Power Integrity and Noise Coupling Effects on Signal Integrity – Methodology for Identifying the Deterministic Jitter Components and their Generating Sources in Data Communication Systems

Cosmin Iorga, Ph.D., NoiseCoupling.com
[cosmin.iorga@noisecoupling.com, (805)231-9786]
Abstract

This paper presents a circuit and methodology for identifying the sources of deterministic jitter in data communication systems. On-chip waveform viewers sample the data stream signal and the power distribution noise on chip, package, and PCB. A mathematical algorithm calculates the Fourier transforms of the data stream and noise waveforms, correlates the spectral components, and identifies the source of each deterministic jitter component in the data stream. A software program controls the waveform recording, data processing algorithm, and color-coded graphical display of magnitude, frequency, and generating source for each deterministic jitter component.

Author Biography

Dr. Cosmin Iorga is the founder and president of NoiseCoupling.com, a company that provides noise coupling, power integrity, and signal integrity solutions at integrated circuit, package, printed circuit board, and system levels. Dr. Iorga has accumulated more than 20 years of experience in signal integrity, mixed-signal board and system level design, and, in the most recent 8 years, high-speed integrated circuit design using CMOS and BiCMOS technologies. Dr. Cosmin Iorga holds a Ph.D. in Electrical Engineering from Stanford University. He has filed more than 10 patents with 8 granted so far, covering innovative solutions in noise coupling suppression and signal integrity improvement. Dr. Iorga is the author of the book: “Noise Coupling in Integrated Circuits: A Practical Approach to Analysis, Modeling, and Suppression”. Dr. Iorga provides consulting services, training courses, and team mentoring covering noise coupling, power integrity, and signal integrity.
1. The Jitter Problem in Data Communication Systems

Jitter in data communications reduces the setup and hold margins of data samplers increasing the bit error rate (BER). The main two components of jitter are random jitter (RJ) and deterministic jitter (DJ).

Random Jitter (RJ)

The random jitter (RJ) is originated by the thermal noise of the electronic components used in the generation, transmission, and reception of the electric signals making the data stream. The random jitter follows the Gaussian distribution, and it is often expressed as rms value. The peak to peak jitter can be as large as six times the rms value. Random jitter is typically reduced by properly designing the driver and receiver circuits.

Deterministic Jitter (DJ)

The deterministic jitter (DJ) has three components: data dependent jitter (DDJ) caused by the data stream itself, crosstalk induced jitter (CJ) caused by electromagnetic coupling from other signals, and noise coupling induced jitter (NCJ) caused by noise coupling from other circuits in the system. Each jitter component is generated through a different physical mechanism, and identifying these mechanisms is essential for implementing suppression techniques.

Data Dependent Jitter (DDJ)

The data dependent jitter (DDJ) is caused primarily by the frequency bandwidth limitation of the transmission line cables and printed circuit board traces and by the impedance mismatch reflections.

The bandwidth limitation is caused by the skin-effect and degrades the rising and falling edges in the form of waveform roll-off. Depending on the time separation between consecutive pulses, the signal may not have enough time to completely settle at either high or low levels before the next edge comes. Because the signal is not settled, a rising edge starts above the logic low level shifting it earlier in time, and a falling edge starts below the logic high level also shifting it earlier in time.

Impedance mismatch of various components in the transmission line path generate reflections which show as positive or negative "bumps" on the signal waveform. These bumps may cause the transition edges start at a higher or lower voltage level, which then shifts the edge earlier or later in time depending on the positive or negative direction of
the bump. The choice of reflection bumps occurring on transition edges depends on the sequence of zeroes and ones in the data stream.

Due to these edge shift mechanisms, both the bandwidth limitation and impedance mismatch reflections produce jitter dependent on the data transmitted through the path. This jitter is typically called data dependent jitter (DDJ) or inter symbol interference (ISI). Besides the contributions of bandwidth limitation and mismatch reflections, the DDJ and ISI contain also the effect of duty cycle distortions from active circuits in the path.

**Crosstalk Jitter (CJ)**

Crosstalk between the active signal and neighboring signals produce waveform aberrations in the form of positive and negative "bumps" similar to the impedance mismatch reflection bumps. If these bumps occur on the rising or falling edges they induce timing shifts in the rising and falling edges, which are referred as jitter. Crosstalk occurs mainly on the transmission line paths on the packages, boards, and backplanes.

**Noise Coupling Jitter (NCJ)**

The simultaneously switching transient currents flowing through the parasitic inductance of the power and ground delivery system cause supply and ground noise. This noise then propagates through the power distribution and chip substrate. Noise coupling affects the electronic circuits that process the data signals. Noise coupling may modulate the propagation delay through one of these circuits, which reflects in rising and falling edges shifts and thus jitter. Noise coupling may also modulate the threshold voltage level of the switching point, thus shifting in time the transition event. This shift translates also into jitter. Supply and ground noise at the transmitter or at the receiver is seen as level noise on single ended data signals or common mode noise on differential data signals. This voltage noise may also translate into jitter.

**Things to Remember about Jitter Components**

Random jitter depends on the thermal noise of the devices used in the data processing circuits. Data dependent jitter depends on the quality of the transmission line interconnects and the sequence of ones and zeroes in the signal pattern. Crosstalk and noise coupling jitter depends on the activity of other circuits on the chip, board, and system.

**2. Addressing Jitter Problems**

Jitter reduction addresses each jitter component described above. Pre-emphasize of transmitted data or equalization of received data compensates for skin effect, dielectric absorption, and multiple reflection effects in transmission line cables and PCB traces. Designing high quality terminations, connectors, and connector launches reduce the
impedance mismatch reflections on the transmission line path. Crosstalk is addressed in layout by a variety of techniques including increased spacing between parallel signal traces and inserted shielding traces between signal traces. Noise coupling jitter is addressed by reducing the noise generation especially by controlling the simultaneously switching noise and by designing circuits less sensitive to power supply and substrate noise.

While a lot of effort is made to reduce the jitter, timing errors in data sampling still remain. To further troubleshoot these problems analyzing the signal integrity of waveforms is essential.

Measurements of waveforms are typically represented in eye-diagram format, on which the setup time, hold time, and noise margins can be measured. This information helps verify the signal integrity and optimize the pre-emphasize and equalization circuits. While optimizing the pre-emphasize and equalization circuits compensates primarily for skin-effect, discontinuity reflections, and dielectric absorption losses, there are applications where the remaining jitter is still unacceptable. In these applications troubleshooting is necessary to identify the problems root-cause and develop solutions. To do this it would be nice to be able to identify each jitter component and what generates it in the system. The proposed circuit and methodology addresses this problem.

3. Circuit and Methodology to Identify the Deterministic Jitter Components and their Sources in the System

Architecture

The proposed technique measures the waveform at the receiver and simultaneously the noise on the power supplies on chip, package, PCB, and chip substrate. A mathematical algorithm then calculates the Fourier transform of the data and noise signals, extracts the jitter components, and identifies where each of these components is originated from. This algorithm implemented in a Microsoft Excel Visual Basic program displays the magnitude, frequency, and where in the system each jitter component is generated.

The waveform measurement can be done either externally using measurement instruments or internally on the chip using waveform viewer samplers. In this work internal waveform samplers are used.

Figure 1 shows a simplified block diagram of the proposed circuit. This circuit consists of one waveform sampler for the data stream signal and another waveform sampler for the noise on the chip substrate and power supplies on chip, package, and PCB.
Figure 1. Proposed circuit architecture implementing noise sensors and samplers for data stream and noise

The data stream signal is routed to one waveform sampling circuit. Multiple noise sensors probe the chip substrate as well as the power and ground on chip, package, and PCB. All these sensors are multiplexed into a single signal, which is routed to the second waveform sampling circuit. A pair of signal sampler and noise sampler is placed on each of the transmitter and receiver chips.

Each waveform sampler uses a latched comparator strobed by an adjustable delay clock. The comparator reference is provided by a digital-to-analog converter (DAC). Both strobe delay and DAC level are controlled by a software program. The input signal into this comparator is set in repetitive mode and the timing of the sampling clock is synchronized with the input signal.

For a given sampling clock timing delay, the DAC output is swept across the full range. The recorded digital data shows a sequence of 1’s corresponding to the case when the DAC output voltage is lower than the input signal followed by a sequence of 0’s corresponding to the case when the DAC output voltage becomes higher than the input. The transition from 1’s to 0’s corresponds to the DAC value equal to the input signal. We record this value as the input magnitude at the corresponding timing delay. Alternately, instead of sweeping the DAC, a binary search can detect this level. We repeat this level detection process while sweeping the sampling clock delay and record the input levels in a time-voltage table, from which the input waveform can be reconstructed.

The signal sampler uses the data stream clock as reference for sampling clock. Since the supply and ground noise are correlated with the switching activity on the chip, the clock of the noise sensors sampler needs to use a local system clock rather than the data stream clock. When not in troubleshooting mode, the noise sampler can be disabled. Figure 2 shows the block diagram of a jitter probing system implemented in a generic data communication circuit.
Figure 2. Block diagram of a jitter probing system implemented on a generic data communication system.

The data stream is sent from one board through the backplane to a second board. This data path includes the pins, wire traces and connectors on chips, packages, PCBs, and backplane. Noise on chip, package, and PCB may couple into the active circuits that process the data stream generating jitter. Sensors probe the substrate noise on both chips as well as power and ground noise on the chip, package, and PCB. Figure 2 illustrates the location of probing points by the labels: SNS_VDD_CHIP, SNS_VDD_PKG, SNS_VDD_PCB, SNS_VSS_CHIP, SNS_VSS_PKG, SNS_VSS_PCB, and SNS_VSUB_CHIP. When multiple supply domains exist, sensors may probe each of these domains. The schematics of each individual sensor are shown in Figure 3.

Figure 3. Sensors schematic: (a) bias circuit, (b) package ground, (c) PCB ground, (d) chip substrate, (e) chip supply, (f) package supply, and (g) PCB supply.
All sensors use the same bias reference current, which is fanout through a multiple-branch current mirror. Each sensor uses a diode connected transistor to level-shift the measured value while maintaining a close to unity voltage gain. The bandwidth depends on the transistor characteristics and loading capacitance. Based on what types of devices are available, different fabrication technologies may require different types of sensors.

**Functionality**

The digital activities of both transmit and receive chips are set in repetitive modes to assure correlation between the generated noise and the sampler timing clock. The data stream is set in a PRBS mode for eye-diagram measurements and in a uniform 101010 pattern for jitter analysis. The signal sampler captures multiple cycles of the data stream and the noise sampler captures the sensors’ outputs.

For a given sampling clock timing set, the DAC output is swept across the full range. The recorded digital data shows a sequence of 1’s corresponding to the case when the DAC output voltage is lower than the sensor output followed by a sequence of 0’s corresponding to the case when the DAC output voltage becomes higher than the sensor output. The transition from 1’s to 0’s corresponds to the DAC value equal to the sensor input. We record this value as the input magnitude at the corresponding timing set. Alternately, instead of sweeping the DAC, a binary search can detect this level. We repeat the level detection process while sweeping the sampling clock timing and record the input levels in a time-voltage table, from which the input waveform can be reconstructed.

Figure 4(a) shows an eye-diagram example of a sampled and reconstructed PRBS waveform after passing through a transmission path that includes skin-effect losses, crosstalk, and coupling from power supply noise. It can be noticed that the eye-opening is very small, which makes the bit detection almost impossible. This is mainly due to jitter and waveform roll-off. Equalization compensates for waveform roll-off leaving only the effects of jitter, as shown in Figure 4(b). Turning off the switching activities on both transmitter and receiver chips we obtain a larger eye-opening as shown in (c).

![Figure 4](image-url)

Figure 4. Eye-diagram of a received signal before equalization (a), after equalization (b), and after suppressing the deterministic jitter (c)
While in most cases equalization alone provides an acceptable eye-opening, there are applications where the equalized signal still has significant amount of jitter. To reduce further more the remaining jitter, it is necessary first identify the sources of this jitter. While part of it consists of random jitter, the dominant components are often generated by crosstalk and power distribution noise.

To further analyze the contribution of each jitter source, we send a uniform 101010 pattern at 3.2GHz and reconstruct the received waveform after equalization with the signal sampler. We further compute the Fourier transform of this waveform. Figure 5(a) shows the magnitude of the Fourier transform of the 101010 pattern reconstructed waveform using 4096 samples.

![Fourier Transform of Sampled Data Stream](image)

Without jitter we expect to see a 3.2GHz fundamental component and a harmonic at 9.6GHz. The additional components represent deterministic jitter. We record the first jitter component of 0.03V magnitude at 1GHz, and from calculations we deduce that the components at 3GHz, 5GHz, 7GHz, and 9GHz are harmonics of this 1GHz fundamental. Similarly, we record another two jitter components at 1.6GHz and 2.4GHz, and we deduce from calculations that all the remaining components are only harmonics of the already identified fundamentals. So the deterministic jitter components are located at 1GHz, 1.6GHz, and 2.4GHz.

We analyze next the frequency components of the sensors waveforms. Figure 5(b) shows these components for the power supply sensor on the receiver chip. Using the same procedure we extract a 0.1V magnitude component at 1GHz and a 0.07V magnitude component at 2.4GHz. From correlation with the data stream components we decide that the power supply noise on the receiver chip injects deterministic jitter of 1GHz and 2.4GHz frequencies. We repeat this analysis for all sensors. The remaining unmatched jitter components in the data stream, 1.6GHz in this case, are generated from signal crosstalk.

To study where the crosstalk occurs, we analyze the signals traveling on adjacent traces and we found an 11001100 data stream flowing through a nearby channel. At 3.2GHz clock, this data stream has a 1.6GHz frequency. To further study if this signal couples through crosstalk into our data stream, we change the pattern from 110011 to 111000111,
which changes the frequency from 1.6GHz to 1066MHz. We sample the data stream and perform the Fourier transform. The resulted frequency spectrum is shown in Figure 6(a).

![Fourier Transform of Sampled Data Stream (1.6GHz 010101) pattern with 00111000 pattern through adjacent backplane trace](a)

![Fourier Transform of Sampled Data Stream (3.2GHz 010101 pattern with 0001111000 pattern through adjacent backplane trace)](b)

Figure 6. Fourier transform of a 111000111 data stream (a), and 111100001111 data stream (b)

Extracting the jitter components from Figure 6(a) plot we notice that the power supply induced jitter remained the same while the 1.6GHz component moved to 1066MHz. Therefore, we conclude that the 1.6GHz 0.08V jitter component comes from crosstalk with a neighboring signal trace. Additional pattern change to 111100001111 reduces the crosstalk jitter frequency to 800MHz, as shown in Figure 6(b).

This algorithm has been implemented in a Microsoft Excel Visual Basic program. The results are displayed in a two-dimensional block chart showing the freq and magnitude of each jitter component and, through color-coding, the source that generates it. For the above crosstalk identification example these plots are shown in Figure 7.

![Sources of Deterministic Jitter](a)
![Sources of Deterministic Jitter](b)
![Sources of Deterministic Jitter](c)

Figure 7. Deterministic jitter components generated by power supply noise and crosstalk

Figure 7(a) shows the 1Ghz, 1.6GHz, and 2.4GHz jitter components. The crosstalk component at 1.6GHz, shown in green color, is coupled through crosstalk from an adjacent channel running a 110011 data pattern, which divides by two the 3.2GHz to 1.6GHz. By changing this pattern to 111000111 we see the green crosstalk bar shifted to the 3.2GHz / 3 = 1.066GHz in chart (b). Changing again the pattern to 111100001111 we see the crosstalk bar shifting more down to 800MHz in chart (c), as previously explained using the Fourier transform plots.
In a general case when jitter couples from various sources probed by sensors, the result shows a block bar corresponding to each jitter source, as shown in Figure 8. The TX and RX notations correspond to sensors placed on the transmitter and receiver circuits.

![Sources of Deterministic Jitter](image)

Figure 8. Deterministic jitter components: magnitude, frequency, and their generating sources in the system

### 4. Conclusions

This work proposed a methodology for waveform sampling of a data stream and various noise sensors in the system, and a data processing algorithm that identifies the deterministic jitter components in the data stream and correlates them with specific noise sources and crosstalk paths in the system. The implementation of this algorithm in a software tool provides a graphical display of each jitter component magnitude, frequency, and generating source. Troubleshooting can further use this information to identify and study the jitter generation mechanisms and jitter coupling paths, and to develop jitter suppression solutions.