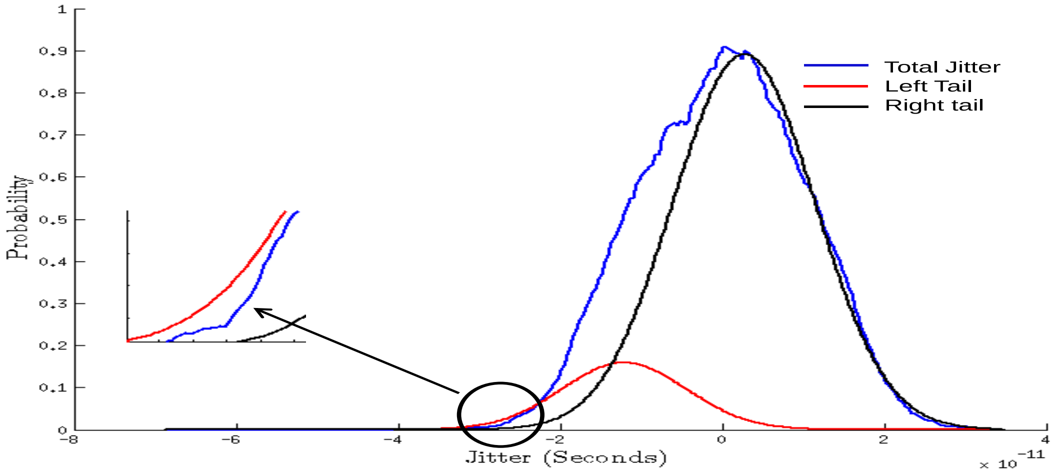


Fig.3.29 Tail-fit Result in Presence of ISI.

3.2.2 Improved Tail Fit Algorithm

The above algorithm gives significant error due to presence of DDJ(specially ISI). So to avoid these error its better to remove DDJ component before applying tail fit algorithm. Two extra steps are included in improved algorithm. After calculation of TIE data, next step is interpolation of data. And after interpolation, we first extract the amount of DDJ present in TJ and then we apply original tail-fit algorithm. The flow chart of improved tail fir algorithm as shown in Fig.3.31.

1. Interpolation of the data

Interpolation of data is done only to simplfy the process of DDJ. To identify bits which are interpolated we use a position vector. This vector is same length as interpolated TIE and contains a value of 0,1 and 2 for every rising edge, falling edge and interpolated data respectively. Flow chart of interpolation of data is shown in Fig.3.30.

Algorithm

First define a time reference and then linearly interpolate the refernce signal with time refernce. In the next step a position vector that have same length as interpolated TIE is constructed. Initially this postion vector equates with zero. The algorithm starts with a "*for*" loop for 1:length(tie). Inside the *"for"* loop is an *"While"* loop for j is less than equal to length of interpolated tie. In while loop is an "if" statement, which compares the original TIE and interpolated TIE. This comparison is for finding interpolated data position. If the differnce is greater than the defined limit in if statement. At that position 2 is allotted in position vector and then i and j are incremented. The For loop and While loop are continuous and finally we assign 0s and 1s alternatively where value 2 is not found.

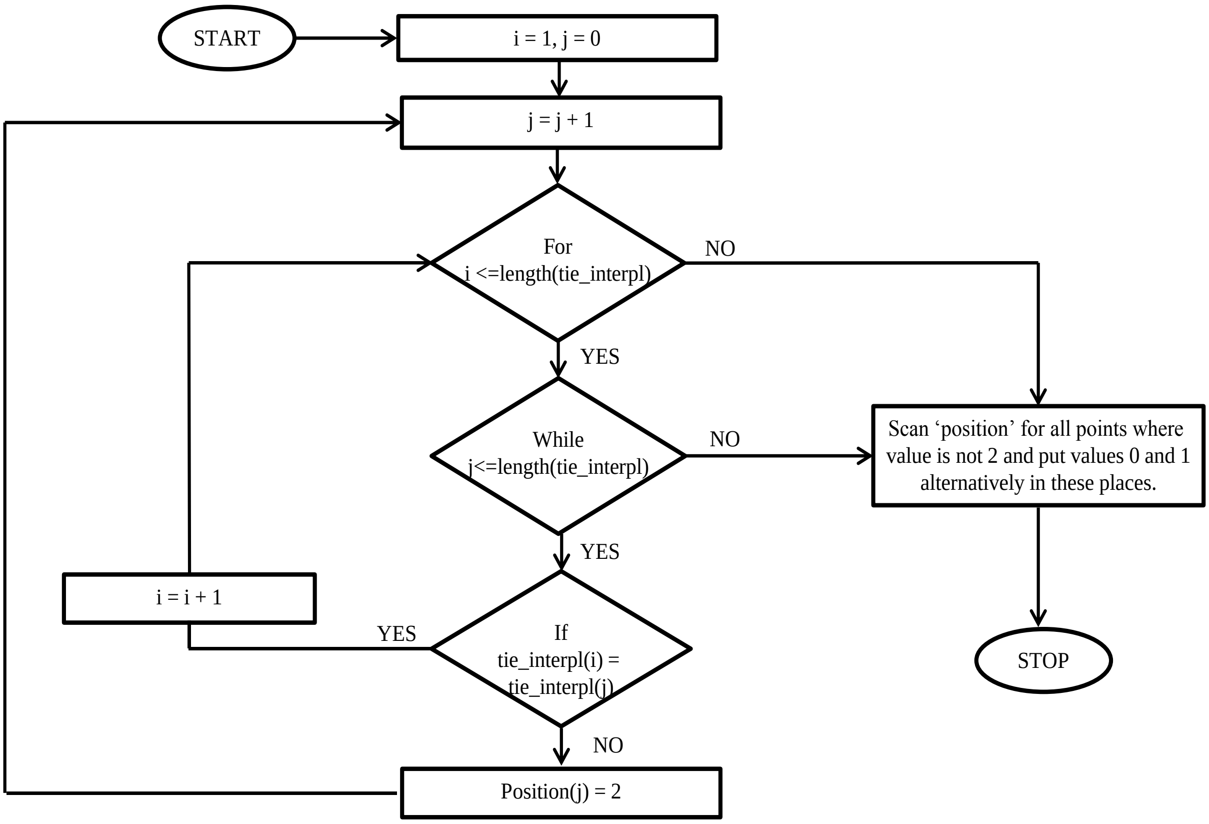


Fig.3.30 Flow Chart of Interpolation of Data

2. Obtaining DDJ

DDJ have one special property that it repeats if the data pattern repeates [7]. Thus if the data pattern is PRBS-9, then DDJ repeates after every 511 bits. If we take the average of the total jitter over a

very large samples than PJ and RJ gives a μ of zero. This is because, RJ and PJ are uncorrelated to data pattern. And there is no effect on DDJ because it is data dependent.
Mathematically,

$$\frac{1}{M}\sum_{m=1}^{M}\Delta t_m(t_n) = \frac{1}{M}[\sum_{m=1}^{M}\Delta t_{DDJ_m}(t_n) + \sum_{m=1}^{M}\Delta t_{PJ_m}(t_n) + \sum_{m=1}^{M}\Delta t_{RJ_m}(t_n)] \qquad (23)$$

$$\frac{1}{M}\sum_{m=1}^{M}\Delta t_m(t_n) = \Delta t_{DDJ}(t_n) \qquad (24)$$

Steps for Improved Tail Fit Algorithm

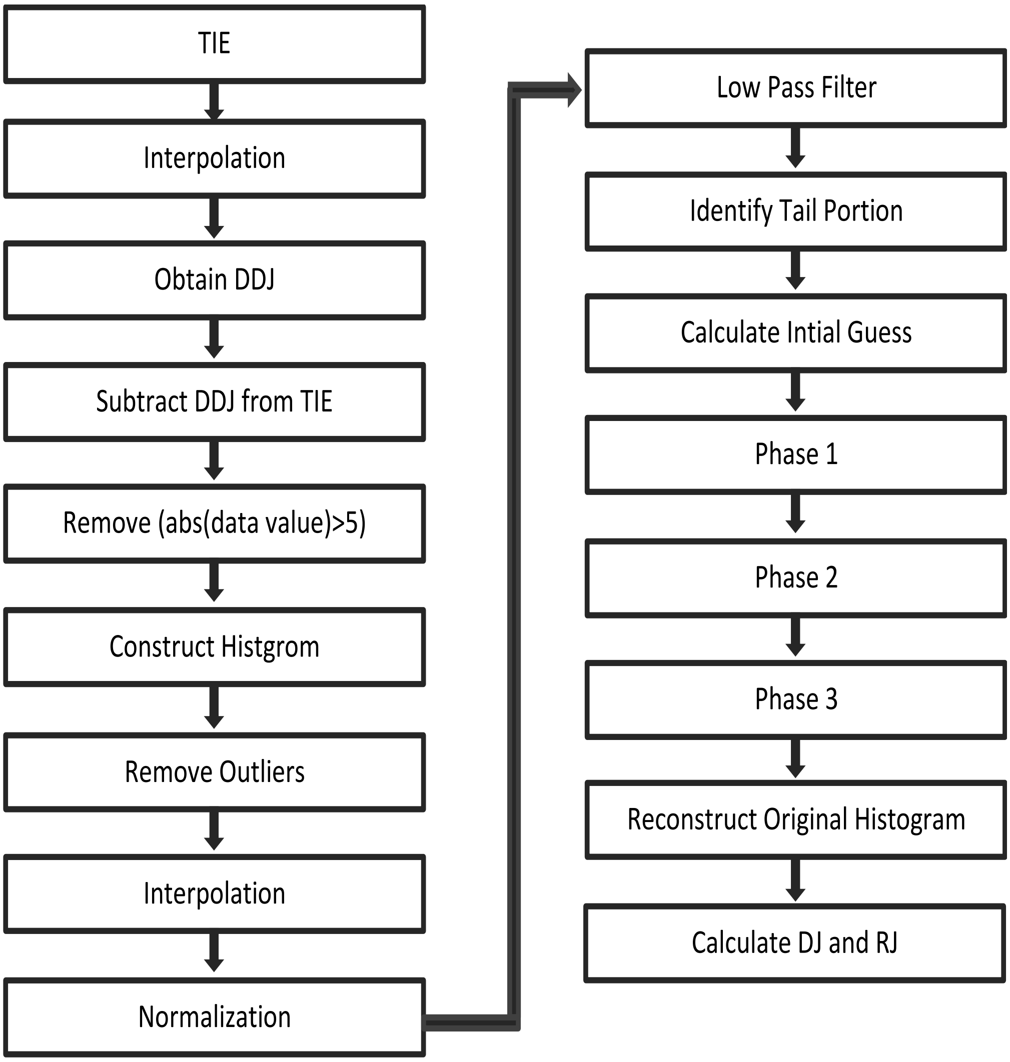


Fig.3.31 Flow Chart of Improved Tail-fit Algorithm.

3.2.2.1 Problems

Results of improved tail-fit algorithm are better than original tail fit algorithm, shown in Table 2 . But still the error percentage is above 20 %, which is due to the presence of periodic jitter in the test pattern. The PJ is a sinusoidal having very low frequency as compared to input pattern. Each point in the PJ histogram behaves as an impulse response, shown in Fig.3.32. Therefore, there are lots of Gaussian curves that are superimposing upon each other and the tail parts of the overall jitter distribution are adversely affected while fitting, and an error is created in the measurement of RJ, shown in Fig.3.33and in Fig.3.34.

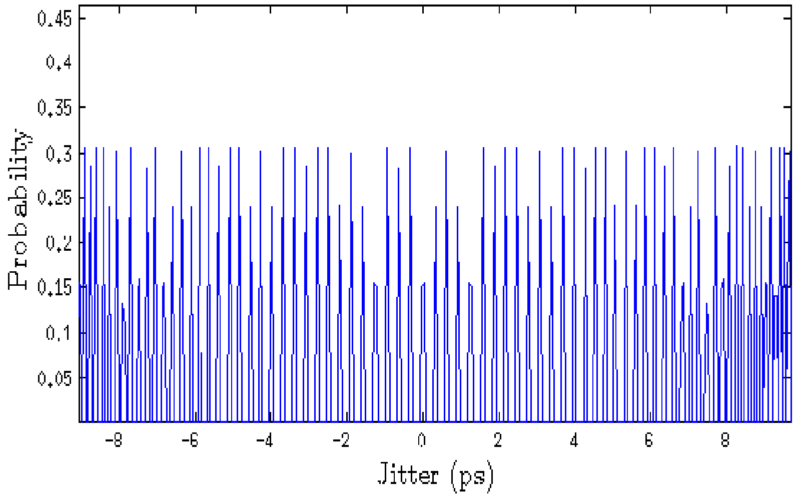


Fig.3.32 PJ Distribution

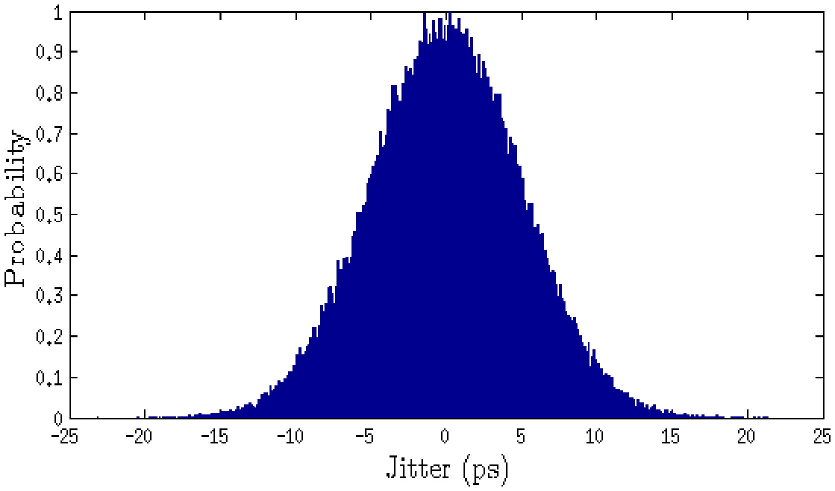


Fig.3.33 RJ Distribution

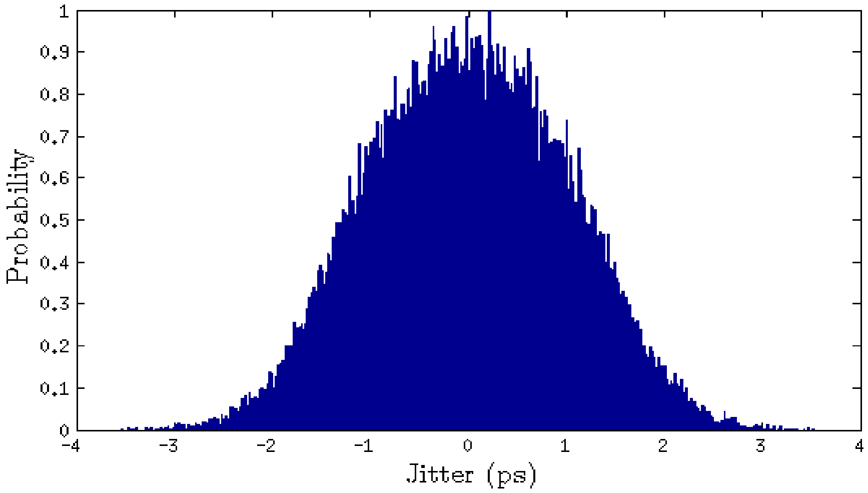


Fig.3.34 Overall Distribution

3.2.3 Jitter estimation using Clock Signal

The problem arises in tail fit algorithm is that it gives large errors in presence of DDJ components and doesn't give the accurate results for RJ and DJ separation.

In this approach, a clock signal is applied parallel to the original pattern (PRBS pattern). It is assumed that the effect of jitter is same on both clock pattern and PRBS pattern except that ISI component of jitter is not present in the clock signal [7]. To obtain accurate results of DJ and DDJ, a spectrum approach gives us better result. This is implemented either the FFT based method or PSD method [35].

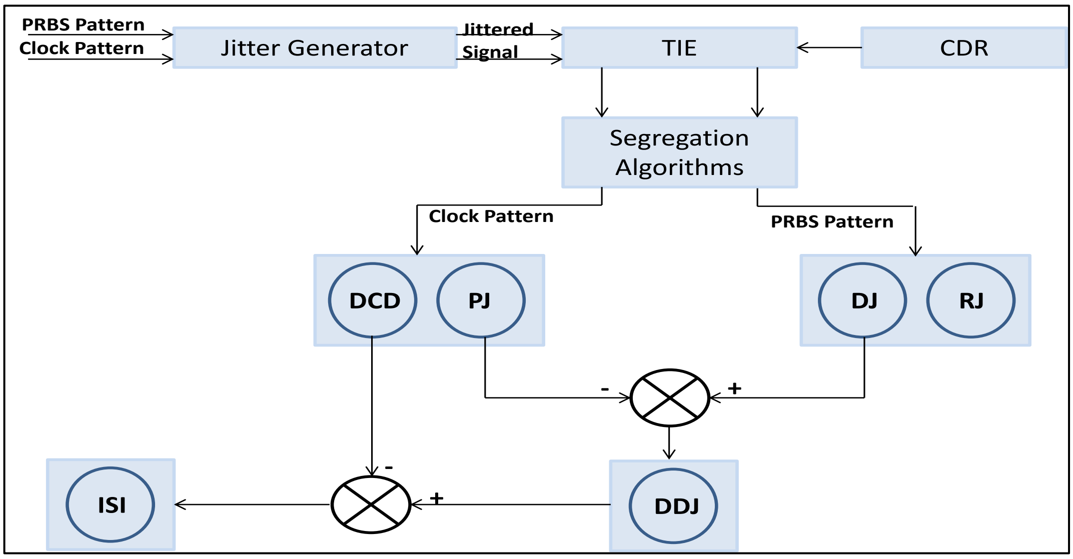


Fig.3.35 Block Diagram of Jitter Estimation using Clock Pattern.

The block diagram shown in Fig.3.35 shows the complete demonstration of jitter test setup which includes insertion of jitter in the signal and then estimation of PJ, DCD from PRBS pattern and RJ and DJ from clock pattern using FFT. And then finally, subtraction of PJ from DJ followed by subtraction of DCD from DDJ, gives ISI component from TJ. The detailed explanation of this has been explained below.

Input Signal

The input signals are both PRBS pattern and clock pattern. The clock signal is useful for ISI estimation. Some assumption for this approach-

1. The length of both the pattern (i.e. clock pattern and PRBS patter) should be the same.
2. The PRBS pattern should be repeated at least 50 times.

Jitter Generator

Jitter generation and insertion methodologies are discussed in section 4.1. The input pattern is passed through this generator. The output signal of this is degraded in terms of parameters like, rise time, fall time, bit period, amplitude distortion, etc.

Crossover Points

The time instant at the zero voltage for NRZ bit pattern or mid-voltage level of NRZ (Non-Return to Zero) bit pattern of each rising and falling edge is define as crossover points. We calculate the crossover points of jittered as well as reference (input) signal. The crossover points can be obtained in MATLAB using-

$$[Crossover, mid\_voltage] = midcross('file\ name', 'sampling\ frequency') \tag{25}$$

TIE

Time interval error is the difference of time between reference signal crossover points and jittered signal crossover points. The reference signal can be obtained using equation (19).

$$TIE = Ref_{crossingpoints} - Jittered_{crossingpoints} \tag{26}$$

Interpolation

The data have long string of consecutive 0’s and 1’s, TIE data is not complete. For instance, since there is no transition edge between two consecutive 1s, no TIE is recorded for that edge; But PJ and RJ have existence at every bit interval boundary. Therefore, interpolation of TIE is important. In this approach, linear interpolation method is used. The algorithm is shown in Fig.3.30.

Truncate interpolated TIE to make power of 2

This approach is based on FFT, and FFT works best when the input data set has a length of 2 [36]. Ensure that the number of sample points is a power of 2 for better result. If the number of sample points is more, then the data has to be truncated.

RJ and DJ separation

Estimation of RJ and DJ via tail fitting algorithm gives large amount of error. To estimate more precise value of RJ and DJ, frequency spectrum method is better [35].

Follow the following steps for separation RJ and DJ and the flow chart as shown in Fig.3.36

1. Time domain to frequency domain – The time to frequency domain operation is made to TIE data of PRBS pattern, the TIE frequency spectrum is obtained. This can be done either by a Fast Fourier Transform (FFT), a Discrete Fourier Transform (DFT), or a Laplace Transform (LT) and so on. In this case, FFT method is used to compute fast operation.

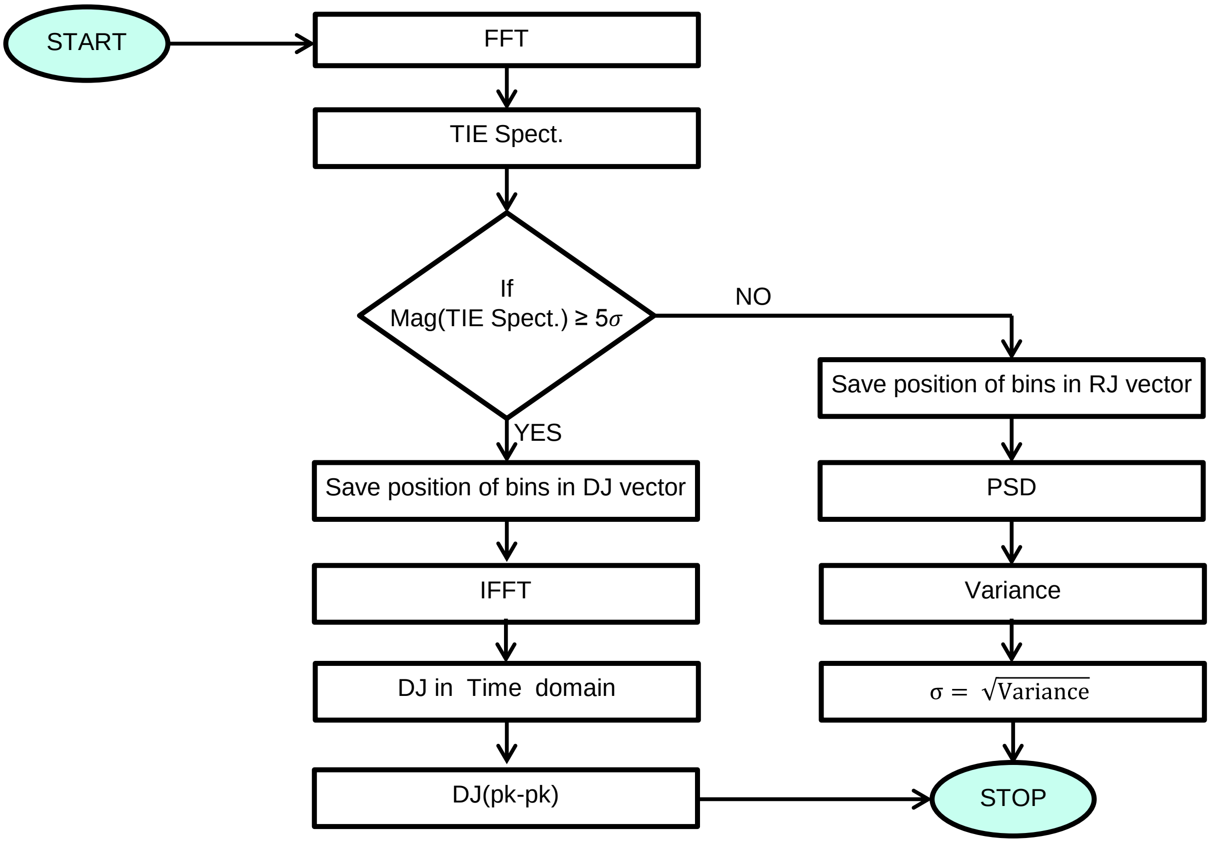


Fig.3.36 Flow Chart of RJ and DJ Estimation.

2. Deterministic jitter estimation – DJ can be separated from total jitter by taking FFT of TIE data (obtained from PRBS pattern). The deterministic jitter shows up spikes or spectral lines in frequency domain, making it easier to identify and quantify, shown in Fig.3.37 and in Fig.3.38. By extracting these spikes from frequency spectrum, DJ spectrum is obtained as shown in Fig.3.39.

   These spikes are extracted from TIE spectrum by finding the points where the spectrum has a value greater than N*σ as shown in Fig.3.39. Normally, the range of N varies from 3 to 5 [35]. The value of N depends on the number of sample points. Specifically, larger the sample points, lesser is N and vice versa. For accurate measurement of DJ, the value of N should be taken very precisely.

   This spectrum is converted in time domain by using Inverse Fourier Transform. Take the histogram of this time domain data and the difference of maximum value and minimum value of this gives deterministic jitter.

3. RJ estimation – Subtract the DJ spectrum form spectrum of TIE. The positions where the DJ spikes are extracted are interpolated. This steps provides the spectrum for random jitter as shown in Fig.3.40.

   The PSD of RJ spectrum provides us the information of how its energy is distributed over the frequency. Integration of PSD over a range of frequency gives us standard deviation of random jitter.

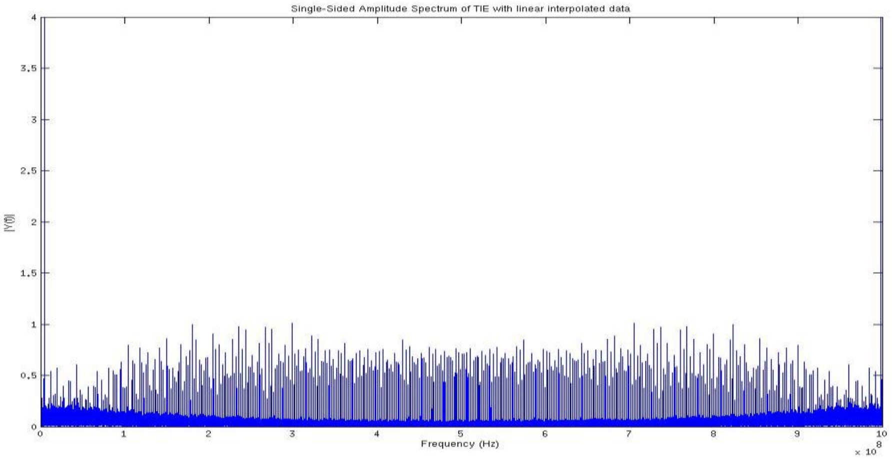


Fig.3.37 Spectrum of Interpolated TIE Data

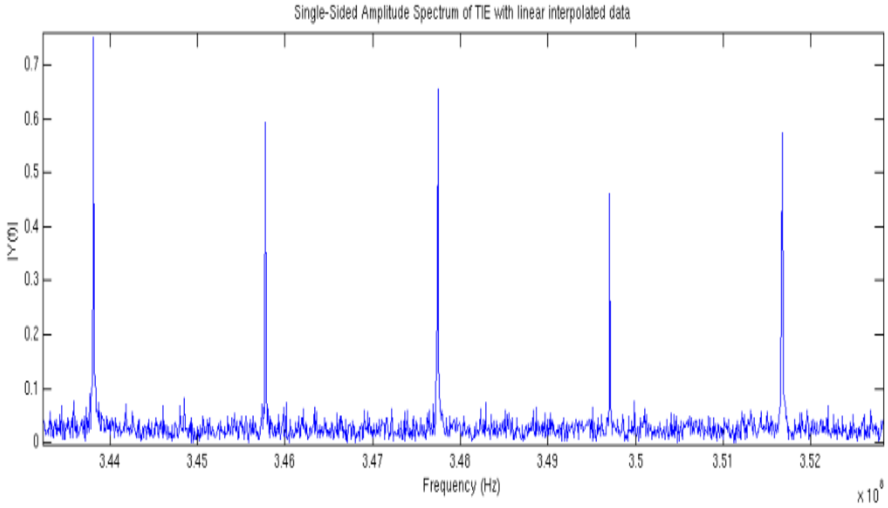


Fig.3.38 Zoomed Spectrum of Interpolated TIE Data

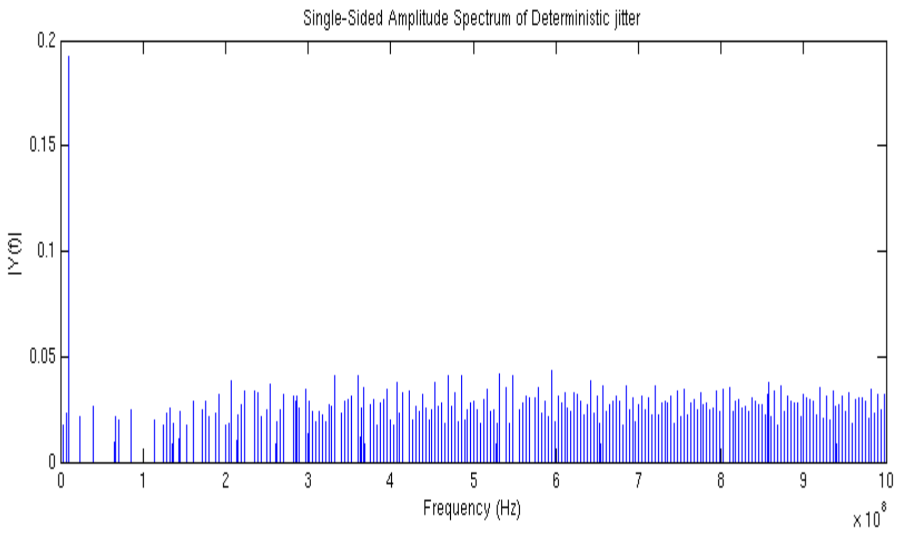


Fig.3.39 DJ Spectrum

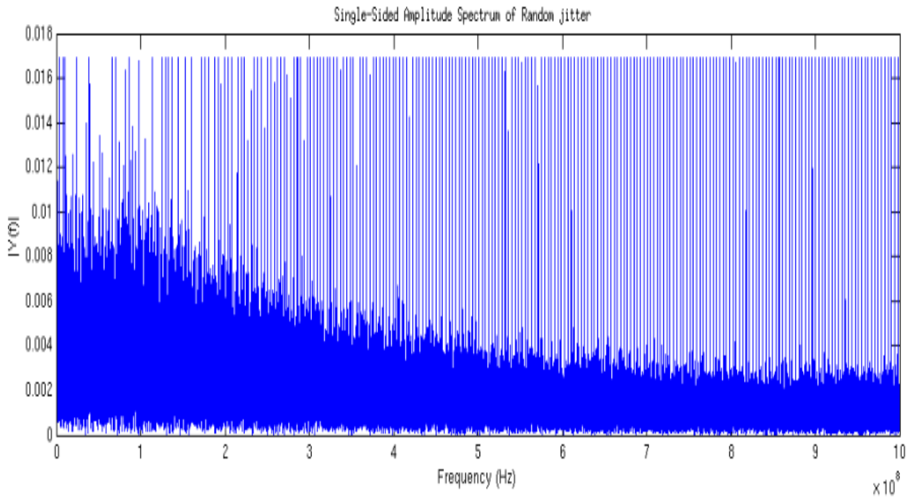


Fig.3.40 RJ Spectrum.

PJ and DCD estimation

Clock pattern can be used for PJ and DCD estimation. As it is defined earlier that ISI is not present in the clock pattern when it passed through a channel. Only DCD, PJ and RJ components are present. The flow chart of PJ and DCD estimation as shown in Fig.3.41

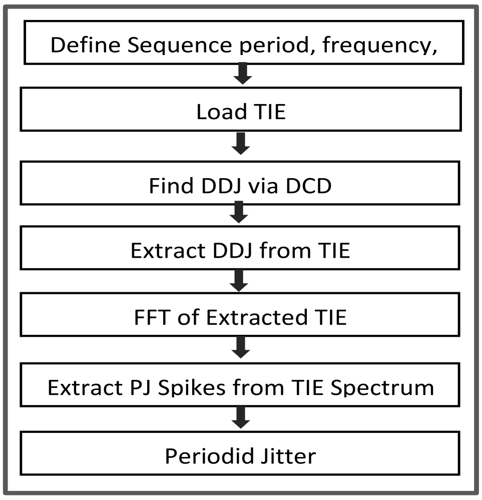


Fig.3.41 Flow Chart of PJ and DCD Estimation.

Steps for measurement of PJ and DCD:

1. Define the sequence of clock pattern. The sequence is always 2 because clock pattern has alternative 0 and 1 in the whole sequence. Also define sampling frequency that is useful in calculation of reference signal crossover points as discussed earlier.
2. Load the TIE data that is obtained by subtraction of reference crossover and jittered crossover signal.
3. Third step is to obtain DDJ value their extraction from total TIE data value. The procedure of extraction of DDJ is exactly same as used in improved tail fitting algorithm. The only difference is in the input pattern.
4. Duty Cycle distortion has a property that each of rising edge is advanced/delayed by the same amount and similarly each of falling edge is also advanced/delayed by the same amount. Thus DCD can be obtained using the following equation

$$DCD = mean(TIE\ of\ rising\ edges) - mean(TIE\ of\ falling\ edges) \qquad (27)$$

   This equation effectively finds the difference between the mean of the rising edge tie and falling edge tie.

5. Once DDJ is subtracted from the jitter vector, the remaining jitter vector only contains PJ and RJ. PJ can also be obtained by using FFT. Periodic jitter appears as spikes within the spectrum. Extract these spikes from the spectrum by defining the points where spectrum value has greater than N*σ. Mathematically,

$$TIE_{spect} \geq N\sigma_{rms} \qquad (28)$$

   PJ spectrum can be transformed into time domain to obtain peak to peak value of PJ.
   It is known that deterministic jitter is composed of periodic jitter and data dependent jitter.
   Mathematically,

$$TJ = DJ + RJ \qquad (29)$$

$$DJ = DDJ + PJ \qquad (30)$$

$$DDJ = DJ - PJ \qquad (31)$$

$$ISI = DDJ - DCD \qquad (32)$$

6. DDJ and PJ are the part of DJ. Thus, DDJ can be obtained using the equation. Subtraction of PJ (obtained from clock pattern) from DJ (obtained from PRBS pattern) that includes ISI, DCD and PJ.
7. From equation, we obtain ISI by subtracting DCD from the result of step6 (i.e from DDJ).
   Finally, we estimate all the jitter components values. The results shown in Table 3 and in Table 4 are better than tail fitting algorithm.

3.2.4 Frequency Domain Algorithm

The error percentage for ISI can be further reduced in this algorithm. The ISI can be calculated before that of converting TIE data from time domain to frequency domain. In this algorithm, we use the DDJ vs bit waveform. From this we can obtain total DDJ excursion for both rising and falling edges [37]. And then using spectrum analysis, we estimated PJ and RJ components. The flow chart of this algorithm is shown in figure.

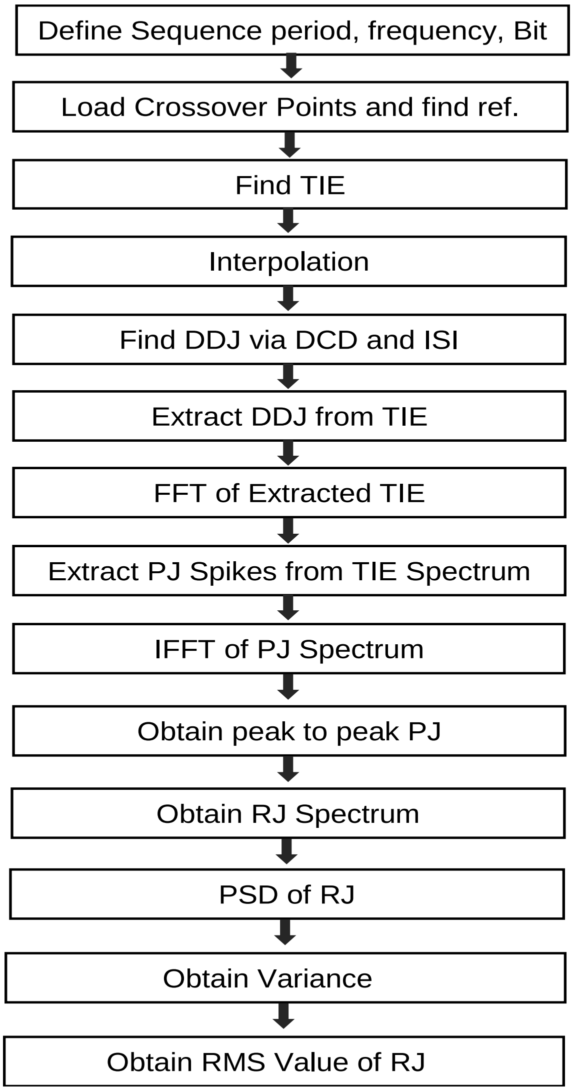


Fig.3.42 Flow Chart of Frequency Domain Algorithm.

Algorithm Description

The first steps in this algorithm are same as of previous algorithm. The only difference is that in sequence period step. The sequence period of pattern is needs to be known in advance in this algorithm. We consider the sequence period of pattern in this work is 511. Another difference in this algorithm is that the input pattern is Pseudo random pattern.

The length of interpolated TIE data is limited to a power of 2. If the number of samples points is more, then the data has to be truncated while if number of samples points is less, then the extra zeros are added in the total data. The next steps are estimation of DDJ, DCD and ISI from time domain data.

DDJ Estimation

DDJ has a special property that it repeats if data pattern repeats. Besides, PJ and RJ give a mean of zero over a large number of samples.

Steps for DDJ calculation-

- In the first step, a window of maximum repeatable pattern length is applied on interpolated TIE vector. For example if a pattern repeats itself after every 511 bits, first 511 bits will be selected in first step and then the window will be shifted to next 511 bits.
- In next step, data is arranged row wise in a matrix.

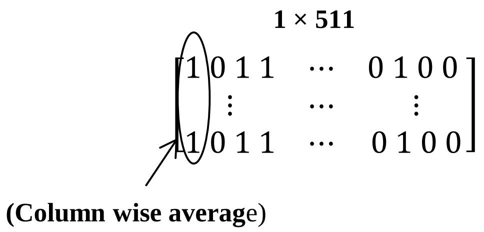


Fig.3.43 Estimation of DDJ

- DDJ with respect to each bit is obtained by taking the average of each column as shown in Fig.3.43 (Refer equation (23) and (24)).
- Place DDJ with respect to bit vector in a vector of size which is equal to the interpolated jitter vector. From this DDJ vs time waveform; we can obtain DDJ (peak-peak) value.

DCD and ISI Extraction

For obtaining the amount of DCD present we make use of the DDJ waveform obtained earlier. Now, to obtain DCD we take note of the fact that due to DCD, each of rising edge is advanced/delayed by the same amount and each falling edge is also advanced/delayed by the same amount. Thus, DCD can be obtained using equation (27).

To obtain ISI we use the DDJ vs bit waveform. From this waveform we obtain the total DDJ excursion for the rising and falling edges. A mean of these values gives us ISI.

Mathematically,

$$ISI = \frac{1}{2}\left[\left\{\max(DDJ_{risingedge}) - \min(DDJ_{risingedge})\right\} + \left\{max\left(DDJ_{fallingedge}\right) - min\left(DDJ_{fallingedge}\right)\right\}\right] \quad (33)$$

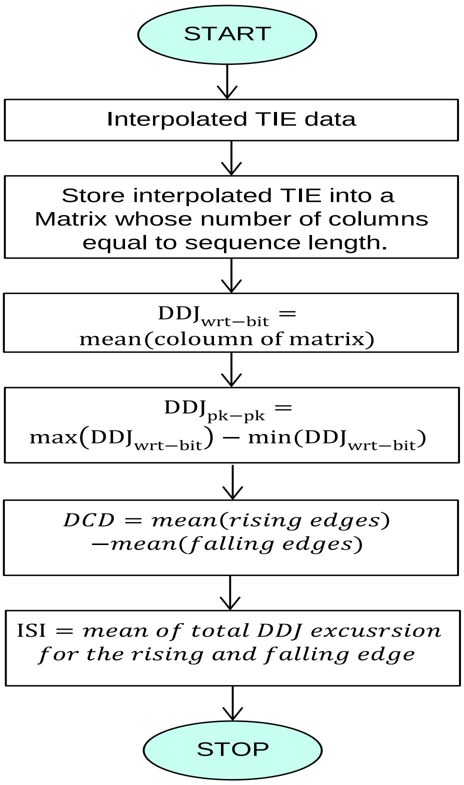

START
Interpolated TIE data
Store interpolated TIE into a Matrix whose number of columns equal to sequence length.
$\mathrm{DDJ}_{\mathrm{wrt-bit}} = \mathrm{mean(coloumn\ of\ matrix)}$
$\mathrm{DDJ}_{\mathrm{pk-pk}} = \max(\mathrm{DDJ}_{\mathrm{wrt-bit}}) - \min(\mathrm{DDJ}_{\mathrm{wrt-bit}})$
$DCD = mean(rising\ edges) - mean(falling\ edges)$
$\mathrm{ISI} = mean\ of\ total\ DDJ\ excusrsion\ for\ the\ rising\ and\ falling\ edge$
STOP

Fig.3.44 Flow Chart of ISI and DCD Extraction.

Extraction of DDJ from TIE

The obtained DDJ vector is than subtracted from TIE vector. This provides us with a vector that TIE values that have been caused only by PJ and RJ. The next step of this algorithm is estimation of PJ and RJ components. The procedure of PJ and RJ estimation is same as that we followed in previous algorithm for DJ and RJ separation (i.e. using FFT method)

Estimation of PJ

The algorithm for PJ and RJ estimation is shown in Fig.3.45. PJ can be obtained using either FFT or PSD. We use the FFT method. This method employs Fast Fourier Transform (FFT) to obtain the amount of periodic jitter. We use the interpolated TIE vector from which DDJ has been removed and find its FFT, spectrum is shown in Fig.3.46. Periodic jitter appears as spikes within the spectrum, as shown in Fig.3.47. If these spikes are extracted from the spectrum, we obtain the spectrum of PJ. The PJ spectrum can be transformed into time domain to obtain PJ (pk-pk) value.

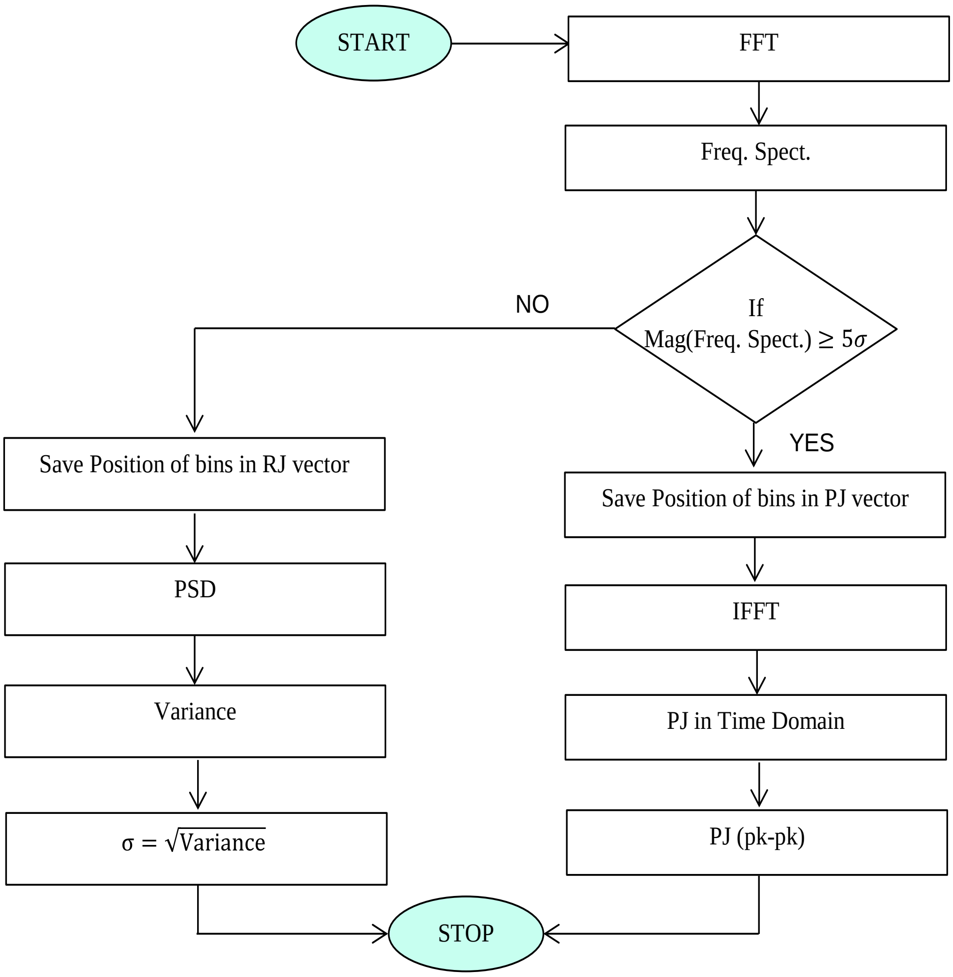


Fig.3.45 Flow Chart of PJ and RJ Extraction.

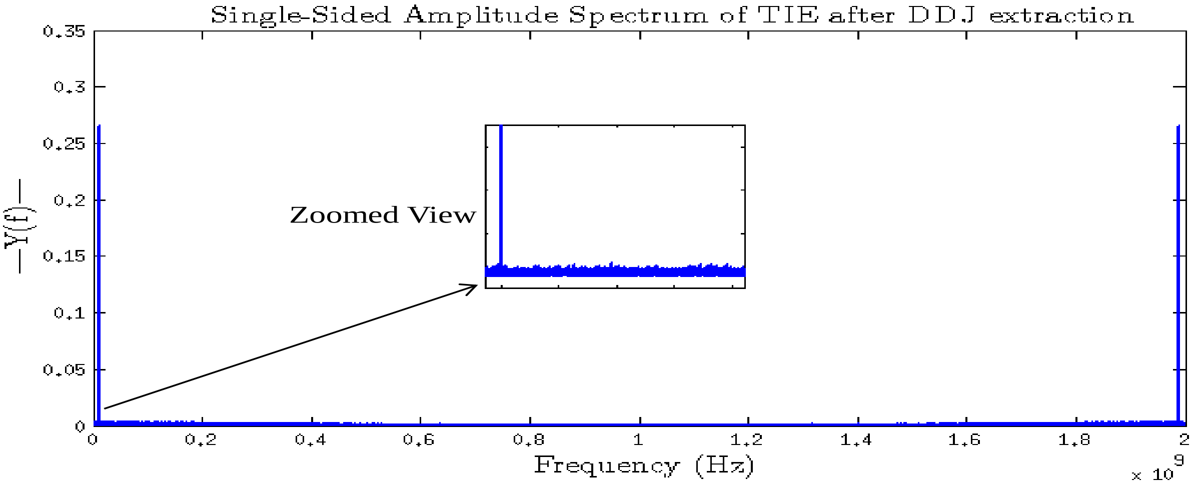


Fig.3.46 Frequency Spectrum after Removal of DDJ.

In Fig.3.47 shows PJ as spikes. These spikes can be extracted from the spectrum by finding the points where the spectrum has a value greater than N∗ σ. Here σ represents the standard deviation of the FFT coefficients and N is and an integer.

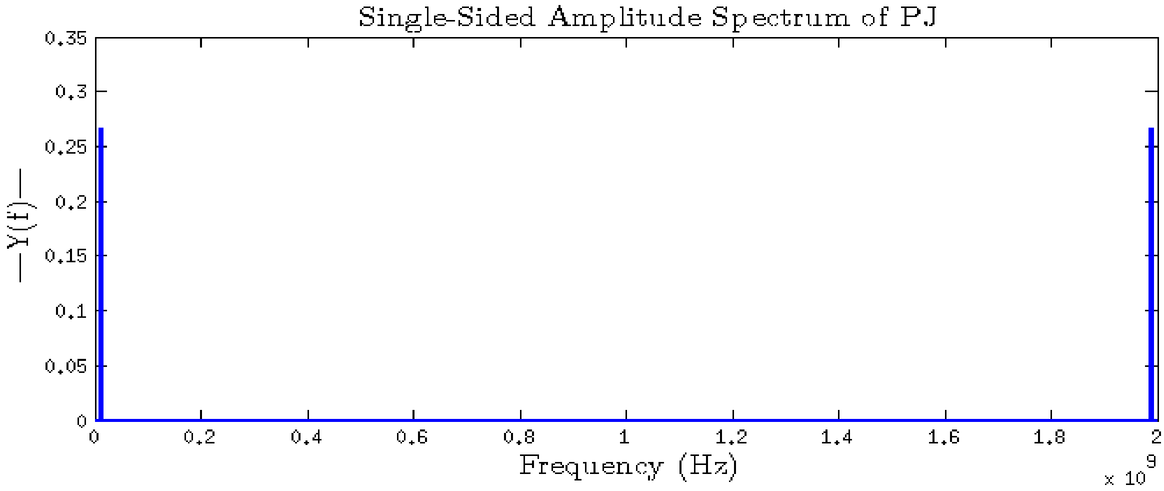


Fig.3.47 Frequency Spectrum of PJ from Frequency Domain Algorithm.

The spectrum is converted into time domain using Inverse Fourier Transform (IFFT). This can be accomplished using a built IFFT function or directly employing summation of sine and cosines.

Estimation of RJ

From the spectrum of interpolated TIE obtained earlier, we subtract the PJ spectrum. The positions from where the PJ spikes are extracted are interpolated. This step provides us with spectrum of random jitter. Fig.3.48 shows the spectrum of RJ.

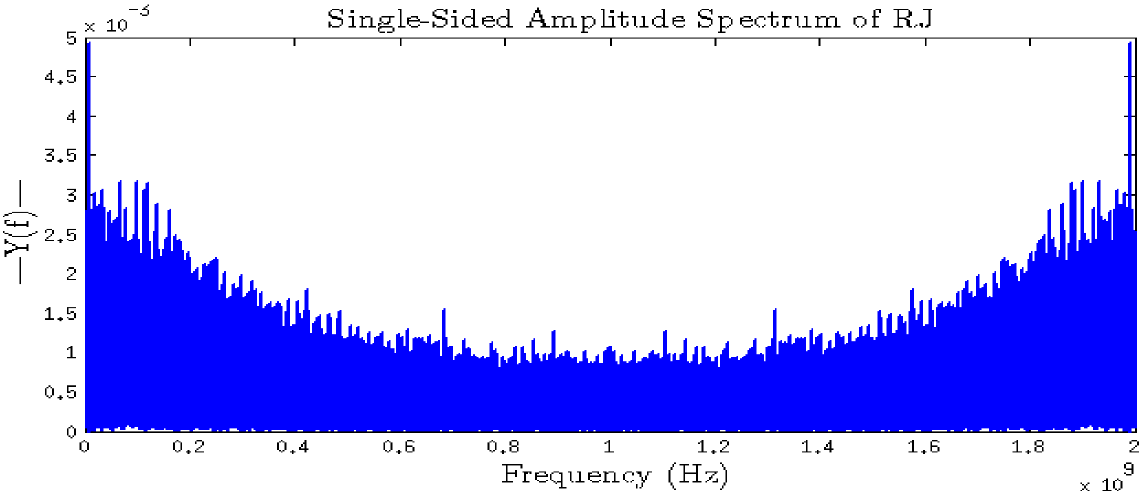


Fig.3.48 Frequency Spectrum of RJ from Frequency Domain Algorithm.

Finally, RMS value of RJ is obtained from PSD of RJ spectrum and then using Parseval’s theorem [38] [39].

$$\sum_{n=0}^{N-1} |x[n]|^2 = \frac{1}{N} \sum_{k=0}^{N-1} |X[k]|^2 \qquad (34)$$

Thus, if we take the square of all Fourier Coefficients in the RJ spectrum, we obtain the standard deviation of the random jitter.

Chapter 4 Results

In this chapter, results of jitter segregation algorithms are discussed. We considered 15 different sets of cases to validate all the above discussed algorithms. In these cases, different types of jitter components are present, to see the effect of one jitter component on the other. The input of the jitter generator is non-return to zero PRBS pattern having data rate is 2Gbps. The .mat file of the zero crossover points of jittered signals are loaded in the algorithms. These algorithms are implemented in MATLAB and results of these algorithms are extensively verified with Agilent ADS.

Table 1 Jitter Segregation using Tail-Fit Algorithm:

Case	Expected value of Jitter (ps)						Obtained Value (ps)		Error Percentage	
No.	RJ	PJ	ISI	DCD	DDJ	DJ	RJ	DJ	RJ	DJ
1	5.00	20.00	16.44	24.80	41.20	61.20	11.2	31.9	124%	-48%
2	5.00	0.00	0.00	0.00	0.00	00	4.76	1.17	-5%	NA
3	0.00	20.00	0.00	0.00	0.00	20	0.1	24.10	NA	20%
4	5.00	20.00	0.00	0.00	0.00	20	6.16	11.00	23%	-45%
5	0.00	0.00	0.00	24.80	24.80	24.80	1.79	12.40	NA	-50%
6	0.00	20.00	0.00	24.80	24.80	44.80	0.29	44.50	NA	-1%
7	5.00	0.00	0.00	24.80	24.80	24.8	5.02	24.90	0%	1%
8	5.00	20.00	0.00	24.80	24.80	44.80	5.71	37.40	14%	-16%
9	0.00	0.00	16.40	0.00	16.40	16.40	0.30	13.70	NA	-17%
10	0.00	0.00	16.40	24.80	41.20	41.20	2.49	7.99	NA	-81%
11	0.00	20.00	16.40	0.00	16.40	36.40	0.39	35.5	NA	-3%
12	5.00	0.00	16.40	0.00	16.40	16.40	5.01	16.6	0%	1%
13	5.00	0.00	16.40	24.80	41.20	41.20	5.00	43.8	0%	6%
14	5.00	20.00	16.40	0.00	16.40	36.40	10.3	6.56	105%	-82%
15	0.00	20.00	16.40	24.80	41.2	61.2	0.8	61.4	NA	05

The results of tail-fit algorithm are shown in Table 1 for different set of jitter component presents in the signal. From the Table 1, we have seen that a large amount of error is present in case 1 and in case 14. While precisely investigating these results, we conclude that due to presence of DDJ component in the signal, the tail fit algorithm gives large amount of error. In Fig.3.28 and in Fig.3.29, the explanation of presence of ISI in the signal is mentioned. To minimized or reduced these error percentage in tail-fit algorithm, extraction of DDJ components from TJ components before applying tail fit algorithm are advantageous.

Table 2 Jitter Segregation using Improved Tail Fit Algorithm

Case	Expected value of Jitter (ps)						Obtained Value (ps)		Error Percentage	
No.	RJ	PJ	ISI	DCD	DDJ	DJ	RJ	DJ	RJ	DJ
1	5.00	20.00	16.44	24.80	41.20	61.20	6.12	55.60	22%	-9%
2	5.00	0.00	0.00	0.00	0.00	00	4.81	1.71	-4%	NA
3	0.00	20.00	0.00	0.00	0.00	20	0.2	19.70	NA	-1%
4	5.00	20.00	0.00	0.00	0.00	20	6.04	12.20	21%	-39%
5	0.00	0.00	0.00	24.80	24.80	24.80	0.94	24.80	NA	0%
6	0.00	20.00	0.00	24.80	24.80	44.80	0.26	44.5	NA	-1%
7	5.00	0.00	0.00	24.80	24.80	24.8	4.81	26.4	-4%	7%
8	5.00	20.00	0.00	24.80	24.80	44.80	6.05	38.80	21%	-18%
9	0.00	0.00	16.40	0.00	16.40	16.40	$2*10^{-8}$	16.40	NA	0%
10	0.00	0.00	16.40	24.80	41.20	41.20	$2*10^{-8}$	43.50	NA	6%
11	0.00	20.00	16.40	0.00	16.40	36.40	0.49	36.00	NA	-1%
12	5.00	0.00	16.40	0.00	16.40	16.40	4.94	18.80	-1%	14%
13	5.00	0.00	16.40	24.80	41.20	41.20	5.03	45.30	0.4%	10%
14	5.00	20.00	16.40	0.00	16.40	36.40	6.19	28.30	24%	-22%
15	0.00	20.00	16.40	24.80	41.2	61.2	0.33	67.60	NA	10%

The results shown in Table 2 are better than the original tail-fit algorithm. But the error percentage is still high as seen in case 1, 4, 8 and in case 14. This is because of presence of PJ in the signal. The reason for this explained in the section 3.2.1.1 and shown in Fig.3.32, Fig.3.33 and inFig.3.34.

In the absence of both PJ and ISI in the signal, tail-fit algorithm gives best results, shown in Fig.3.28. Further, to minimize this high percentage of error and to get accurate estimation of jitter components, frequency domains algorithms are better.

In Table 3, the results of jitter estimation using clock pattern and percentage error are shown in Table 4 . The results are better than tail-fit algorithm and improved tail-fit algorithm. The error percentage is below 10 % except in case 12, shown in Table 4 . In this algorithm, the calculation of ISI is dependent on DJ component and DJ component is calculated by spectrum approach. Therefore, due to leakage of RJ component while extraction of DJ from TJ, estimation of DJ value not accurate as we have to predict. Another drawback of this algorithm is need of extra clock that cause more power consumption.

Table 3 Jitter Segregation using Clock pattern

Case	Expected value of Jitter (ps)						Obtained Value of Jitter (ps)					
No.	RJ	PJ	ISI	DCD	DDJ	DJ	RJ	PJ	ISI	DCD	DDJ	DJ
1	5.00	20.00	16.44	24.80	41.20	61.20	5.46	22.50	16.44	27.00	43.40	65.90
2	5.00	0.00	0.00	0.00	0.00	00	5.01	00	0.52	0.02	0.5	0.5
3	0.00	20.00	0.00	0.00	0.00	20	0.75	22.30	2.21	$1.9*10^{-6}$	2.21	24.5
4	5.00	20.00	0.00	0.00	0.00	20	5.04	22.70	2.80	0.02	2.82	25.60
5	0.00	0.00	0.00	24.80	24.80	24.80	$1.1*10^{-9}$	$1.2*10^{-7}$	1.34	24.70	26.10	26.10
6	0.00	20.00	0.00	24.80	24.80	44.80	0.5	22.00	0.83	24.70	25.60	22.00
7	5.00	0.00	0.00	24.80	24.80	24.8	5.01	00	1.98	24.70	26.70	26.70
8	5.00	20.00	0.00	24.80	24.80	44.80	5.04	22.70	1.8	24.70	26.40	49.20
9	0.00	0.00	16.40	0.00	16.40	16.40	$1.3*10^{-8}$	$1.1*10^{-7}$	18.11	$4.4*10^{-4}$	18.11	18.11
10	0.00	0.00	16.40	24.80	41.20	41.20	$1.3*10^{-8}$	$2.1*10^{-7}$	17.50	26.90	44.40	44.40
11	0.00	20.00	16.40	0.00	16.40	36.40	0.50	21.90	17.10	$4.6*10^{-4}$	17.22	39.00
12	5.00	0.00	16.40	0.00	16.40	16.40	5.49	0.00	18.90	0.02	18.92	18.92
13	5.00	0.00	16.40	24.80	41.20	41.20	5.45	0.00	17.30	27.00	44.20	44.20
14	5.00	20.00	16.40	0.00	16.40	36.40	5.46	22.50	16.70	$2.7*10^{-2}$	16.70	39.30
15	0.00	20.00	16.40	24.80	41.2	61.2	5.94	22.20	15.00	27.00	42.0	64.30

Table 4 Error percentage in Jitter Segregation using Clock Pattern.

Case	Error Percentage					
No.	RJ	PJ	ISI	DCD	DDJ	DJ
1	9.20%	13%	-0.24%	8.93%	5.27%	7.66%
2	0.12%	NA	NA	NA	NA	NA
3	NA	11%	NA	NA	NA	22.35%
4	0.8%	14%	NA	NA	NA	27.77%
5	NA	NA	NA	0.00%	5.41%	5.41%
6	NA	11%	NA	0.00%	3.37%	6.91%
7	0.12%	NA	NA	0.00%	7.98%	7.98%
8	0.80%	14%	NA	0.00%	6.79%	9.85%
9	NA	NA	10.10%	NA	10.10%	10.10%
10	NA	NA	6.57%	8.77%	7.89%	7.89%
11	NA	9%	4.20%	NA	4.20%	7.08%
12	9.80%	NA	14.96%	NA	15.13%	15.13%
13	9.00%	NA	5.05%	8.89%	7.36%	7.36%
14	9.20%	13%	1.64%	NA	1.81%	7.73%
15	NA	11%	-8.58%	9.05%	2.02%	5.00%

The results of frequency domain algorithm are shown in Table 5 and Table 6. The procedure estimation of jitter components using frequency domain algorithm is somewhat similar to jitter components using clock pattern, as discussed in Chapter 3. The error percentage is below 10%. The problem in frequency domain algorithm that it are more prone to errors when the presence of high sinusoidal jitters in the signal.

Table 5 Jitter Segregation using frequency domain algorithm.

Case	Expected value of Jitter (ps)						Obtained Value of Jitter (ps)					
No.	RJ	PJ	ISI	DCD	DDJ	DJ	RJ	PJ	ISI	DCD	DDJ	DJ
1	5.00	20.00	16.44	24.80	41.20	61.20	4.62	21.1	17.20	27.0	44.10	65.30
2	5.00	0.00	0.00	0.00	0.00	00	4.42	0.36	0.74	0.02	0.76	1.13
3	0.00	20.00	0.00	0.00	0.00	20	0.5	22.20	0.01	$1.9*10^{-6}$	0.01	22.30
4	5.00	20.00	0.00	0.00	0.00	20	4.46	21.11	0.74	0.02	0.76	21.90
5	0.00	0.00	0.00	24.80	24.80	24.80	$1.1*10^{-8}$	$7.1*10^{-8}$	$2*10^{-9}$	24.70	24.70	24.70
6	0.00	20.00	0.00	24.80	24.80	44.80	0.5	22.20	0.001	24.80	24.80	47.00
7	5.00	0.00	0.00	24.80	24.80	24.8	4.42	0.36	0.07	24.70	25.50	25.80
8	5.00	20.00	0.00	24.80	24.80	44.80	4.46	21.11	0.07	24.70	25.50	46.10
9	0.00	0.00	16.40	0.00	16.40	16.40	$1.3*10^{-8}$	$1.8*10^{-7}$	16.40	$4.4*10^{-4}$	16.40	16.50
10	0.00	0.00	16.40	24.80	41.20	41.20	$1.4*10^{-8}$	$1.8*10^{-7}$	16.70	26.90	43.60	43.60
11	0.00	20.00	16.40	0.00	16.40	36.40	0.5	22.20	16.50	$4.6*10^{-4}$	16.50	38.7
12	5.00	0.00	16.40	0.00	16.40	16.40	4.61	0.36	17.11	0.02	17.10	17.49
13	5.00	0.00	16.40	24.80	41.20	41.20	4.60	0.36	17.20	27.00	44.10	44.50
14	5.00	20.00	16.40	0.00	16.40	36.40	4.62	21.10	17.00	27.40	17.11	38.20
15	0.00	20.00	16.40	24.80	41.2	61.2	0.05	22.20	16.60	27.00	43.5	65.80

Table 6 Error Percentage in Jitter Segregation using frequency domain algorithm.

Case	Error Percentage					
No.	RJ	PJ	ISI	DCD	DDJ	DJ
1	-8.00%	6.00%	4.00%	9.00%	7.00%	7.00%
2	-12.00%	NA	NA	NA	NA	NA
3	NA	11.00%	NA	NA	NA	11.00%
4	-11.00%	6.00%	NA	NA	NA	9.00%
5	NA	NA	NA	0.00%	0.00%	0.00%
6	NA	11.00%	NA	0.00%	0.00%	5.00%
7	-12.00%	NA	NA	0.00%	3.00%	4.00%
8	-11.00%	6.00%	NA	0.00%	3.00%	4.00%
9	NA	NA	0.00%	NA	0.00%	0.00%

10	NA	NA	1.00%	9.00%	6.00%	6.00%
11	NA	11.00%	0.00%	NA	0.00%	6.00%
12	-8.00%	NA	4.00%	NA	4.00%	6.00%
13	-8.00%	NA	5.00%	9.00%	7.00%	8.00%
14	-8.00%	6.00%	4.00%	NA	4.00%	5.00%
15	NA	11.00%	1.00%	9.00%	6.00%	7.00%

The results of the above algorithm are extensively verified with Agilent ADS. The same amount of jitter components presents in signal are segregated with the help of ADS tool. The results of this are shown in Table 7 and the percentage errors are shown in Table 8. The error percentage is below 15% from ADS tool. Therefore, the results of jitter segregation using clock pattern technique and using frequency domain algorithm are almost same ad Agilent ADS. Hence, the validation of accuracy of our designed approach is proved.

Table 7 Jitter Segregation using Agilent ADS.

Case	Expected value of Jitter (ps)						Obtained Value of Jitter (ps)					
No.	RJ	PJ	ISI	DCD	DDJ	DJ	RJ	PJ	ISI	DCD	DDJ	DJ
1	5.00	20.00	16.44	24.80	41.20	61.20	5.23	16.30	18.90	27.00	45.9	62.2
2	5.00	0.00	0.00	0.00	0.00	00	4.99	1.88	0.21	0.002	0.24	2.12
3	0.00	20.00	0.00	0.00	0.00	20	0.3	17.90	0.007	$8*10^{-5}$	0.007	17.9
4	5.00	20.00	0.00	0.00	0.00	20	4.99	16.50	0.2	0.002	0.2	16.8
5	0.00	0.00	0.00	24.80	24.80	24.80	0.01	0.00	0.00	24.80	24.8	24.8
6	0.00	20.00	0.00	24.80	24.80	44.80	0.03	17.90	0.007	24.80	24.8	42.7
7	5.00	0.00	0.00	24.80	24.80	24.8	4.99	1.86	0.2	24.70	24.9	26.8
8	5.00	20.00	0.00	24.80	24.80	44.80	4.98	16.60	0.2	24.70	25.00	41.6
9	0.00	0.00	16.40	0.00	16.40	16.40	0.01	0.00	16.40	$5.5*10^{-4}$	16.40	16.40
10	0.00	0.00	16.40	24.80	41.20	41.20	0.01	0.00	18.80	26.9	45.70	45.70
11	0.00	20.00	16.40	0.00	16.40	36.40	0.08	19.50	16.50	$6.5*10^{-4}$	16.50	36.00
12	5.00	0.00	16.40	0.00	16.40	16.40	5.25	1.85	16.60	0.02	16.60	18.50
13	5.00	0.00	16.40	24.80	41.20	41.20	5.23	1.87	18.90	27.0	45.90	47.70
14	5.00	20.00	16.40	0.00	16.40	36.40	5.20	16.3	16.66	0.02	16.60	32.90
15	0.00	20.00	16.40	24.80	41.2	61.2	0.09	19.6	18.70	27.00	45.70	65.30

Table 8 Error Segregation in Jitter Estimation using Agilent ADS.

Case No.	Error Percentage					
	RJ	PJ	ISI	DCD	DDJ	DJ
1	5.00%	-18.0%	15.00%	9.00%	11.00%	2.00%
2	0.00%	NA	NA	NA	NA	NA
3	NA	-11.00%	NA	NA	NA	-11.00%
4	0.00%	-17.00%	NA	NA	NA	-16.00%
5	NA	NA	NA	0.00%	0.00%	0.00%
6	NA	-10.00%	NA	0.00%	0.00%	-5.00%
7	0.00%	NA	NA	0.00%	1.00%	8.00%
8	0.00%	-17.00%	NA	0.00%	1.00%	-7.00%
9	NA	NA	0.00%	NA	0.00%	0.00%
10	NA	NA	14.00%	9.00%	11.00%	11.00%
11	NA	3.00%	0.00%	NA	0.00%	-1.00%
12	5.00%	NA	1.00%	NA	1.00%	12.00%
13	5.00%	NA	15.00%	9.00%	11.00%	16.00%
14	4.00%	-19.00%	1.00%	NA	1.00%	10.00%
15	NA	2.00%	14.00%	9.00%	11.00%	7.00%

4.1 Summary

1. The tail fit algorithm works best only when DJ has dual Dirac probability distribution function. Another drawback of tail fit algorithm is that it cannot estimate all jitter components. These problems can be solved in jitter estimation using clock pattern and by spectrum analysis.

2. The advantage of tail-fit algorithm and our designed technique for jitter segregation is that there is no need to know the pattern length in advance while in the improved tail fit algorithm and in spectrum approach technique, pattern length needs to be known in advance.

3. The error between expected DJ and obtained DJ is relatively low in jitter segregation using clock pattern and using spectrum approach techniques. However, these are more prone to errors in case of high frequency sinusoidal jitter and when the edge transition frequency is not sufficient to satisfy the Nyquist criteria.

4. Another assumption while using frequency domain approach and in clock pattern technique is that the number of samples is sufficient to ensure that at least one cycle of the slowest jitter is accounted for.

5. On the other hand, these limitations should not affect much in case of tail-fitting algorithm because these are based on statistics of data. The summary of all jitter segregation algorithms are shown in Table 9.

Table 9 Summary of Algorithms

Algorithms **Parameters**	**Tail-fit**	**Improved Tail-fit**	**Spectrum Analysis**	**Jitter estimation using Clock Pattern**
% of Error	< 30%	< 20 %	> 10%	10-15%
Input Pattern	PRBS	PRBS	PRBS	Clock Pattern, PRBS
Sequence Period	No	Yes	Yes	NO
Jitter Component	RJ and DJ	RJ, DJ and DDJ	RJ, DCD, ISI and PJ	RJ, DCD, ISI, PJ
Identification of jitter source	No	No	Yes	Yes

Chapter 5 Conclusion & Future Work

This report provides the basic understanding of jitter fundamentals and flow charts of the different jitter segregation algorithms such as tail-fit algorithm, improved tail-fit algorithm, jitter segregation using clock pattern and jitter estimation using spectrum approach. Meanwhile, we developed jitter generator module using their mathematical model in MATLAB. Further, the strength and limitation of these algorithms are discussed.

This thesis provides the accuracy of RJ can be affected by a large amount in case DJ does not have a Dual Dirac distribution. Several cases are discussed with different combination of jitter components for above mentioned algorithms. The error percentage is also divergent for different algorithms in all cases. The results of these algorithms are validated from Agilent ADS. The error percentage in jitter segregation using clock pattern and using spectrum method is equivalent to Agilent ADS results.

Future Work

The effect of jitter on the signals increases as we move towards the high speed serial links. The effect of Bounded Uncorrelated Jitter (BUJ) is more on the signal as frequency and data rates and interconnect speeds are continuously increasing. The main cause of BUJ is crosstalk due to density and cost concern. Another aspect can develop a more accurate method for precise value for all jitter components that includes DJ, RJ as well BUJ. In addition to these, look for the reason for errors in the frequency domain algorithm.

REFERENCES

[1] http://www.sims.berkeley.edu/research/projects/how-much-info/execsum.htm.

[2] http://www.internetlivestats.com/internet-users/

[3] Buckwalter, James Franklin, "Deterministic jitter in broadband communication", Diss. California Institute of Technology, 2006.

[4] "Desktop Processor," Intel Corporation, [Online]. Available: http://ark.intel.com/

[5] R. Stephens, B. Fetz, S. Draving, J. Evangelista, M. F. Reumann, G. LeCheminant, J. Stimple, "Comparison of Different Jitter Analysis Techniques With a Precision Jitter Transmitter",White Paper, DesignCon, February 2, 2005.

[6] Johnson, Howard W., and Martin Graham, "High-speed digital design: a handbook of black magic", Vol. 1, Upper Saddle River, NJ: Prentice Hall, 1993.

[7] Li, Mike Peng, "Jitter, noise, and signal integrity at high-speed", Pearson Education, 2007.

[8] Thierauf, Stephen C., "Understanding Signal Integrity", Artech House, 2011.

[9] Tektronix Inc, "Fundamentals of Signal Integrity", Primer, 2008. [Online]. Available: www.tektronix.com/signal_integrity.

[10] J. S.-A. U. R. Andreas Cangellaris, "Interactive Tutorial on Fundamentals of Signal Integrity for High-Speed/High-Density Design," in *DAC*, 2001

[11] Wavecrest, "An Introduction to Jitter Analysis," 1 Feb 2000. [Online]. Available: class.ee.iastate.edu/djchen/ee507/intro_to_jitter%20_analysis.ppt.

[12] N. Soo, "Jitter Measurement Techniques," Pericom, 30 November 2000. [Online]. Available: http://www.pericom.com/assets/App-Note-Files/AB036.pdf.

[13] S. Corporation, "Clock Jitter Definitions and Measurements Methods," January 2014.. [Online]. Available: http://www.sitime.com/support2/documents/AN10007-Jitter-and-measurement.pdf.

[14] SEIKO EPSON CORPORATION, "Jitter and Phase Noise-The Fundamentals", Technical Notes. [Online]. Available: http://www5.epsondevice.com/en/quartz/library/tech_note/techl_notes_jitter0416.pdf.

[15] R. Stephens, "All About the Acronyms: RJ, DJ, DDJ, ISI, DCD, PJ, SJ," Tektronix, 2006. [Online]. Available: www.ece.ncsu.edu/asic/ece733/2009/docs/Jitter_Tektronics.pdf.

[16] National Instruments, "Understanding and Characterizing Timing Jitter", White Paper, Apr 17, 2013. [Online]. Available: http://www.ni.com/white-paper/14227/en/.

[17] C. Emmerich, "Introduction to Jitter", Wavecrest [Online]. Available: http://class.ee.iastate.edu/mmina/ee418/Notes/Introduction_to_Jitter.pdf.

[18] N. Radhakrishnan, "Stressed Eye Analysis and Jitter Separation for High Speed Serial Links," Missouri University of science and technology, 2009.

[19] J. Hancock, "Jitter: understanding It, Measuring It, Eliminating It Part I, II, III: Jitter Fundamentals," in Agilent Technologies, High Frequency Electronics, Summit Technical Media, LLC, 2004.

[20] ON Semiconductor, "Understanding Data Eye Diagram Methodology for Analyzing High Speed Digital Signals", Application Note. [Online]. Available: http://onsemi.com

[21] K. Kachigan, "Measuring Jitter – The New Digital Design Challenge", RTC , 6 July 2005.

[22] Mike Li, "Deterministic Jitter (DJ) Defination and Measurement Methods: An Old Problem Revisted", IEEE P802.3ba, 40G/100G Meeting, January, 2009, New Orleans.

[23] Martin Miller, "Six Tales of RJ and DJ or A Discussion of Which Measurement Approach is Best Adapted to Specific Needs", 2005. [Online]. Available: www.lecroy.com.

[24] Buckwalter, J. et al, "Predicting Data-Dependent Jitter", IEEE Transactions on Circuits and Systems, Vol. 51, No. 9, Sep. 2004, pp. 453-457.

[25] E. Song, J. Kim et al, "Data-dependent jitter estimation using single pulse analysis method", Electronic packaging Technology Conference, 2005.

[26] E. Song, J. Kim et al, "Data-dependent jitter estimation using single pulse analysis method in High-Speed Differential Signaling", Electronics Systemintegration Technology Conference, 2006.

[27] Dey, Aritra, and Hong Jiang Song. "Modeling of ISI in high speed serial I/Os terminated with discontinuities." Electrical Performance of Electronic Packaging and Systems (EPEPS), 2011 IEEE 20th Conference on. IEEE, 2011.

[28] K. K. Kim, J. Huang, Y. B. kim, F. Lombardi, "On the Modeling and Analysis of Jitter in ATE Using Matlab", Defect and Fault Tolerance in VLSI Systems, 2005. Pp. 285 – 293.

[29] K. K. Kim, Y. B. kim, F. Lombardi, “Data Dependent Jitter Characterization Methodologies”, Defect and Fault Tolerance in VLSI Systems, 2005. Pp. 294 – 302.

[30] Mike P. Li, Jan Wilstrup, Ross Jessen, Dennis Petrich, “A New Method for Jitter Decomposition Through Its Distribution Tail Fitting”, International Test Conference, 1999 , pp. 788 – 794.

[31] Jie Sun, mike Li, and Jan Wilstrup, “A Demonstration of Deterministic Jitter (DJ) Deconvolution”, IEEE Instrumentation and Measurement Technology Conference, May 2002.

[32] F. Nan, Y. Wang, F. Li, W. Yang, X. Ma, “A Better method than Tail-fitting Algorithm for Jitter Separation Based on Gaussian Mixture Model”, Journal of Electronic Testing, December 2009, Volume 25, Issue 6, pp. 337-342.

[33] Srinath MD, Rajasekaran PK, Viswanathan R, “Introduction to statistical Signal Processing with Applications”, Prentice Hall, Englewood Cliffs, 1996.

[34] N. K. Chhabra, K. bhatheja, J. N. Tripathi, R. Nagpal, R. Malik, “Mitigating the Impact of Sinusoidal Jitter and Duty Cycle Distortion on Random Jitter Estimation by Tailfit Algorithm”, Electrical Performance of Electronic Packaging and Systems (EPEPS), 2013 IEEE 22nd Conference, 2013, pp. 151-154.

[35] Masashi Shimanouchi, et. “New Paradigm for Signal Paths in ATE Pin Electronics are Needed for Serialcom Device Testing”, IEEE International Test Conferecne, 2002, pp. 903-912.

[36] C. K. Ong, D. Hong, K. T. Cheng, Li-C Wang, “Jitter Spectral Extraction for Multi-gigahertz Signal”, Design Automation Conference, 2004, pp. 298-303.

[37] Denis Donnelly and Bert Rust, “The Fast Fourier Transform for Experimentalists, Part I: Concepts”, Computing in Science & Engineering, 2005, pp. 80-88.

[38] Tecktronix Inc., “Analyzing Jitter Using a Spectrum Approach”, Application Note, 2002.

[39] C. K. Ong, D. Hong, K. T. Cheng, Li-C Wang, “Random Jitter Extraction Technique in a Multi-gigahertz Signal”, Design Automation Conference, 2004, pp. 286-291.

[40] Alan V. Oppenheim and Ronald W. Schafer, “Discrete-Time Signal Processing ” 2nd Edition Prentice Hall Signal Processing Series, 1999.

[41] P. R. Belington, D. K. Robinson, “Data reduction and error analysisfor the physical science”, McGraw-Hill, Inc, 1992.