FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Jitter Analysis

Luis Cruz

PROVISIONAL VERSION

Frebuary 2011

© Luis Cruz, 2011

Contents

List of Figures

List of Tables

Abbreviations and Symbols

BER	Bit Error Ratio (Number of errors divided by the number of transmitted bits)
BERT	Bit Error Ratio Tester
CAD	Computer-Aided Design
CASE	Computer-Aided Software Engineering
CDR	Clock and Data Recovery
DDJ	Data Dependent Jitter
DJ	Deterministic Jitter
ISI	Inter-symbol interference
PCS	Phase Changing Speed
PDF	Probability Density Function
PLL	Phased-looked Loop
RMS	Root Mean Square
RJ	Random Jitter
SerDes	Serializer/Deserializer
TIE	Time Error Interval
TJ	Total Jitter

Chapter 1

Fundamentals of Jitter

1.1 Jitter definition

Rapidly demand for high speed interface standards raised considerable signal integrity issues. Designers now have to ensure correct functionality and signal integrity, resulting on correct data transmission at bit rates higher than 1Gb/s. At that frequencies digital signal behavior is more related with analog signals, the logical ones and zeros aren't a "1s" or "0s", it will have voltage and timing variations that can lead to wrong signal recovery. Timing variations are called as jitter (wander for frequencies bellow 10Hz).

Jitter is defined as the deviation of the digital timing event from it's ideal timing event. Such deviation can be represented as a probability density function(PDF) histogram, since jitter by definition is a time measurement. Jitter histogram can be obtained by the quantization of the measured deviations, allowing the user to identify specific types of jitter.

There are different jitter measurements, each measurement will return a different value.The must common are:

- **Cycle to cycle jitter -** Applied to clocks; measures the difference between the current clock period and the precious one. It can also be applied to data signals with the necessary improvements. This measurement isn't used on current designs because a low value doesn't mean that the system is working properly.

- **Period jitter -** Applied to clock signals; it represents the maximum deviation of the real clock transition point to the ideal one.

- **Long term jitter -** Applied to clock signals; it represents the difference between the ideal clock transition and the real one, this measure is performed over a large number of clock cycles and can be represented on a histogram form.

- **Time interval error jitter -** Can be applied to data and clock signals; it represents the difference between the event of the signal being measured and the ideal recovered clock event. This measurement is performed over a large number of bits and can be represented on a histogram form.

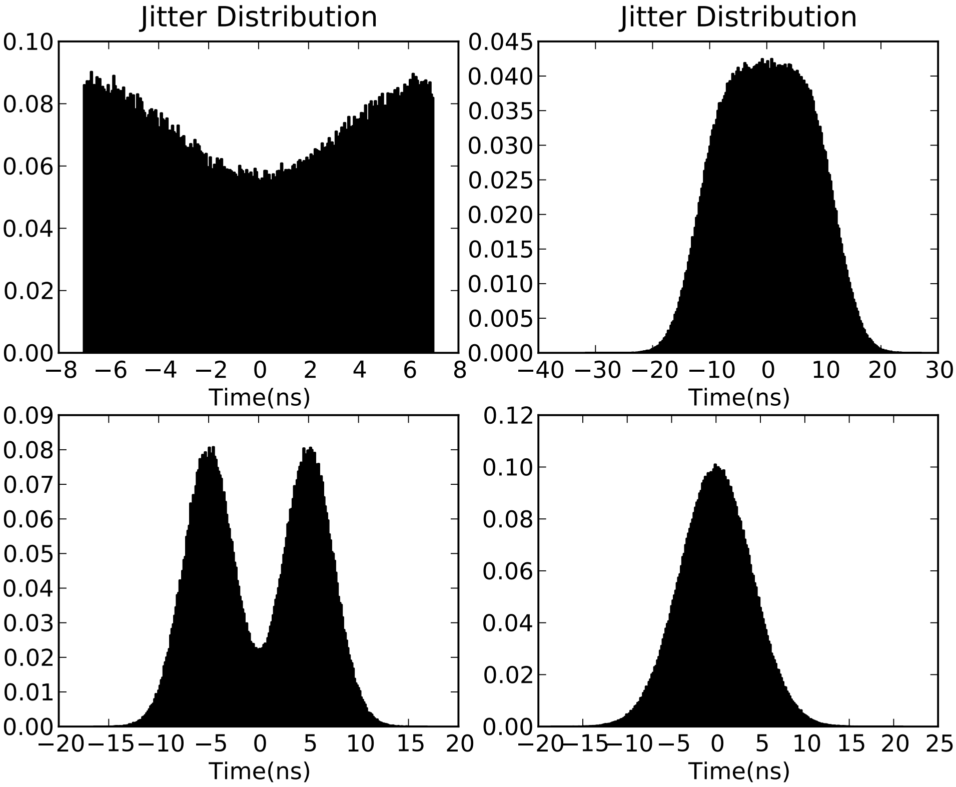


Figure 1.1: Jitter Distribution examples

Figure 1.1 on page 2 shows different types of jitter distributions, each distribution was obtained from different jitter sources. Through observations it was clear that the total jitter is data dependent, it increases with the number of captured events and is pattern dependent. Such observations allowed the designers to conclude that jitter isn't bounded. In fact jitter is a combination of phenomenons divided in two major groups: deterministic and random.

During many years the effect of random jitter (RJ) on the overall total jitter (TJ) was negligible due to the low data rates involved, on this cases deterministic factors like periodic jitter dictate the total jitter value. For this reason today total jitter is defined as RMS or peak-to-peak in some specifications, such definitions aren't accurate because the total jitter isn't bounded as such definitions suggest. To correctly define the total jitter new methods should be used. New models combine the effects of deterministic jitter and random jitter allowing the definition of a total jitter value for a given bit error ratio. $TJ = DJ + N \times RJ$ where N = number of standard deviations corresponding to the required BER.

Total jitter can be divided into the following components [1](see Figure 1.2):

- **Deterministic jitter -** Jitter with non-Gaussian probability density function. It is always bounded in amplitude. Possible causes are imperfections of devices, EMI, grounding problems.

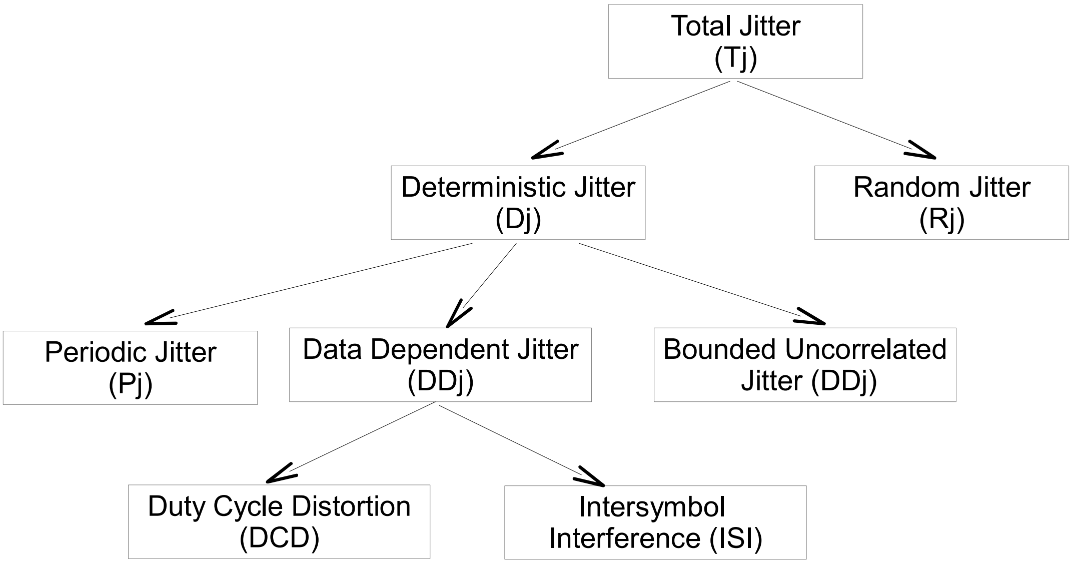

Total Jitter (Tj)
Deterministic Jitter (Dj)
Random Jitter (Rj)
Periodic Jitter (Pj)
Data Dependent Jitter (DDj)
Bounded Uncorrelated Jitter (DDj)
Duty Cycle Distortion (DCD)
Intersymbol Interference (ISI)

Figure 1.2: Jitter Subcomponents

- **Periodic jitter -** Refers to periodic variations of signal edge positions over time. Possible causes of PJ are electromagnetic interference sources such as power supplies.
- **Data dependent jitter -**Corresponds to a variable jitter that depends on the bit pattern transmitted on the link under test.
    - **Duty cycle distortion -**Refers to the bit period variation of consecutive alternated patterns (101010). Possible cause of DDJ is the difference between rise and fall times on the driver buffer stage.
    - **Inter-symbol interference -**Refers to bit time relation with the previous transmitted pattern.Possible cause of ISI is the low pass frequency response of the link under test.
- **Bounded uncorrelated jitter -**Refers to the bit time influence from adjacent links, the value is bounded but not correlated with the transmitted bits. Possible cause of BUJ is crosstalk.

- **Random jitter -** The principal source is Gaussian (white) noise within system components. It interacts with the slew rate of signals and produces timing errors at the switching points.

Decomposition of total jitter into subgroups allows a more accurate jitter budget definition, helping to control the different jitter sources. On today SerDes (Serializer/Deserializer) interfaces the SER part can be resumed into two blocks, PLL and Driver, DES part can be resumed to a PLL and a CDR (clock and data recovery). Each different block will be affect by a different jitter component, the specification of a jitter budget on subcomponents like: deterministic jitter and random jitter, allows the designers to accurately design each SERDES block.

1.2 Jitter Components

1.2.1 Periodic jitter

A fundamental limitation in high-speed digital communication systems is the jitter of phased-locked loops (PLL) [2], typically such jitter is on the form of periodic jitter. The periodic jitter added by PLLs must be tolerated by the clock and data recovery circuit (current standards define a sinusoidal jitter tolerance mask). Clock signal is a square wave with a fundamental frequency of f_s, such signal can be decomposed by a Fourier analysis into a sum of sine harmonics of frequencies f_s, $3f_s$, $5f_s$, etc. When this signal is passed through a bandpass filter (driver, channel) with a center frequency at f_s , the components outside the fundamental ($3f_s$, $5f_s$, $7f_s$, etc.) can be discarded. The square wave can be written as (fundamental sinusoidal component with amplitude A and frequency f_s):

$$Acos(j(t)) = Acos(2*\pi*f_s*t+\emptyset_0+\Delta\emptyset(t)) \quad (1.1)$$

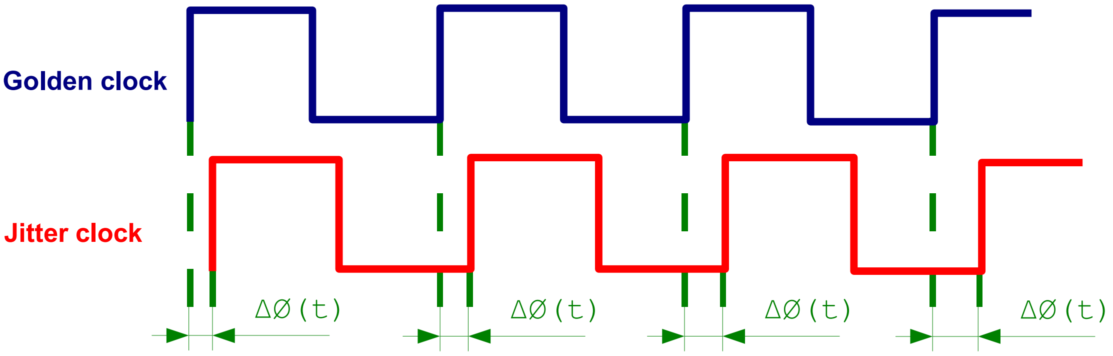


Figure 1.3: Sinusoidal Timing Jitter

The phase modulation component $\Delta\emptyset(t)$ (see figure 1.3), which is the timing jitter can be defined as: $\triangle\emptyset(t) = \Delta w*sin(w_j*t)$. Assuming $A_j*Tbit\ \ (s)$ as the maximum phase difference between golden clock and jitter clock, $\triangle\emptyset(t)$ can be written as:

$$\triangle\emptyset(t) = A_j*2*\pi*sin(w_j*t)\ \ (rad) \quad (1.2)$$

Please note that to ensure correct correlation between equations 1.1 and 1.2, the maximum phase jitter was converted from seconds to radians, since Tbit corresponds to $2*\pi$ radians, A_j*Tbit in seconds will correspond to $A_j*2*\pi$ in radians. The jitter frequency ($w(t)$) can be obtained by differentiating $j(t)$ [3] in relation to time as

$$w(t) = w_s+A_j*2*\pi*w_j*cos(w_j*t)\ \ (rad) \quad (1.3)$$

The maximum and minimum period can be defined ($T = 2*\pi/w$) as:
$T = \frac{2\pi}{ws\pm A_j*2*\pi*w_j}\ \ (s)$

$T = \frac{2\pi}{2*\pi*f_s \pm A_j*2*\pi*2*\pi*f_j} \ (s)$

$$T = \frac{1}{f_s(1 \pm A_j*2*\pi*\frac{f_j}{f_s})} \ (s) \tag{1.4}$$

Maximum and minimum frequency can be defined as: $f = f_s(1 \pm A_j*2*\pi*\frac{f_j}{f_s}) \ (Hz)$.

Another important measure is the phase changing speed (PCS), this parameter defines the minimum phase tracking on the CDR side. PCS can be obtained by differentiating $\Delta\emptyset(t)$

$$PCS = A_j*2*\pi*w_j*cos(w_j*t) \ (rad/s) \tag{1.5}$$

Equation (1.5) can be converted to the form of UI/UI (maximum phase change in terms of bit time per bit time). To convert seconds in UI (unit interval) it's necessary to multiple the seconds value by f_s, because $1 \ (UI) = 1/f_s \ (s)$. The conversion from rad to UI can be done by dividing the radian value by $2*\pi$. The maximum phase changing speed (MPCS),that must be tracked by the CDR, can be defined as:

$$MPCS = A_j*2*\pi*\frac{f_j}{f_s} \ (UI/UI) \tag{1.6}$$

Figure 1.4 was obtained trough simulation. Two clocks were created following the equations defined above. Golden clock was created using a pure cosine wave with a frequency of $f_s =$ 207MHz, jitter clock was created assuming a sinusoidal jitter with $f_s =$ 207Mhz, $f_j = 10MHz$, $A_j = 0.3Tbit$. The obtained experimental results are correlated with the previous equations (see table 1.1).

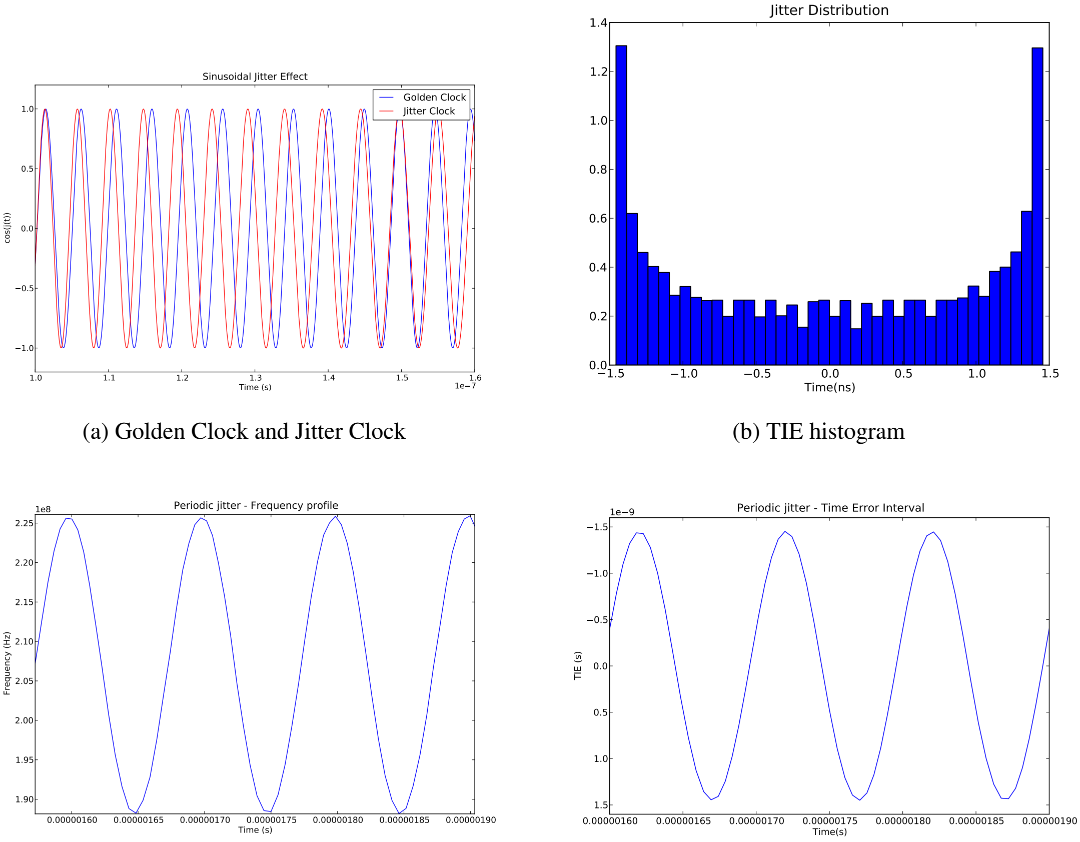
Measurement	Formula	Expected	Obtained
Max Frequency	$f_s(1+A_j*2*\pi*\frac{f_j}{f_s})$	$225.850e6$	$226.030e6$
Min Frequency	$f_s(1-A_j*2*\pi*\frac{f_j}{f_s})$	$188.150e6$	$226.030e6$
Max TIE	$0.3*Tbit$	$1.449e-9$	$1.456E-9$
Min TIE	$-0.3*Tbit$	$-1.449e-9$	$-1.456E-9$

Table 1.1: Sinusoidal Jitter Simulation

1.2.2 Data dependent jitter

Data dependent jitter is the deviation of each data edge from the ideal point due to the memory of the system. Duty cycle distortion and Inter-symbol interference are good examples of the system memory, this means, that the current bit waveform will depend from the previous bits. Duty cycle distortion refers to a deterministic and bounded value characterized by the period difference of alternate consecutive bits. Inter-symbol interference refers to the band limited characteristic of the system (cable, driver, connectors), this effect is proportional to the number of consecutive equal transmitted bits.

Duty cycle distortion phenomenon is related with the unbalancing of the logical ones generation and the logical zeros generation. Starting by looking at the driver side, today's driver are half



(a) Golden Clock and Jitter Clock

(b) TIE histogram

(c) Frequency Variation of Jitter Clock

(d) Difference between Jitter Clock and Golden Clock edge locations

Figure 1.4: Sinusoidal Clock Jitter example ($f_s = 207e6, f_j = 10e6$, $A_j = 0.3 * Tbit$)

rate this means that the high speed serial clock is in terms of frequency half of the bit rate, the selection between bits is made based on the clock level, this means that the duty cycle of the high speed clock will be directly translated into the overall duty cycle distortion. Remaining at the driver side there are two other factors responsible for duty cycle distortion, different rise and fall times and unbalancing between positive and negative networks. The last two in conjunction with the clock duty cycle distortion can produce a clean signal (cancelation), but the majority of the cases the three contribute to the duty cycle distortion, on this case the overall $DCD = ck\_dc + rt\_ft + n\_p \ \ (s)$, where ck_dc represents the clock duty cycle distortion, rt_ft represents the difference between rise and fall times and n_p represents the difference between positive and negative networks (all parameters are specified in seconds).

Inter-symbol interference phenomenon is related with the low pass characteristic of the link. Different edge patterns have different frequency components. Fast-changing patterns behave as high-frequency signals; slow-changing patterns behave as low-frequency signals. Because of the conductors filtering effects, different patterns propagate at different speeds through the conductors. This difference in propagation speeds causes bits to smear into adjacent bits, resulting in ISI [1].

Data dependent jitter decomposition isn't trivial, at the end the designer will be able to specify only a peak to peak number, the decomposition of this final value by each circuit present on the entire system will be unpractical. The solution is to measure each block independently, $DDJ_t = \sum_{i=0}^{N} DDJ_{block}(i)$ where DDJ_t represents the total data dependent jitter of the system (theorical),N represents the total number of blocks and $DDJ_{block}(i)$ represents the data dependent jitter of each block. As stated before the observed Total DDJ can be lower than DDJ_t. To better understand this type of effects let's take a look at a real system:

- Two independent data sources LFSR7 and LFSR15. Linear feedback shift register module 7 (LFSR7) ensures a maximum run length of 127 then the sequence is repeated, the maximum number of consecutive equal bits is 7. LFSR15 has a bigger run length 32767, the maximum number of consecutive equal bits is 15. Each data source generates 3.4Gbps ($tbit \simeq 294ps$).

- Two duty cycle distortion jitter sources per each data source, with 3.5ps and 6.5ps of deterministic jitter respectively.

- Cable with a low pass characteristic per each data source. The frequency domain response is described on figure 1.5.

The characterization of the system was done using a spice simulator, cable was modulated as a series of inductors, resistors and capacitors. The frequency response of the cable was obtained doing an AC analysis. Spice simulator was used to simulate the entire system, for that reason the data sources and jitter source were described in spice like format. During the simulation 34000 bits were sent, the final results were post-processed to generate the jitter distributions.

Figure 1.6 contains booth waveforms, LFSR7 and LFSR15 systems, the differences on the waveforms aren't clear, one thing is observed: the signal on the LFSR15 seems to be "slower" than the one on LFSR7 system, since it takes more time to reach the final value. On figure 1.7 the

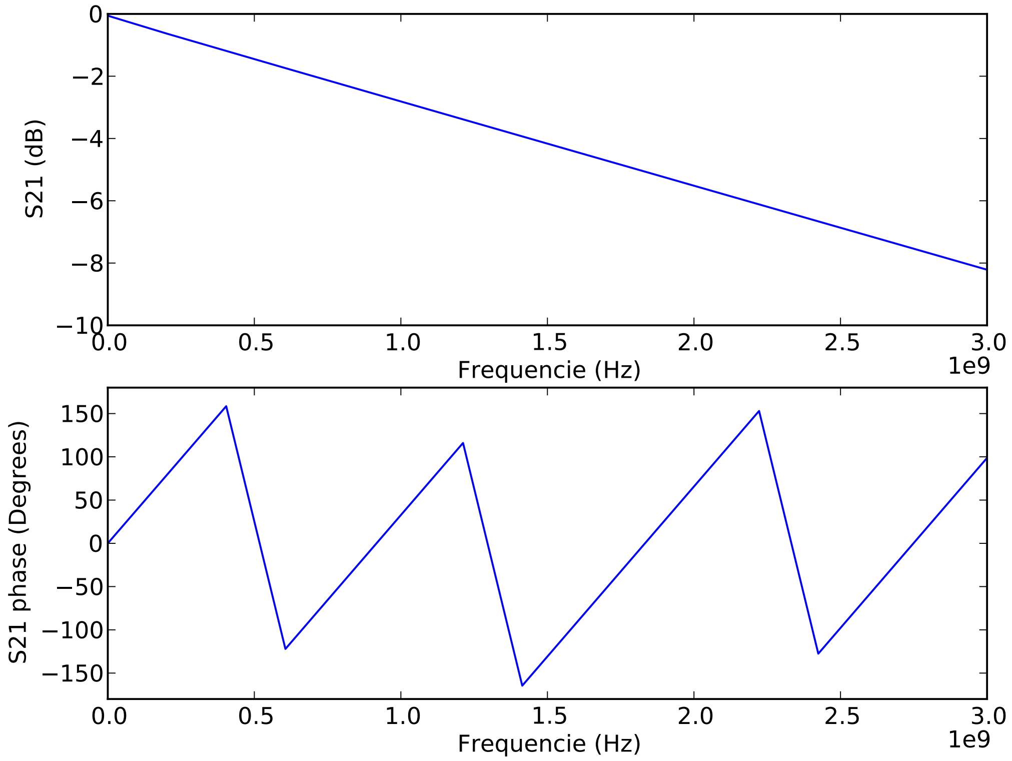


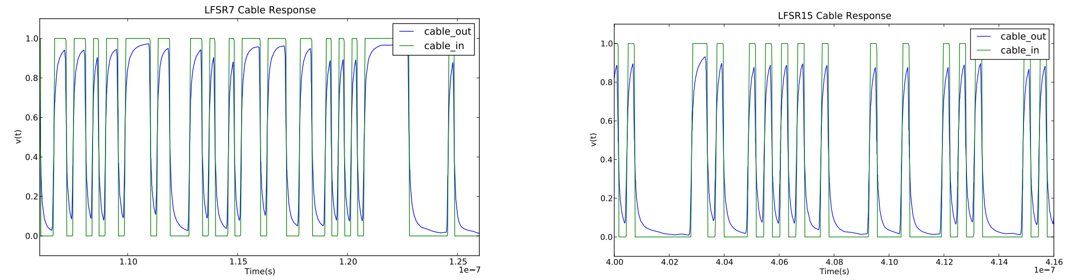
Figure 1.5: Cable frequency characteristic

effect of the data pattern is clearly observed through the respective probability density functions. On LSFR7 system it's possible to separate the DCD from ISI, as expected the DCD "diracs" are placed on 0ps, 3.5ps, 6.5ps and 10ps (sum of the two DCD sources), the ISI effect is seen on the spread effect, resulting on $DDJ_{LFSR7} = 12ps$.ISI has a bigger effect on LFSR15 system, the only difference between LFSR7 and LFSR15 is the data pattern, mining that ISI is directly related with the data pattern. The DCD "diracs" aren't clear on LFSR15 this is due to the ISI bigger factor and as expected the total DDJ is bigger $DDJ_{LFSR15} = 16ps$.

Previous example clearly shows the importance of a correct data codification on today's standards, because the increase of speed requires lower jitter values, since data pattern is translated to jitter, lower jitter values allow higher speed interfaces and lower BER.

1.2.3 Bounded uncorrelated jitter

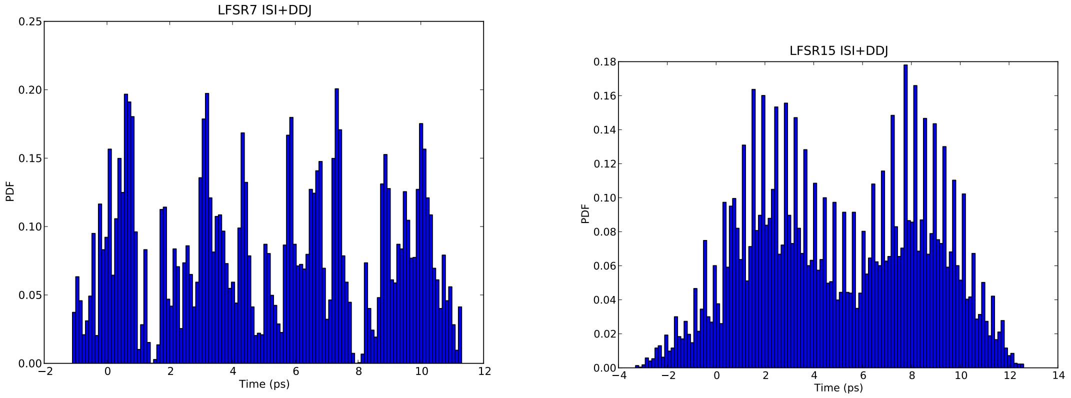
Crosstalk in most of the cases represents BUJ. This phenomenon can be observed when two pcb lines (lineA and lineB) separated by a thin space transmit different patterns, lineB will introduce trough the intrinsic capacitors/indutors signal on lineA and vice-versa. The signal at the end of the lines in some cases will have an higher amplitude in other cases a lower amplitude, resulting on different crossing points (timing jitter).



(a) LFSR7 waveform

(b) LFSR15 waveform

Figure 1.6: Data patterns LFSR15 and LFSR15 waveforms at the input and output of the cable



(a) LFSR7 PDF

(b) LFSR15 PDF

Figure 1.7: DDJ probability density function for LFSR7 and LFSR15 systems

Crosstalk effect can be reduced by separating the lines, increasing the shield or using differential signals. No further analysis will be done over this topic, since the use of differential signals reduces this effect.

1.2.4 Random jitter

Random Jitter (RJ), as the name implies, is caused by things like thermal oscillations, flicker, and shot noise that result in levels of jitter that cannot be predicted on an edge-by-edge basis. The thing to understand about RJ is that it is caused by a huge number of very small effects. The variation in the width of the traces on a printed circuit board, fluctuations of conductivity in a conductor caused by impurities, variations in resistance caused by random fluctuations in the local voltage that individual electrons feel, and literally zillions of other tiny effects combine in a way that is dictated by the fundamental theorem of Statistical Physics: The Central Limit Theorem, which makes the whole thing very simple.The Central Limit Theorem says that a large number of small independent processes combine in a way that follows a Gaussian distribution. This type of jitter can't be described by a simple peak to peak value, since such value will differ from acquisition to acquisition even if the pattern sent was the same in all acquisitions. It's necessary to use a different measure to describe such variations, for this reasons RJ is described as a probability function (PDF), such waveform can be described by equation 1.7 where σ represents the standard deviation and μ represents the mean value.

$$PDF_{RJ}(x) = \frac{1}{\sigma\sqrt{2\pi}} exp[-\frac{(x-\mu)^2}{2\sigma^2}] \tag{1.7}$$

Random jitter can't be described as a maximum peak to peak value, but it can be described having a maximum peak to peak value within for certain probability. Taking into consideration BER, a peak to peak value for RJ can be defined ensuring that such value will have a probability of occurrence bellow than the considered BER, this definition allows de designer to take into consideration a realistic RJ value on the total jitter definition. Figure 1.8 describes the relation between $BER(x)$ and $PDF_{RJ}(x)$, it's clearly seen that the same $BER(x)$ is equal on the right and left side ($BER_l(x)$ and $BER_r(x)$, the contribution for total jitter will be equal to the contribution of booth sides. Mathematically $BER_r(x) = \oint_x^\infty PDF_{RJ}(u)du$, resulting on a total jitter of $2 * BER_r^{-1}$. Ideally the designer would like to have an equation on the form of:

$$TJ_{RJ}(BER) = 2 * N\sigma \ \ (s) \tag{1.8}$$

Removing the need to integrate an exponential function. As described above,

$$BER_r(x) = \frac{1}{2\sqrt{2\pi}} \oint_x^\infty exp[-\frac{u^2}{2\sigma^2} du] \tag{1.9}$$

Using the following variable transformation $a = (\frac{u}{\sqrt{2}\sigma})$,

$$BER_r(x) = \frac{1}{2\sqrt{2\pi}} \oint_x^{\infty} exp(-a^2)da \tag{1.10}$$

Such function is closely related with the complementary error function described as $erfc(x) = \frac{2}{\sigma\sqrt{2\pi}} \oint_x^{\infty} exp(-u^2)du$, resulting

$$BER_r(x) = \frac{1}{2}erfc(a) \tag{1.11}$$

Applying a second variable transformation $u = N\sigma$, results on $a = \frac{N}{\sqrt{2}}$. The multiplication value (N) that returns the desired jitter for the specified BER can be find by replacing $BER_r(x)$ by the desired bit error ratio and solving the equation in relation to N.

$$N = \sqrt{2} * erfc^{-1}(2 * BER) \tag{1.12}$$

From equation 1.12 and 1.8, the total jitter value for a given BER can be described as:

$$TJ_{RJ}(BER) = 2 * \sqrt{2} * erfc^{-1}(2 * BER) * \sigma \ \ (s) \tag{1.13}$$

Complementary error function is tabulated allowing the designer to easily find the total jitter value of the random jitter for a specific bit error ratio. Table 1.2 contains the multiplication factor for a few BER cases. Random jitter represents a big issue on today systems, even a random jitter with a low standard deviation can result on a big total jitter due to the multiplication factor.

BER	N
$1e-6$	4.75
$1e-8$	5.61
$1e-10$	6.36
$1e-11$	6.71
$1e-12$	7.03
$1e-13$	7.35
$1e-14$	7.65
$1e-15$	7.94

Table 1.2: TJ_{RJ} multiplication factor

1.3 Total jitter

Total Jitter is an increasingly important quantity in the development and specification of serial data links but, while it is well defined, it is not well understood. Total Jitter is like peak-to-peak jitter but referenced to a given Bit Error Ratio. Total jitter can’t be described as a static and stand-alone value, since one of the components is RJ that varies with BER.

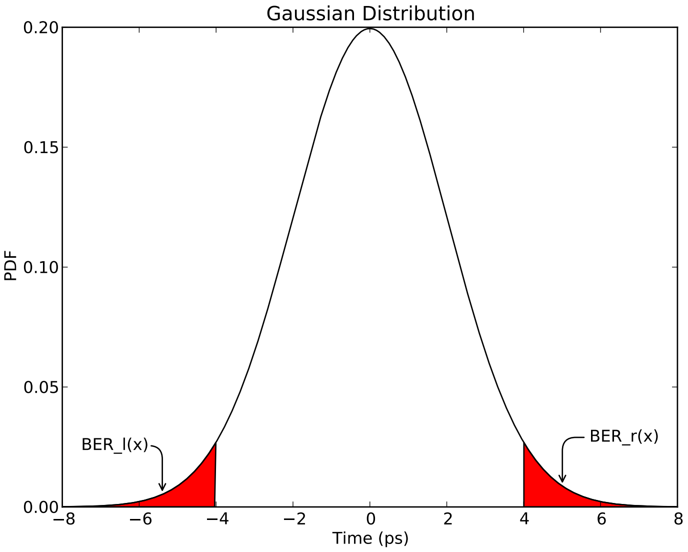


Figure 1.8: Probability density function for Random Jitter ($\mu = 0, \sigma = 2ps$)

Total jitter can be described as a probability density function (PFD) as the random jitter. Such function can be obtained trough time interval error (TIE) measurements translated to a probability histogram, like on figure 1.9. All jitter elements defined previously will be combined, jitter provided by deterministic sources will be summed and jitter provided by random phenomenons will be combined on a single σ_{Total}. Total jitter will be estimated based on the following equations:

$$\begin{aligned} DJ_{Total} &= DCD(p-p) + ISI(p-p) + BUJ(p-p) \qquad (1.14) \\ &= ck\_dc(p-p) + rt\_ft(p-p) + n\_p(p-p) + ISI(p-p) + BUJ(p-p) \\ \sigma_{Total} &= \sqrt{\sigma_1^2 + \sigma_2^2 + \cdots + \sigma_n^2} \end{aligned}$$

Resulting on,

$$TJ(BER) = 2 * N(BER) * \sigma_{Total} + DJ_{Total} \quad (s) \tag{1.15}$$

The *N(BER)* value can be found on table 1.2. In terms of jitter PDF it's necessary to perform a convolution between all the different PDF, as described on eq. 1.16

$$PDF_{TJ} = PDF_{RJ_{Total}} \otimes PDF_{DJ_Total} \tag{1.16}$$

1.4 Jitter measurement techniques

Jitter measurements can be done using different equipments, there are two major groups: BERT Scan and Oscilloscopes. Bit Error Ratio Tester (BERT) equipments can automatically measure the total jitter value, the measurement is based on the number of errors found for a specific number

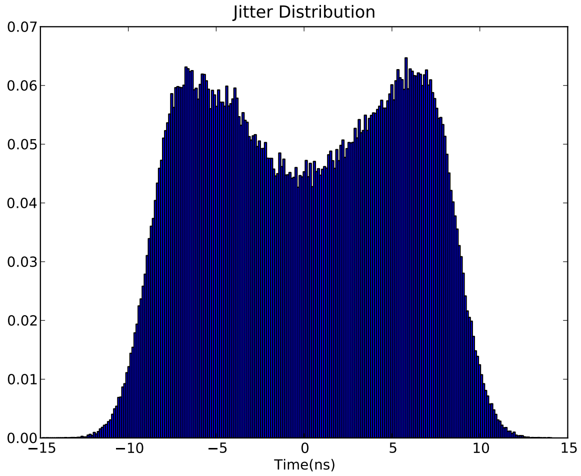


Figure 1.9: Probability density function for Total Jitter

of acquisitions, these equipments can accurately measure the total jitter. Oscilloscopes work on a different way, these equipments acquire samples therefore additional data processing is needed to estimate the total jitter value, the major limitation on these equipments is related with the maximum number of samples that can be stored, BERT equipments on opposite only store the number of errors allowing accurate total jitter measurements.

BERT equipment is able to shift the sampling point across an entire bit time, to perform a BERT scan the equipment generates a known pattern of data and artificially slides the sampling point across an entire bit and counts the number of errors found in each sampling point. The number of errors acquired on each sampling point is then divided by number of transmitted bits resulting on graphic that has on x-axis time and on y-axis the bit error ratio value (bathtub plot). Such equipment is able to observe errors (not timing errors, but decoding errors), where it's assumed that such decoding errors are caused due to timing or voltage errors. BERT scan can accurately specify the total jitter boundaries, but no information related with voltage or timing margin is provided. The number of errors will be higher near to the transition bit time, 0 time and bit time, since a small variation on the received signal will be considered as an error. At the middle of the unit interval the number of errors will be the lowest since at this point the allowed variation on the received signal is maximum, for this reason clock and data recovery circuits try to sample the bit at the middle of the bit time.

Oscilloscope equipment works on a different way, this equipment captures the received signal, but since the signal is store on a memory, the number of acquired samples is low compared with BERT equipments. Oscilloscopes are very useful for determining the jitter components (voltage and timing jitter), since isn't possible to acquire enough samples to directly specify the total jitter, post processing techniques are used to determine that value.

Today's standards define a eye diagram mask that should be respected. Eye diagram measurements are easily performed on an Oscilloscope, since the principle is to represent all captured bits on a single waveform with a with a timing window equal to two bit times, this means that from

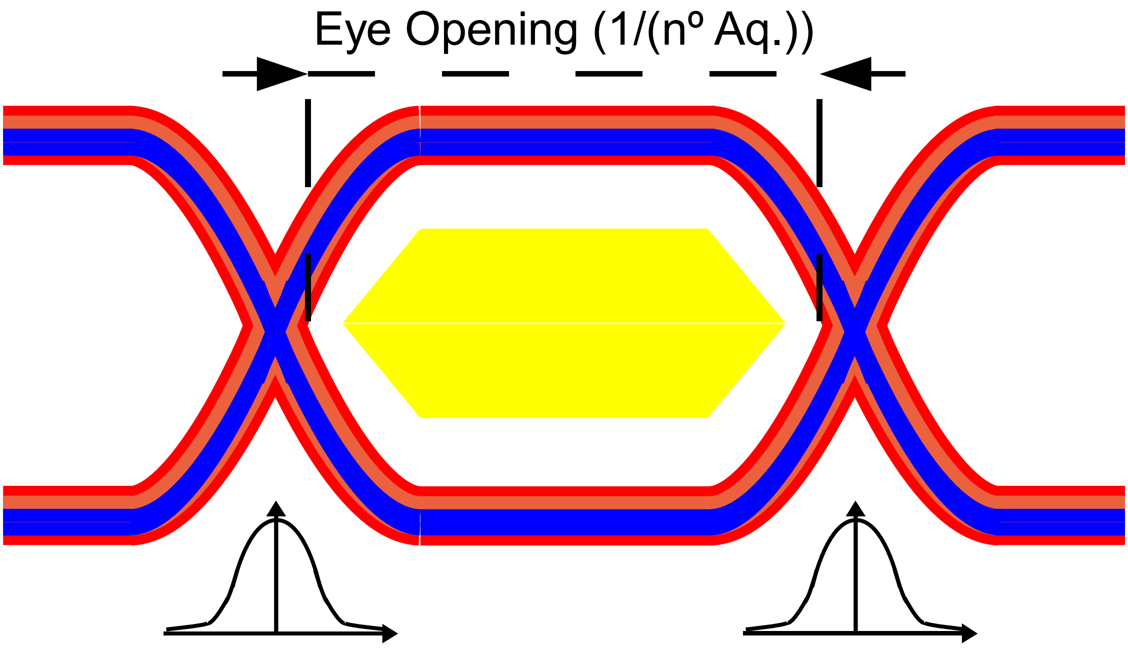


Figure 1.10: Eye diagram

two bits in two bits the time is reset allowing a continuously waveform capture using a time scale of two bits (see figure 1.10). Since all acquired bits waveforms are now place on the bit time is then possible to measure the eye diagram opening and voltage margin. On today's standards its also described an eye diagram mask that shouldn't be crossed by the acquired signal (yellow area in the center of the figure). Eye diagram should be obtained using a clock recovery unit (CRU), sometimes called golden PLL, to correctly defined the bit period of each bit, this unit emulates the PLL and CDR circuits present on the receiver side (figure 1.11). The eye diagram specification also defines the characteristic of such units. The idea of using it is to remove the low frequency noise that is correctly tracked on the receiver side, if such circuit wasn't use a complete closed eye diagram will be obtained. The combination of an oscilloscope, CRU (clock recovery unit) and data processing software ensures a correct jitter measure. Total jitter measurement starts by the generation of a TIE histogram, this task is done using the trigger produced by CRU as the ideal edge location, then it's necessary to determine the difference between the ideal edge and the real edge location, the software uses an internal algorithm to accurately determine the transition points

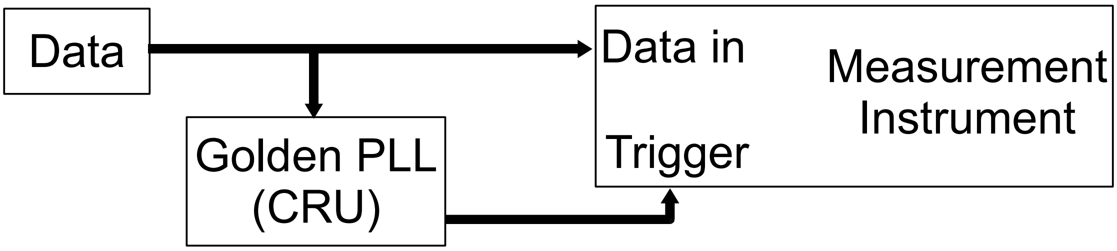
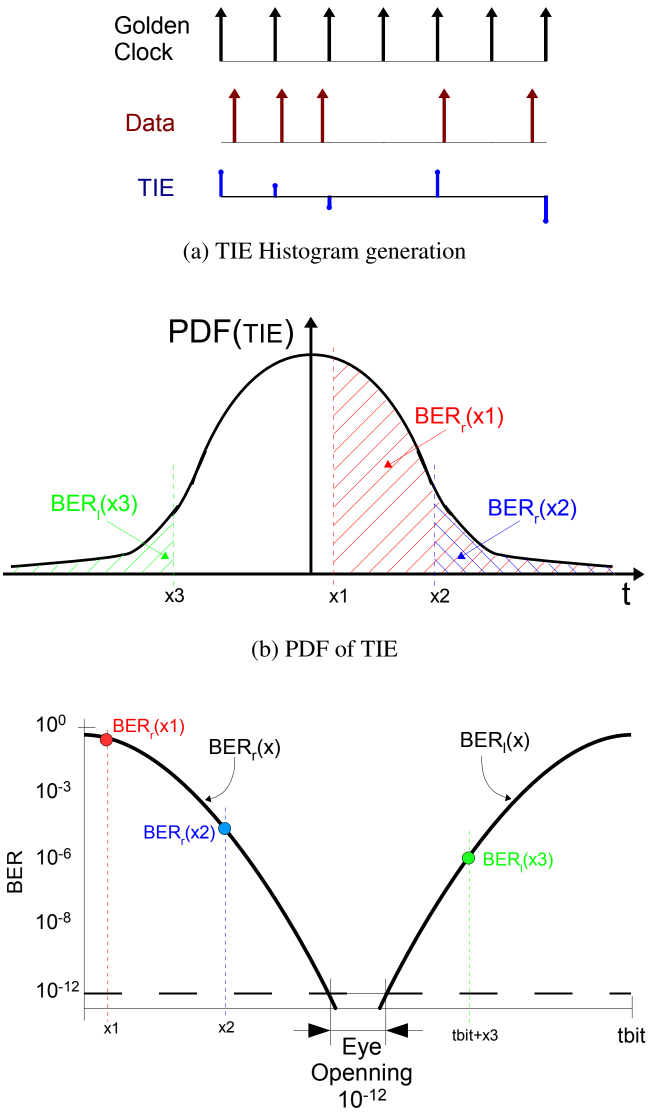


Figure 1.11: Eye diagram



(a) TIE Histogram generation

(b) PDF of TIE

(c) Bathtub Plot

Figure 1.12: Translation from TIE to Bathtub plot

of the real signal, then it's just necessary to store the difference between both transition into a histogram. With the TIE histogram the Bathtub diagram can be directly determined by calculating the correspondent BER for each histogram point. Total jitter for the desired BER can be obtained trough a linear approximation. Figure 1.12 exemplifies that process.

References

[1] Andy Kuo Sassan Tabatabaei Nelson Ou, Touraj Farahmand and André Ivanov. Jitter models for the design and test of gbps-speed serial interconnects. *Copublished by the IEEE CS and the IEEE CASS*, pages 0740–7475, 2004.

[2] Masahiro Ishida Toshifumi Watanabe Takahiro J. Yamaguchi, Mani Soma and Tadahiro Ohmi. Extraction of instantaneous and rms sinusoidal jitter using an analytic method. *IEEE Transactions on Circuits and Systems*, 2003.

[3] Tad Nwasniewski Nathan Kiddinapillai. Jitter tolerance estimation of a 3x oversampling cdr using event-driven simulation. *Microsystems and Nanoelectronics Research Conference*, 2009.