DESIGNCON® 2022
WHERE THE CHIP MEETS THE BOARD

Applications of High Bandwidth AWGs in Receiver Testing: Tricks of the Trade

Anton Unakafov, BitifEye
anton.unakafov@bitifeye.com

Ransom Stephens, Ransom's Notes
ransom@ransomsnotes.com

Nithin Parameshwaraiah, BitifEye
nithin.parameshwaraiah@advantest.com

Julien Henaut, BitifEye
julien.henaut@bitifeye.com

Afshin Attarzadeh, BitifEye
afshin.attarzadeh@bitifeye.com

Valentina Unakafova, BitifEye
valentina.unakafova@bitifeye.com

Philipp Schleicher, BitifEye
philipp.schleicher@bitifeye.com

Abstract

Deep-memory, high-bandwidth Arbitrary Waveform Generators (AWGs) are ideal tools for creating the signals required for stressed receiver tolerance testing. They can transmit waveforms with precise Insertion Loss (IL), Return Loss (RL), Inter-Symbol Interference (ISI), crosstalk, random and sinusoidal noise and jitter, even nonstationary events, and Transmitter FeedForward Equalization (TxFFE). In principle, AWGs are magic. In practice, they are quirky and temperamental, with limited resolution, imperfect interpolation, and finite sampling rate. In this presentation, we share our experiences using (and trying to use) AWGs in automated receiver tests for compliance to a variety of standards: HDMI 2.1, MIPI C-PHY, MIPI D-PHY, and MIPI A-PHY.

Authors' Biographies

Anton M. Unakafov is a scientist at BitifEye since 2019. He is leading test & measurement research projects. Since 2003 he develops mathematical methods and algorithms for signal processing, analysis, and modelling of non-linear systems, biomedical engineering, etc. He received his Ph.D. degree in 2015, www.BitifEye.com

Ransom Stephens is Consulting Senior Scientist at BitifEye, where he provides insights when he can and wisecracks when he can't. Dr. Stephens helps engineers advance to higher data rates with targeted training, including the classes he teaches at Oxford University's Department of Continuing Education, www.RansomsNotes.com

Nithin Parameshwaraiah worked as an Application Expert at BitifEye until the end of 2021 and is now working at Advantest Europe GmbH. He obtained his Master's degree from the Technical University of Darmstadt. He is experienced in supporting and developing test automation software for testing high-speed digital interfaces, www.BitifEye.com

Julien Henaut is the Chief Operating Officer and R&D manager at BitifEye. He is currently leading a team of 20 engineers to establish a new reference and push the limits of high-speed digital receiver testing software in more than 15 of the most advanced digital bus standards, www.BitifEye.com

Afshin Attarzadeh is a Senior Software Engineer at BitifEye, which he joined in 2017. He holds an MSc in Information Technology in Embedded Systems Engineering from the University of Stuttgart and has worked actively in developing industrial controlling devices and the associated software interfaces since 2007, www.BitifEye.com

Valentina A. Unakafova is a scientist at BitifEye. Since 2009 she combined work in industry and research working in the areas of signal processing, mathematical modelling, nonlinear systems analysis, optimization, and algorithms development. She received her Ph.D. degree in mathematics in 2015, www.BitifEye.com

Philipp Schleicher is currently a Senior Software Engineer in BitifEye. After receiving his MSc in Information and System Technology from the University of Ulm, he worked actively in software development, covering fields as various as embedded software, desktop applications, and database-oriented web services, before joining BitifEye in 2020, www.BitifEye.com

1 Introduction

Arbitrary Waveform Generators (AWGs) are instruments that generate signals from numeric waveform descriptions stored in the AWG's memory. Selected samples of the waveform are sent to the AWG's Digital to Analog Converter (DAC) and then, with appropriate filtering and signal conditioning, are output as an analog waveform.

Modern AWGs have high sampling rate (50 GSa/s and above), high bandwidth (>15 GHz), and deep memory (>1 GSa). When a junior engineer encounters such an advanced AWG, it can seem like magic: Just calculate or even draw a waveform with all the desired properties – jitter, noise, crosstalk – and literally connect the dots, plug it in, and press play. But once the engineer encounters the devilish details, things can go sour; from magic to impossible in a few hours.

Example 1. Assume that we want to emulate a 7 GBd signal, for instance to test a receiver. Using an AWG with high sampling rate (65 GSa/s), decent amplitude resolution (8 bits), and deep memory (2 GSa), we generate a signal in a few simple steps and verify the eye diagram with an oscilloscope immediately at the AWG output (see Figure 1). When we see a signal like that shown in Figure 1, we may reasonably be disappointed. These AWGs are either not so good or not so easy-to-use!

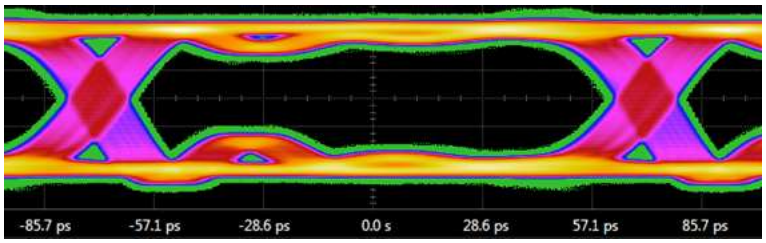

Figure 1. Eye diagram for a 7 GHz signal improperly generated with an AWG. The eye is distorted (note the prominent ripples: outward at –57.1 ps and inward at about –30 ps) and closed by about 15% even before entering the channel.

The insufficient quality of the waveform in this specific case is caused by improper choice of sampling rate and transition times (we address these topics in Sections 3.1 and 4.2). To create a high fidelity signal with an AWG, one must understand how signals are processed.

We have been using deep memory, high bandwidth AWGs in automated testing for years now and would like to share what we have learned.

Our paper concentrates on the diverse abilities of AWGs to generate the signals required for compliance to standards such as HDMI 2.1, MIPI C-PHY, MIPI D-PHY, and MIPI A-PHY. We will share our frustrations and show how we overcame them: cases where AWGs provide the best, fastest, and cheapest solution, as well as cases where AWGs are simply not the right tool for the job.

2 AWG characteristics

2.1 Vertical (amplitude) resolution

The Digital to Analog Convertor (DAC) is the key element of the AWG; taken simply, an AWG is an extended high-speed DAC with a deep memory. The DAC resolution specifies the minimum amplitude step between adjacent samples. AWG resolution is described in

terms of the number of bits. A 6-bit AWG resolution provides 64 amplitude steps, while a 4-bit AWG resolution allows only 16 amplitude steps.

Example 2. Consider the signals shown in Figure 2. An ideal sinusoid (black curve) is represented in discrete steps with 4-bits (blue curve) and 6-bits (red curve) of resolution. The lower graph in Figure 2 shows the deviations of the discretized versions of the signal from the ideal. These deviations, known as *quantization noise*, are noticeably higher for the 4-bit resolution. Obviously, higher AWG resolutions allow the generation of better analog signals.

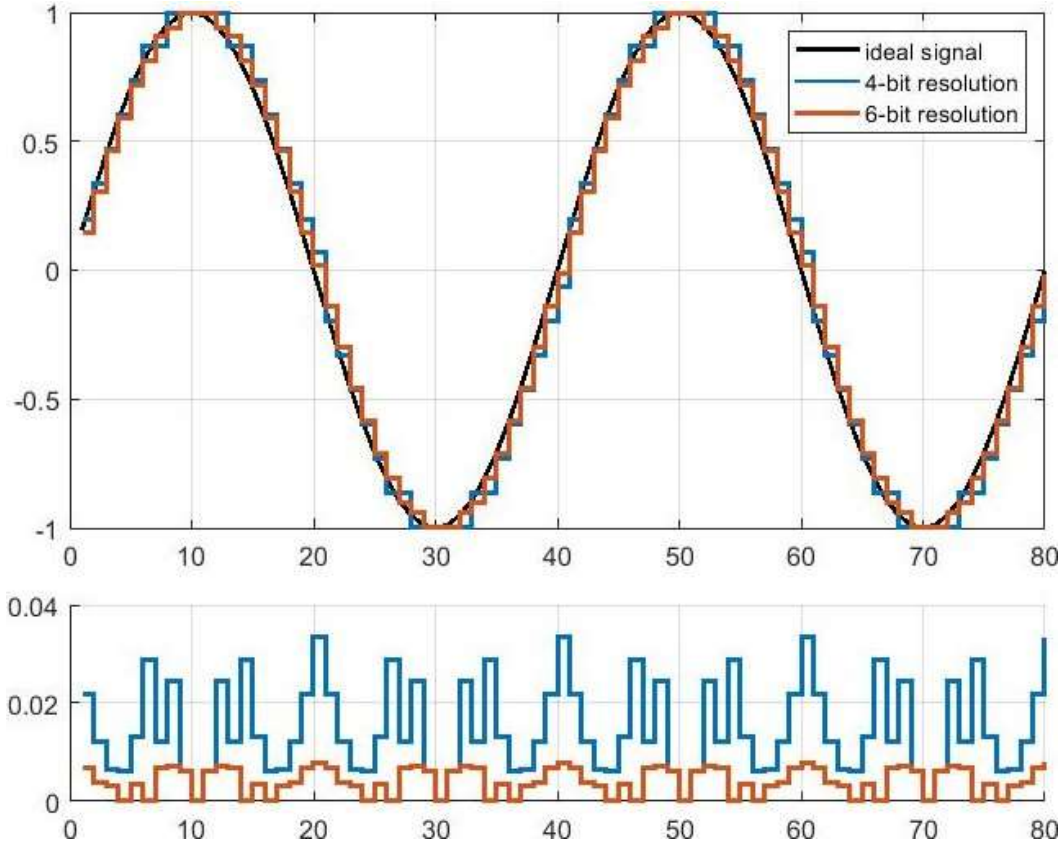

Figure 2. The discretized signals at the AWG output deviate from the ideal sine signal (top). Absolute values of this deviation are shown at the bottom.

The ratio of signal power to quantization noise power (SQR) is given by

$$\mathrm{SQR} = 1.76 + 6.02\ N_{\mathrm{bits}}\ (dB).$$

For example, a DAC with $N_{\mathrm{bits}} = 8$ bits of resolution exhibits an SQR of 49.92 dB.

The relevance of the DAC resolution for the generation of high-speed signals can be easily overlooked. Indeed, when we generate a Non-Return-to-Zero (NRZ) signal with just two levels, what difference does it make whether our resolution is 1/256 or 1/16 of the full scale?

It makes a lot of difference as soon as we do not generate an ideal digital signal (for which we do not need an AWG at all!). Here is an example, see Section 4.2 for further details:

Example 3. To illustrate the effect of AWG resolution in the frequency domain, we generate a combination of complex noises defined by MIPI A-PHY specification 1.0, Chapters 7.3-7.4. Specifically, we combine Alien Cable Bundle related Crosstalk Broadband Noise (defined by its spectral density for frequencies from 0.2 MHz to 4 GHz) with Fast Transient Noise (a high-amplitude decaying sawtooth with frequency 40 MHz). One can see (Figure 3)

how a decrease of DAC resolution worsens the Power Spectrum Density (PSD) of the ideal signal: When we take

- 8-bit resolution, we lose PSD precision above 250 MHz,
- 4-bit resolution, we lose PSD precision already starting from 10 MHz.

This happens because we combined noises with different amplitude ranges – in this case, the resolution effect is especially prominent. Taken by itself, each noise can be reliably generated separately with an 8-bit DAC.

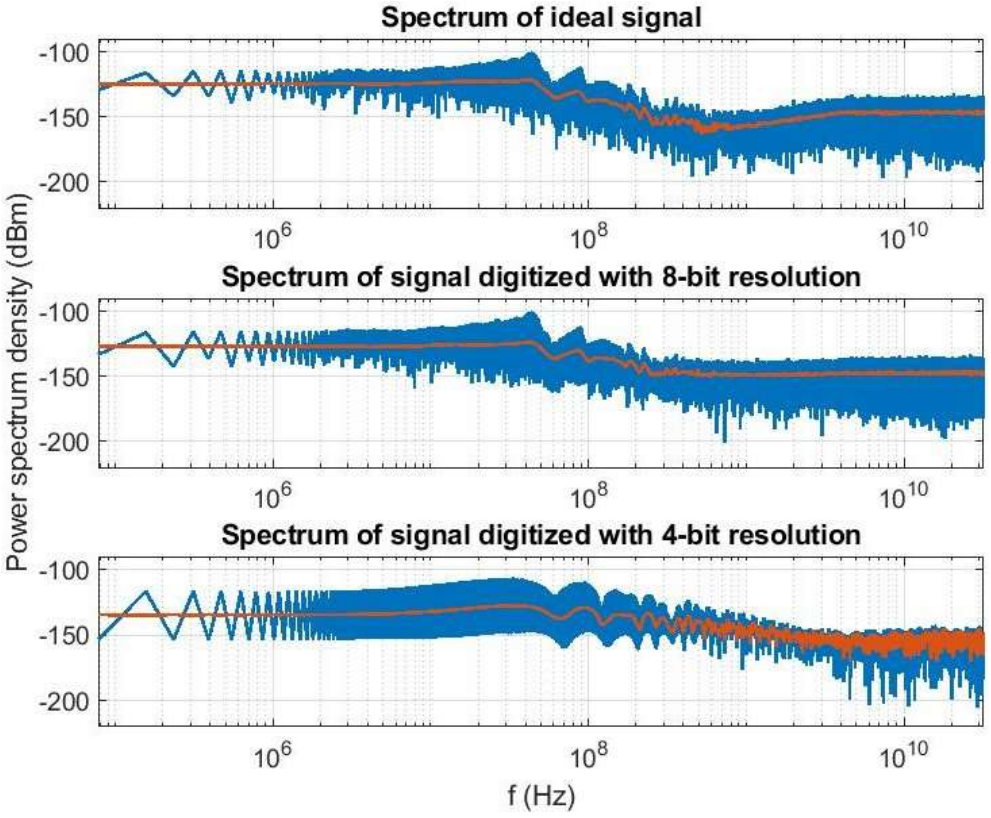

Figure 3. Effect of the AWG resolution on the generated signal.

AWGs used for testing modern high-speed communications typically have 6–10 bits of resolution. Higher resolution is unusual since quantization noise is just one of several noise components at the generator's output. A trade-off between resolution and sampling rate is introduced by thermal noise. This noise is caused by the random thermal motion of charges within the AWG, which generates power proportional to the signal bandwidth. To achieve a meaningful compromise, AWG bandwidth and resolution are selected coherently, such that quantization noise is just slightly higher than thermal noise. Further increase of DAC resolution does not improve the quality of the output due to the thermal noise. For example, the deepest memory AWG from Keysight, M8195A, has 8-bit resolution and the bandwidth of 25 GHz, while its competitor from Tektronix, AWG70000B, has 10-bit resolution, but the bandwidth of 15 GHz. For more details about DACs used in modern AWGs, see (Schmidt et al. 2017).

Note that amplitude resolution in AWGs is defined relative to the maximum (full scale) output amplitude, which is determined by the output amplifier stage. Thus, if we have an 8-bit amplitude resolution and set output amplitude to 1.28 V, absolute resolution is 0.005 V. If

we decrease the amplitude by a factor of five, to 0.256 V, the absolute resolution decreases, respectively, to 0.001 V. Of course, noise and nonlinearity prevent us from having arbitrarily fine absolute resolution; therefore the maximum AWG output amplitude is bounded from below.

2.2 Horizontal resolution (sampling rate)

The horizontal resolution of the AWG is determined by its maximum sampling rate, i.e., the maximum number of samples the AWG can generate per second (Sa/s). With high-speed serial communications baud rates of 32 GBd in PCIe 6 and 56 GBd in OIF-CEI 5.0, high sampling rate is crucial. Modern deep memory AWGs typically have maximum sampling rates of 50–70 GSa/s. Another important AWG parameter related to sampling rate is its analog bandwidth, which determines the highest sinusoidal wave frequency that an AWG can produce with less than 3 dB of loss. A typical bandwidth of modern deep-memory AWGs is 15–25 GHz.

Since AWG continuously takes data from the memory, it does not make sense to have sampling intervals shorter than the memory access time. Therefore, although having both high speed and deep memory is important, combining them both in the same single AWG is a serious design problem. Current fast AWGs have limited memory, 1 MSa or less, since the more memory a databank has, the more difficult effective access to it is. For instance, Keysight's fastest AWG, M8199A, has a maximum sampling rate of 256 GSa/s and only 1 MSa of memory.

Conversely, the deepest memory AWG from Keysight, M8195A, has 16 GSa of memory but full speed limited to 65 GSa/s. Even this comes with a limitation: to achieve the maximum sampling rate, the M8195A "interleaves" the outputs of its four DACs; it takes every fourth sample from AWG memory and inserts it into the output stream. Such interleaving compensates the slow memory access but limits the AWG's number of channels to one for full speed, two for half, and four for a quarter of the full speed. Similarly, the Tektronix AWG70000B with its 32 GSa of memory can use the full speed of 50 GSa/s only when its two DAC outputs are interleaved.

In some cases, it is beneficial to use an AWG sampling rate different from the maximum. For example, it is often convenient to use a sampling rate that is a multiple of the signal baud rate (more on this in Section 3.1); for example, to emulate a 17 GBd signal, it can be more convenient to use a sampling rate of 51 GSa/s (3 samples per symbol) than 64 (3.7647 samples per symbol). Unfortunately, this is not always possible since the AWG clock typically cannot operate at arbitrary frequencies. For example, the Keysight AWG M8195A supports sampling rates of 53.76 GSa/s to 65 GSa/s.

2.3 AWG memory organization

To achieve extremely high baud rates, an AWG should have an effective memory organization (see (Keysight, 2015), Section 3 for details). The memory of modern high-performance AWGs is usually divided into segments that can be dynamically assigned to waveforms. Multiple waveforms can be stored in different segments and users can switch from one to another in a very short period of time, by selecting them from a lookup table. Switching from one segment to another is fast, often at the sampling rate, allowing seamless transmission of long waveforms.

Importantly, modern AWGs offer high flexibility in managing segment sequences, allowing users to create long and complex waveforms (which is important for testing modern high-speed serial interfaces with their complex signaling and high bit-error-rate) that can be adjusted in real time. The waveform may consist of multiple segments, some of which are

looped (repeated either finitely or infinitely). Dynamic sequencing is a feature that makes AWGs capable of adaptive behavior, which is especially valuable when testing modern high-speed serial interfaces that require receivers to control the transmitter in so-called link training (important part of HDMI, OIF-CEI, PCIe, USB, and other standards).

The principal memory limitations for waveform generation in most existing AWGs are record length (RL) and granularity (G). The latter limitation means that the AWG's waveform should fully occupy all its segments. In other words, the record length should be divisible by the granularity, i.e., RL/G should be an integer number. We will see in Section 3.2 that when granularity is high (e.g., G = 256 samples), this condition is quite important.

3 Characteristics of an AWG waveform

3.1 Number of samples per unit interval

The number of Samples per Unit Interval (SpUI) characterizes the time resolution of the generated waveform. It is given by the ratio of the AWG sampling rate to the signal baud rate:

$$\text{SpUI} = \frac{SR_{AWG}}{BR_{signal}}.$$

Multiple contradicting factors influence SpUI selection. However, the principal factor is the reduction of waveform quality at lower SpUI values, as illustrated in Figure 4.

Rule of thumb: The higher the SpUI, the better the waveform quality.

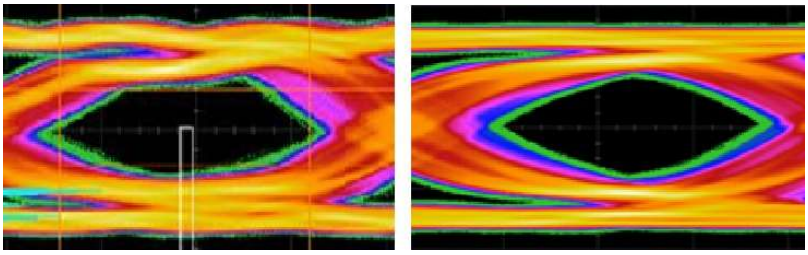

Figure 4. Eye diagrams for a signal at 18 Gbd baud rate. Using SpUI = 3 and sampling rate = 54 GSa/s (left) results in a highly distorted eye. Using SpUI = 3.5 and sampling rate = 63 GSa/s (right) drastically improves the eye shape. Interestingly, the right eye width does not improve compared to the left due to transition jitter, see details in Example 4 below.

We will cover more factors influencing SpUI choice in Sections 3.2 and 4.1.

3.2 Waveform length and record length

The longer the waveform, the more memory it fills. This can be important for receiver testing applications, where many long waveforms are needed. One should bear in mind that the actual memory required to store a waveform (record length, RL) may be much longer than the number of waveform samples and eventually exceed the AWG memory limitations. To explain this, consider an example.

Example 4. Assume that we need to transmit a PRBS13 pattern with baud rate 18 GBd. The pattern has length 2^{13}–1 = 8191. To get optimal signal quality, we use $\text{SpUI} = 3.5$, achieving

a sampling rate of 63 GSa/s. The length of the waveform is then $8191 \cdot 3.5$. Since we cannot store half-samples, we need to repeat the waveform twice, hence we get $8191 \cdot 7$. Now we need to match the granularity requirement (length divisible by 256), which ultimately results in $RL = 256 \cdot 8191 \cdot 7$. Almost 14 MSa of memory is required to generate a relatively simple pattern!

This discouraging result is due to the fact that the waveform should be repeated 512 times to fit into the memory. Two factors contribute to this:

- pattern length is an odd number,
- SpUI is a fractional number.

The first factor is out of our control, as the length of standard PRBS test patterns is always an odd number. This is also the case when the test pattern is the scrambler output. The second factor (fractional SpUI), however, can be alleviated by selecting a proper number of samples per unit interval.

Rule of thumb: To optimize (minimize) total waveform length, use SpUI a power of 2, an even number, or at least an integer.

In Example 4, the only way to optimize memory usage is to use SpUI = 3, which results in a low sampling rate of 54 GSa/s (see Example 1), which does not make much sense since the record length is only half as long. However, if we consider the generation of a signal with baud rate of 14 GBd, we have a choice:

- use SpUI = 4.5 and optimize waveform quality or
- use SpUI = 4 and optimize waveform length (in this case by a factor of 8) without degrading the quality too much.

3.3 Transition time

Transition Time (TT), also known as rise/fall time, is the time it takes for a signal to transition from one value to another. One may expect that, for the signal generated by an AWG, the transition between signal levels should happen within one Sampling Interval (SI) of the AWG, but such short transition times do not match the physically available AWG bandwidth. For realistic waveforms, transitions are shaped by classical functions, Gaussian or raised cosine (Gentile, 2007). Transition times are often defined as the duration from 10% to 90% of the peak-to-peak amplitude. Transition time is a fundamentally important characteristic of a signal since every circuit needs some time to respond to a fast-changing input. For example, in a pure RC circuit, the output transition time (10% to 90%) is approximately equal to $2.2 \cdot R \cdot C$, where R is the resistance and C the capacitance of the circuit.

When an AWG is used to generate a high-speed signal, SpUI is usually low, due to the limited sampling rates of AWGs. This complicates transition shaping since with few samples per unit interval, the transition is too short. In particular, when transition time $TT < 2 \cdot SI$, no proper control over the transition is possible (Tabor Electronics, 2021), as the waveform shape is effectively trapezoidal regardless of transition sampling (see Section 4.2 for details). Generating a proper waveform is especially complicated for fractional SpUI values, where sampling point locations in the transition depend on the symbol position within the waveform, as shown in Figure 5.

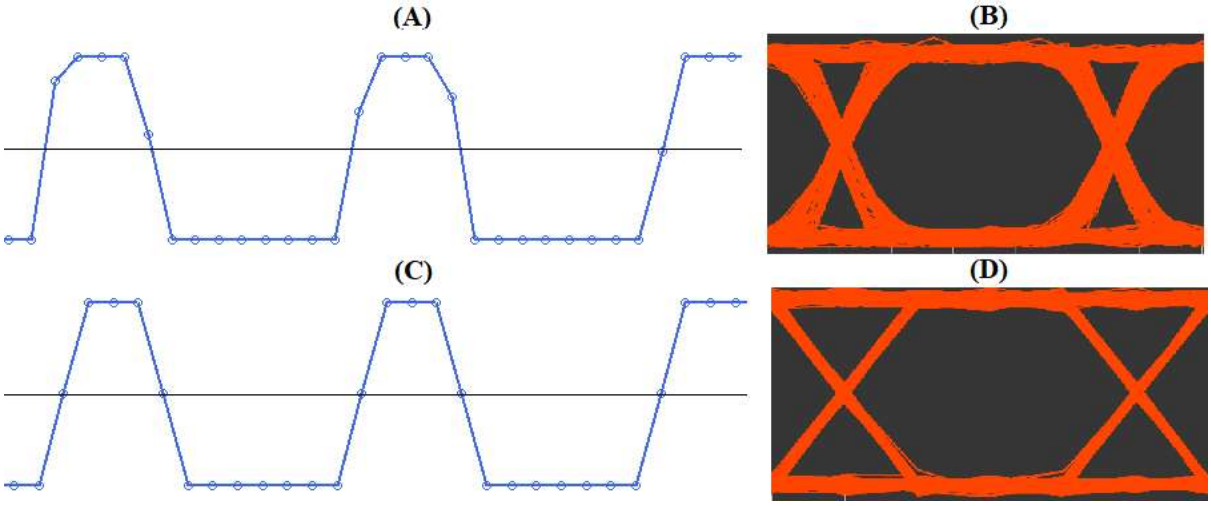

Figure 5. Effect of SpUI on the transition for two waveforms. On the left (A, C) we show a part of each waveform and on the right (B, D) the corresponding eye diagram. (A, B) For fractional SpUI = 4.39, sampling points within transitions vary for different symbols, causing the effective shape and transition times to vary, which results in a more closed eye. (C, D) For integer SpUI = 4, all sampling points have the same positions within transitions (in this example, transition time is equal to two sampling intervals, with sampling points at the ends and in the middle of transitions), resulting in a wider eye opening.

The variability of sampling positions in transitions and, as a consequence, the variability of the transition shapes for fractional SpUI, results in a special "transition" jitter, as demonstrated in Example 5 and Figure 6.

Example 5: Transition jitter for fractional SpUI. Consider an NRZ signal at 7 GBd generated by AWGs using sampling rates of 65 GSa/s (SpUI $= 65/7 \approx 9.3$) and 56 GSa/s (SpUI $= 8$). Figure 6 shows that integer SpUI results in much wider eye opening.

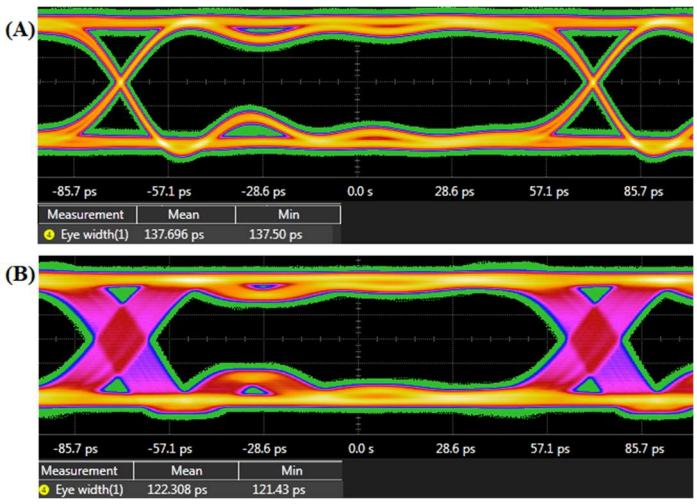

Figure 6. Eye diagrams for a 7 GBd signal generated with an integer SpUI = 8 samples/UI (A) and with a fractional SpUI = 65/7 samples/UI (B). Note that the fractional SpUI increases jitter by 16 ps, which in this case is more than 0.1 UI.

Rule of thumb: when the quality of the waveform is decisive, use integer SpUI to prevent unwanted transition jitter.

4 Challenges in generating high-speed signals with AWGs

4.1 Adding impairments to the signal

To verify a receiver's capabilities or to test its compliance, it is not enough to generate a clean signal: we need to inject into the signal controllable impairments (jitter, noise, inter-symbol interference, crosstalk, etc.). "Controllable" means that the injected amount of jitter or noise is reasonably close to the target amount defined by the specification; otherwise the impaired signal is not useful.

Especially challenging is adding jitter. In modern serial communication standards, UI is typically 20–100 ps, which requires jitter to be defined with at least picosecond precision. At first sight, it seems impossible to do this with an AWG having horizontal resolution (sampling interval) of at most 10 ps.

To solve this problem, we make use of the transition shaping described in Section 3.3. The jitter shifts the center of the transition (with required precision), and then the values of the samples are computed for this shifted transition. The following example illustrates our approach.

Example 6. Consider a waveform consisting of repeated 01, with a UI of 93 ps (i.e., baud rate of 10.7 GBd) and transition time of 31 ps. The AWG's sampling rate is set to 64.2 GSa/s, thus SpUI = 6 and the sampling interval is 15.5 ps. The original waveform is shown in Figure 7 in blue. Let us add jitter to this signal, shifting some 0-to-1 transition by 3.1 ps (one fifth of the sampling interval!). To do this, we simply shift the transition center by 3.1 ps and compute sampling points on this shifted transition. The jittered waveform is plotted in red and is exactly what we were looking for.

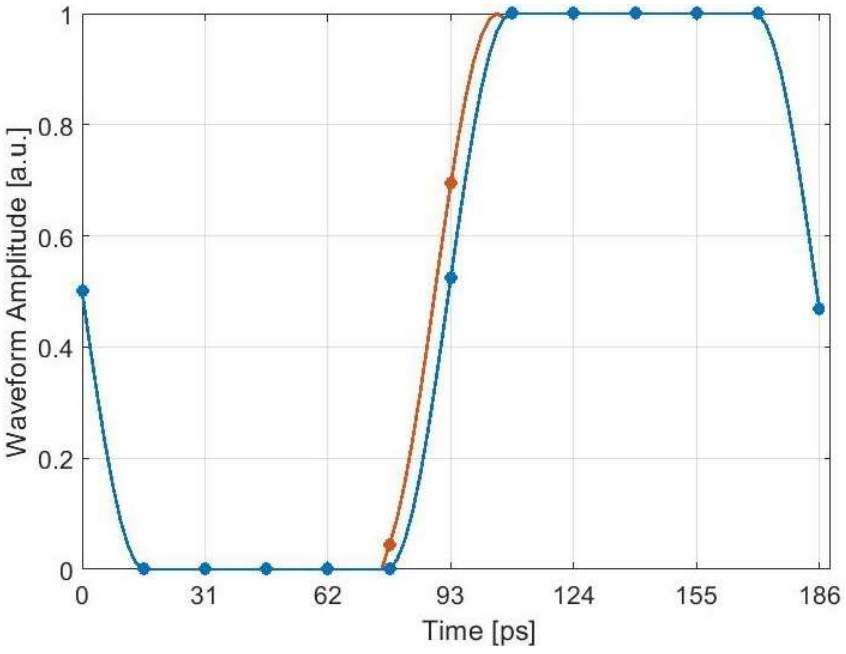

Figure 7. Waveform without impairments (blue) and with small jitter (red). To inject jitter, we shift the transition center (horizontally), resulting in samples on the transition being recalculated (hence shifted vertically).

Note that we are compensating for insufficient time resolution by the "excessive" vertical resolution. Here we again see that a high-resolution DAC (8-bit and above) is not at all a luxury! More generally, we suggest the following rule of thumb linking DAC resolution with SpUI:

Rule of thumb: To generate a decent PAM-X signal with controllable impairments, the number of samples per unit interval (SpUI) and number of bits N_{bits} of the AWG vertical resolution should satisfy the following inequality:

$$\mathbf{SpUI} \cdot 2^{N_{\text{bits}}-8} > \mathbf{X},$$

where X is the number of levels in the PAM-X signal.
For example, in the case of 8-bit AWG we recommend SpUI > 2 for generating an NRZ (PAM2) signal and SpUI > 4 for PAM4.

Adding impairment other than jitter is more straightforward. Inter-symbol interference and crosstalk are applied by convolving the waveforms with the kernels computed from S-parameter files, while noise and interference are simply added to the values of samples. Note that AWG is capable of generating quite complex noises, such as those required by automotive standards such as MIPI A-PHY (see Example 3).

At the end of this section, let us describe how the waveform representation is changed in the process of adding impairments.

1. We start with a *data sequence* consisting of the symbols to be transmitted. For NRZ signaling these symbols are bits, while for PAM4 these symbols are numbers from 0 to 3, etc. If the signal should be de-emphasized using Transmitter Feed-Forward Equalization (TxFFE), we can apply the TxFFE filter to the data sequence.

2. The data sequence is converted to a *non-sampled waveform*. This waveform is represented by two sequences: a sequence of signal levels l_k, and a sequence of transition centers t_k. The lengths of these two sequences depend on the number of changes in the data sequence, e.g., data sequence 0011 can be represented by two voltage levels ($l_1 = 0, l_2 = 1$) and one transition center ($t_1 = 2$ UI). Jitter is added to the non-sampled waveform as described in Example 6.

3. Finally, a *sampled waveform* is computed from the non-sampled one. This waveform is just a sequence of sample values, as should be stored in the AWG memory. Noise, inter-symbol interference, crosstalk etc. are added to the sampled waveform.

4.2 Selecting transition time to counteract ripples in the waveform

As we have already mentioned in Section 3.3, too short transition times $TT < 3 \cdot SI$ turn transition shaping into a problem. In this case, the waveform is nearly trapezoidal, regardless of the transition-shaping scheme (see Figure 8). Trapezoidal signals with high skew are harmonic-rich and severely distorted when generated by an AWG with low SpUI.

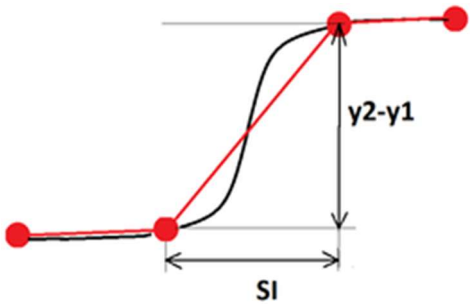

Figure 8. When transition time is short relative to the sampling interval ($TT < 3 \cdot SI$), the AWG renders short almost-linear transitions (red line), regardless of transition-shaping (black curve).

Example 7. Consider three MIPI C-PHY signals at a baud rate of 3.5 GBd generated by an AWG with sampling rate 14 GSa/s (SpUI = 4). The transition times of the signals are 70 ps ($TT = \frac{1}{4}$UI = 1 SI), 95 ps ($TT = \frac{1}{3}$UI = $\frac{4}{3}$ SI), and 142 ps ($TT = \frac{1}{2}$UI = 2 SI). See Figure 9. Note that too short transition time result in waveform distortion (Figure 9A), while excessive transition time causes eye closure (Figure 9C). In this Section we try to find a compromise between these two extremes.

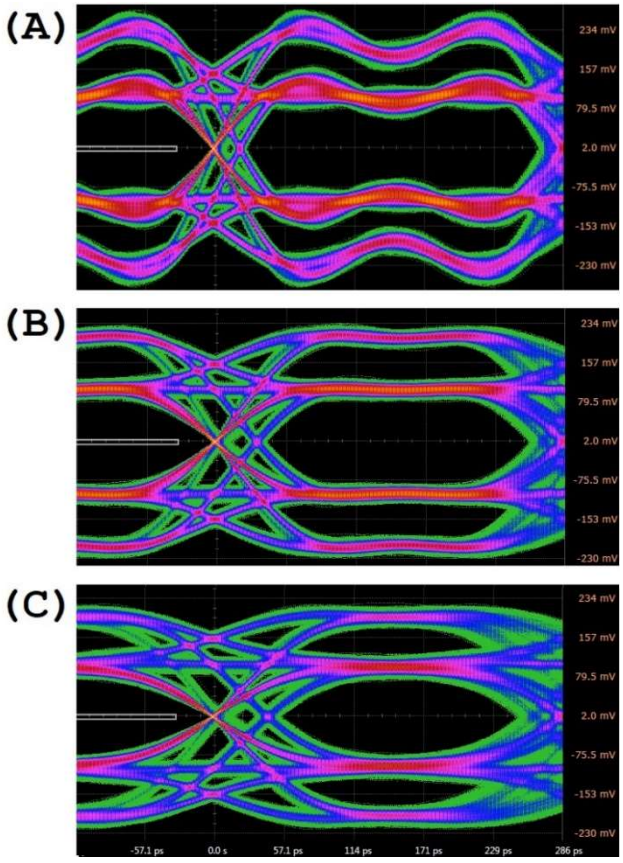

Figure 9. MIPI C-PHY eye strongly depends on the transition time. (A) When the transition time is too short ($TT = \frac{1}{4}$UI = 1 SI), one can see prominent ripples in the signal. (B) As transition time increases to $TT = \frac{1}{3}$UI = $\frac{4}{3}$ SI, the ripples decrease. (C) Further increase of transition time ($TT = \frac{1}{2}$UI = 2 SI) results in eye closure since the signal does not reach its maximum level.

4.2.1 Approximating a waveform with short transition times by a trapezoidal waveform

Here we consider a waveform with two levels, –1 and 1, with Gaussian-shaped transition and short 10%-to-90% transition times $TT < 3 \cdot SI$. We approximate this waveform by a trapezoidal waveform and use this approximation to estimate the amplitudes of the ripples depending on the value of TT. After we have learned the relationship between the ripples and TT, we compute the value of TT resulting in negligible ripples.

The steepest transition takes place when the center of a transition is symmetric relative to the two sampling points. In this case, most of the transition occurs in the sampling interval between these two sampling points (see Figure 8). Their *x*-coordinates, relative to the transition start (10% of the transition), are given by $x_{1,2} = \frac{1}{2}TT \mp \frac{1}{2} \cdot SI$. Using the Gaussian law for the transitions, we compute the *y*-coordinates of the sampling points by the following equation:

$$y_{1,2} = \mathrm{erf}\left(1.8\frac{x_{1,2}}{TT} - 0.9\right) = \mp\mathrm{erf}\left(0.9\frac{SI}{TT}\right),$$

where erf is the Gaussian error function. To assess the amplitude of the ripples, we replace our trapezoid-like waveform by an actual trapezoid of equal skew, see Figure 10. To simplify the frequency analysis, we assume that our trapezoid is periodic with period one.

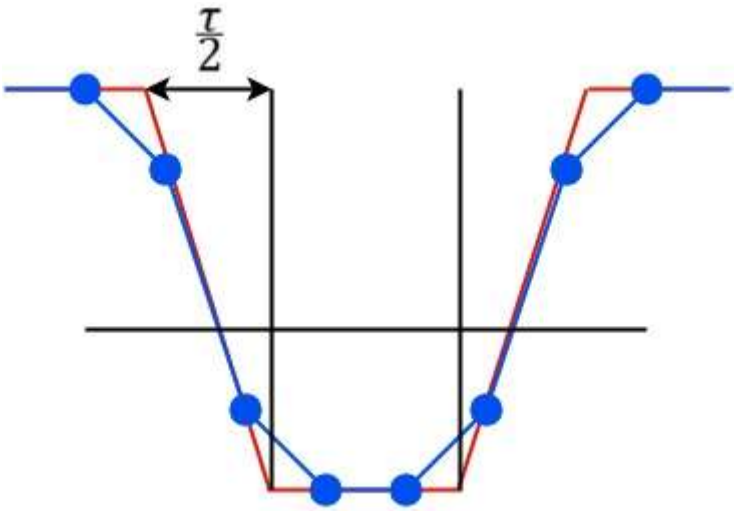

Figure 10. When the waveform is replaced by a trapezoid of equal skew, the transition time of the resulting waveform, τ/2, is less than the original transition time.

The transition time of the resulting trapezoidal waveform $\tau/2$ is then computed from the proportion $\frac{\tau}{1-(-1)} = \frac{SI}{y_2-y_1}$. Thus

$$\tau = \frac{SI}{\mathrm{erf}(0.9\frac{SI}{TT})} \qquad (1)$$

In the frequency domain, a trapezoidal signal with transition duration $\tau/2$ is represented by the sum of the Discrete Fourier Transform (DFT) components (*k*-th component amplitude, c_k, and phase, φ_k):

$$S(t,\tau) = \sum_{k=1}^{2n} \cos\left(2\pi k\left(t+\frac{\tau}{4}+\frac{1}{4}\right)+\varphi_k\right) c_k$$

$$= \sum_{k=1}^{2n} \cos\left(2\pi k\left(t+\frac{\tau}{4}+\frac{1}{4}\right) - 2\pi k\left(\frac{\tau}{4}+\frac{1}{4}\right)\right) \cdot \frac{\sin\left(\pi\frac{k}{2}\right)}{\pi\frac{k}{2}} \cdot \frac{\sin\left(\pi k\frac{\tau}{2}\right)}{\pi k\frac{\tau}{2}}$$

$$= \sum_{i=1}^{n} \cos(2\pi(2i-1)t) \cdot \frac{(-1)^{i+1}\cdot 2}{\pi(2i-1)} \cdot \frac{\sin\left(\pi(2i-1)\frac{\tau}{2}\right)}{\pi(2i-1)\frac{\tau}{2}}, \quad (2)$$

where n is the number of available (odd) harmonics. The number of available odd harmonics n is the largest integer less than or equal to $\frac{\mathrm{SpUI}+1}{2}$. Thus, for $3 \leq \mathrm{SpUI} < 5$, we have $n = 2$ odd harmonics, for $5 \leq \mathrm{SpUI} < 7$, $n = 3$ and so on.

Depending on the transition time, different harmonics become prominent (see Figure 11). The more prominent odd harmonics beyond the bandwidth, the stronger the mismatch between the ideal trapezoidal and its low-pass-filtered version produced by AWG. This mismatch is exactly what causes the ripples in AWG signal (see Figure 12).

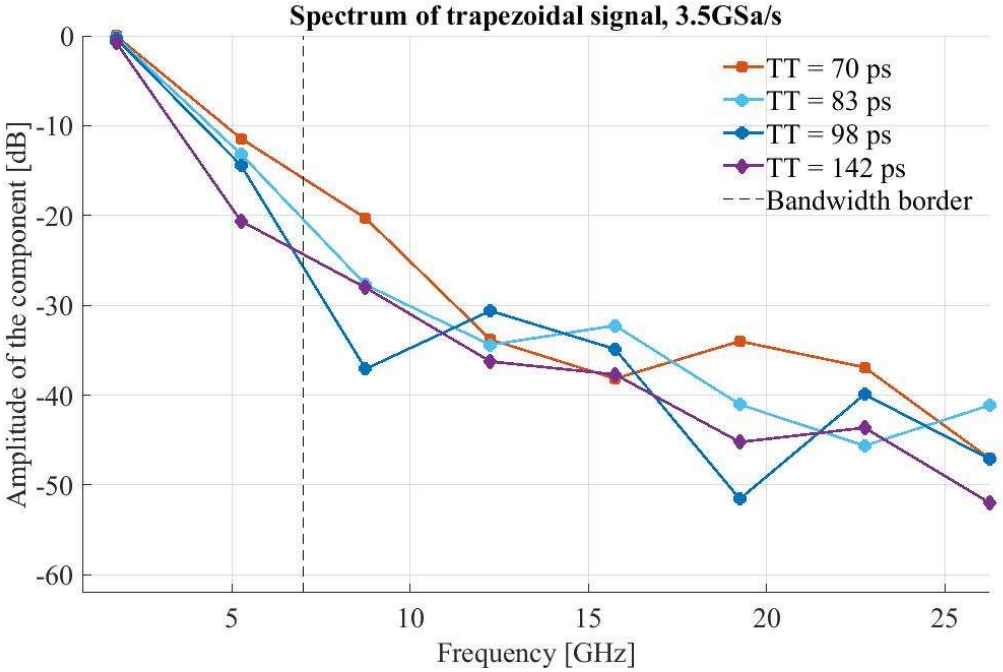

Figure 11. Depending on the transition time, different harmonics become prominent. Notice that the blue curve (for the transition time $TT = 98$ ps) gives the lowest error when only two (odd) harmonics are available, since for it the remaining harmonics have lower amplitudes. Were three odd harmonics available, the violet curve (for the transition time $TT = 142$ ps) would provide the best outcome.

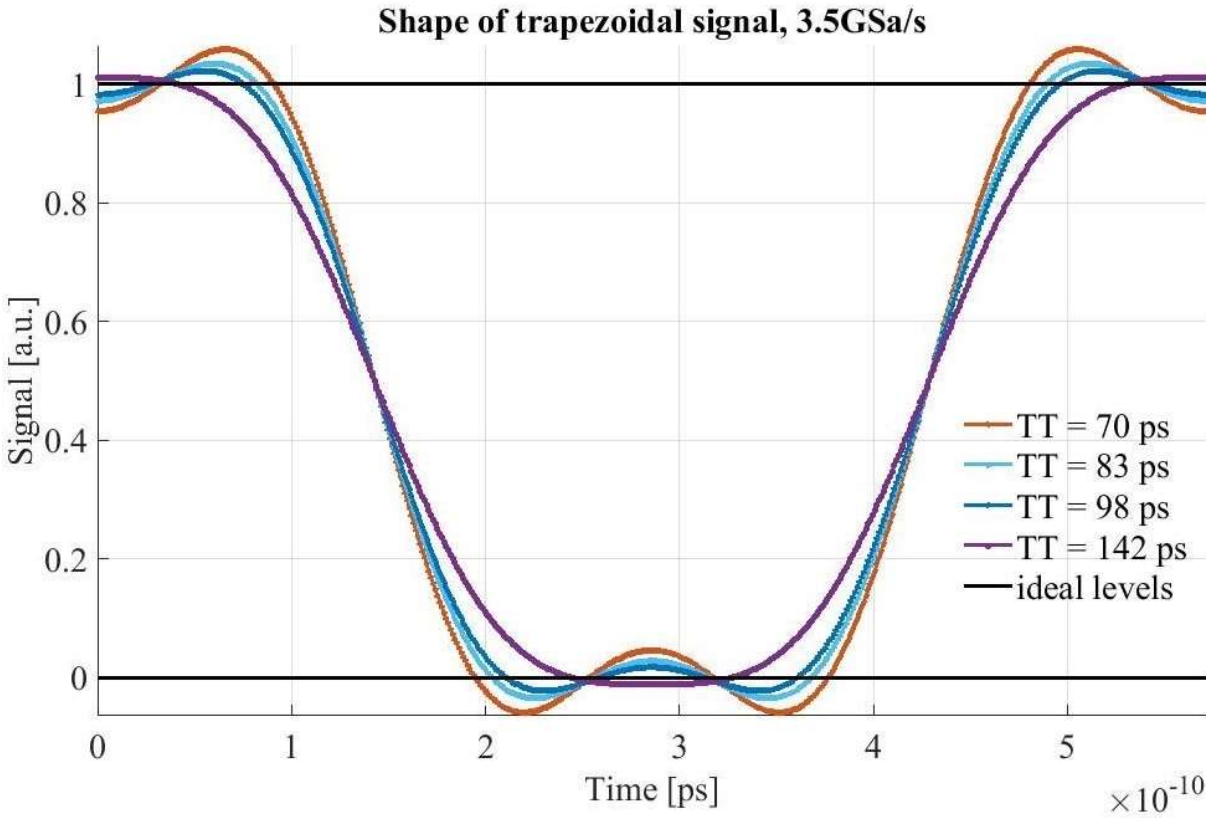

Figure 12. As one may expect from Figure 11, the blue curve (for TT = 98 ps) provides the lowest deviations from the ideal signal level. This is in line with the experimental results in Figure 9.

Combining Eq. (1) and (2) we can estimate the ripple amplitude depending on the transition time TT:

$$R(TT) = \max_t \left| 1 - S\left(t, \frac{SI}{\operatorname{erf}(0.9\frac{SI}{TT})} \right) \right|,$$

where $S\left(t, \frac{SI}{\operatorname{erf}(0.9\frac{SI}{TT})} \right)$ is the signal computed from a limited number of the DFT components according to Eq. (2) for $\tau = \frac{SI}{\operatorname{erf}(0.9\frac{SI}{TT})}$.

4.2.2 Closed-form equation for the optimal transition time

We can also compute the optimal transition time τ using a proxy for the ripple amplitude, the deviation of the waveform from the ideal level 1 at the initial time $t = 0$:

$$\text{ripple}(\tau) = |1 - S(t,\tau)| = \left| 1 + 4\sum_{i=1}^{n} (-1)^i \frac{\sin(\pi(2i-1)\frac{\tau}{2})}{\pi^2(2i-1)^2\frac{\tau}{2}} \right|, \qquad (3)$$

where n is an integer such that $n \leq \frac{\text{SpUI}+1}{2}$.

Figure 13 shows that this ripple amplitude proxy behaves similarly to the ripple amplitude.

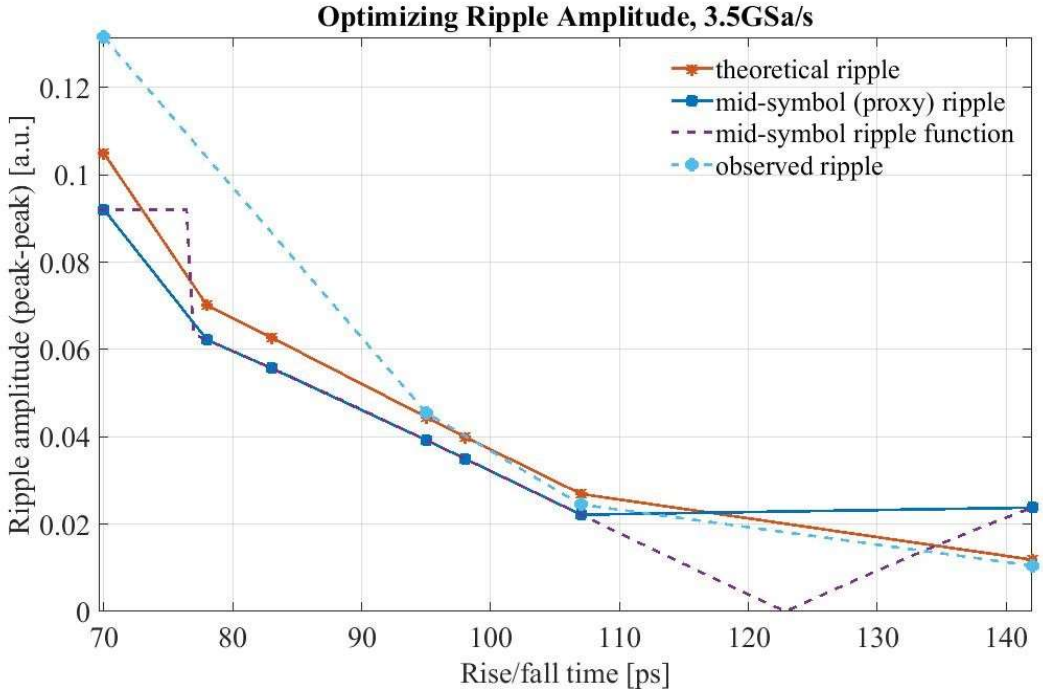

Figure 13. Theoretical ripple amplitude (red curve) computed by Eq. (2) is well correlated with the mid-symbol proxy function (blue curve), which estimates by Eq. (3) the deviation of the signal from the ideal level in the middle of the symbol (violet curve). Notably, the theoretical estimates agree very well with the experimentally measured values of the ripple amplitude (cyan curve).

To get a relatively ripple-free signal, one has to find from Eq. (3) the value of τ for which the $\mathrm{ripple}(\tau)$ is below the required threshold. In particular, one can compute from Eq. (3) the minimal value of τ, giving a ripple-free signal. It depends on SpUI and is given by $\tau = 0.42$ UI for $3 \le \mathrm{SpUI} < 5$, $\tau = 0.27$ UI for $5 \le \mathrm{SpUI} < 7$, etc.

Finally, from Eq. (1) one can compute the optimal transition time TT for the given signal *DataRate*, normalized nominal transition time τ, selected as above, number of Samples per Unit Interval (SpUI), and intrinsic transition time of the instrument $TT_{intrinsic}$ (for instance, for the Keysight AWG M8195A, $TT_{intrinsic} = 28.6$ps):

$$TT > \sqrt{\left(\frac{0.9/\mathrm{SpUI}}{\mathrm{erf}^{-1}(1/(\mathrm{SpUI}\cdot\tau))}\frac{1}{DataRate}\right)^2 + TT_{intrinsic}^2} \qquad (4)$$

For example, for SpUI = 3, $\tau = 0.42$ UI (see above) and $TT > \sqrt{\left(\frac{1.36}{DataRate}\right)^2 + TT_{intrinsic}^2}$.

5 Use cases/applications

In this section, we provide real examples of using AWGs. We consider generating MIPI D-PHY, MIPI C-PHY and HDMI signals.

MIPI D-PHY and MIPI C-PHY belong to the MIPI® Alliance, an organization developing technical specifications for the design of mobile devices such as smartphones, tablets, and laptops. A setup for testing MIPI C-PHY and MIPI D-PHY receivers' physical layer is shown in Figure 14.

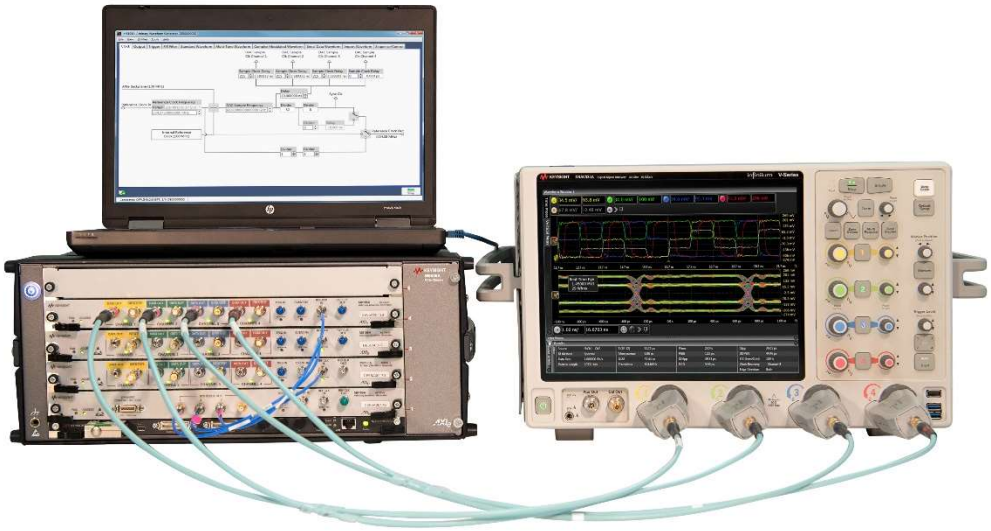

Figure 14. A setup for testing MIPI receivers consists of a controller PC (left, top), several AWG modules in a single block (left, bottom) and a digital oscilloscope (right). The oscilloscope is displaying the differential MIPI C-PHY signal with its four characteristic levels.

The High-Definition Multimedia Interface (HDMI), whose development is guided by the HDMI Forum organization, is intended to transmit audio and digital video data. A setup for HDMI-sink testing is shown in Figure 15.

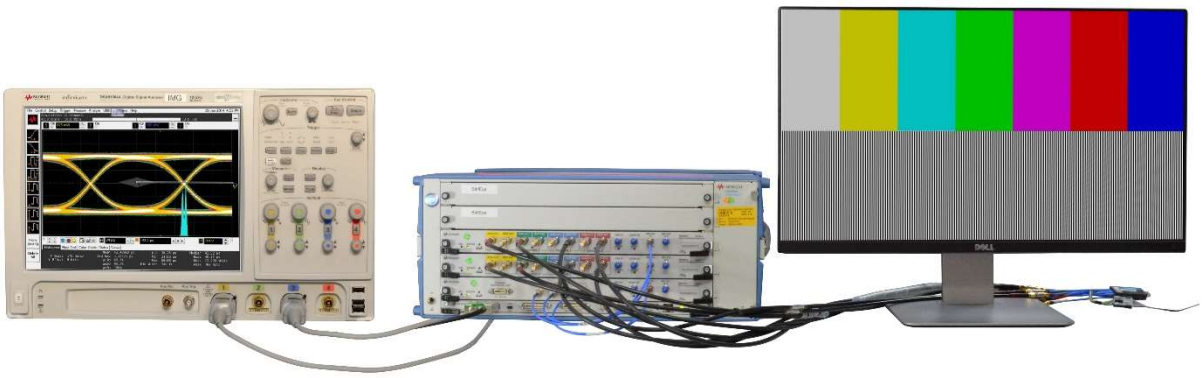

Figure 15. HDMI-sink test setup based on Keysight AWG M8195A.

5.1 MIPI D-PHY

MIPI D-PHY is a high-speed data communications physical layer standard used specifically to connect the camera and display interfaces to a host processor. Unlike most interfaces, MIPI D-PHY employs switching between differential High Speed (HS) and single-ended Low Power (LP) modes in real time. This means that the same physical data paths alternate between two quite different signaling types, with different rates, amplitudes and rise times.

The LP mode is mainly used for controlling purposes, although data transmission is possible. The main data transmission takes place in bursts in the HS mode. There the data is transmitted at the highest specified data rate, 9 Gbps. After the data is transmitted in HS bursts, the interface switches to the LP mode to save power, especially battery capacity, until the next time it is necessary to transmit data.

To emulate the MIPI D-PHY signal, to verify a MIPI D-PHY receiver for example, we have to generate both the low swing differential HS signal and single-ended LP signal in a single transmission, which is not easy.

Using a traditional BERT (Bit Error Ratio Tester), we cannot have different levels for the low- and high-speed sections (although we can generate a signal where some sections of the signal are low speed and others are high speed). An advantage of AWGs here is that they are based on waveforms, not patterns, and can assign different amplitudes to different sequences. The sequences at different amplitudes are combined when the signal is generated.

Consider the number of AWG modules needed to transmit a MIPI D-PHY signal. MIPI D-PHY lanes are differential, formed by two traces or wires – normal and complement. Although each AWG channel can be differential, since the LP mode is single ended we need two channels for each MIPI D-PHY lane (one for normal and another for complement). Since MIPI D-PHY is source synchronous, a clock lane is also required. Therefore, emulation of a single-lane MIPI D-PHY link requires four AWG channels. Using an AWG in a multi-channel mode requires reduction of the sampling rate proportional to the number of channels, since the maximum sampling rate is typically achieved by interleaving several channels (see Section 2.2). Hence, an AWG that nominally provides up to 65 GSa/s, will only deliver 16.25 GSa/s in the 4-channel mode. Even if we use two samples per symbol (which is very low, as explained in Section 3.1), each AWG channel can generate a signal of maximum 8 Gbd baud rate. That means that, in order to generate a 9 Gbd MIPI D-PHY signal, we need at least two AWG modules, one for the data and the other for the clock. This setup provides a sampling rate up to 32.5 GSa/s, which allows the generation of a 9 Gbd signal with SpUI = 3.5.

A MIPI D-PHY link can consist of multiple lanes, which requires even more AWG modules. To ensure that the signals are aligned in time, frequency-locked, a synchronization module is necessary to trigger the AWG modules with deterministic latency.

Using such a ponderous setup comes with additional complications: skew within and between modules must be calibrated. The more lanes, the longer the calibration times, since calibrating each lane requires modifications to the setup.

Once the test is calibrated, generating an accurate MIPI D-PHY signal can still be quite difficult due to signal switching from LP to HS and back and with data coming in bursts. Some MIPI D-PHY tests (e.g., the “hysteresis test”) require signal voltage levels to be changed without pauses in the signal. While BERTs often allow signal levels to be changed without recalculating the signal, for AWGs this is not possible. Instead, we use a special type of dynamic sequencing called “Memory Ping-Pong”. When the waveforms are not known in advance, “Memory Ping-Pong” allows the contents of a waveform segment to be updated during active signal generation. Then the output can be switched glitch-free to the updated segment. So, to implement the MIPI D-PHY hysteresis test, the initial waveform is loaded into a sequence of AWG memory segments (Figure 16). When the test requires the change of voltage, the waveform is calculated for the next step without interrupting the first waveform. The new waveform is saved in a second sequence of memory segments, and when it is time to switch to the second step of the test, the transition is seamless.

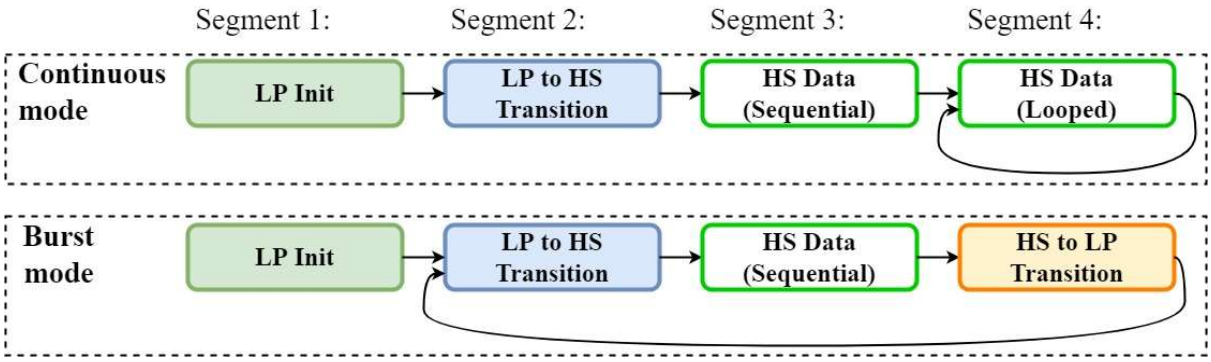

Figure 16. AWG memory segmentation for the MIPI D-PHY continuous and burst modes.

5.2 MIPI C-PHY

MIPI C-PHY is a high-speed data communications physical layer standard widely used in smartphones, the Internet of Things (IoT), and automotive camera and display applications. MIPI C-PHY provides a physical layer for the MIPI Camera Serial Interface 2 (MIPI CSI-2®) and MIPI Display Serial Interface 2 (MIPI DSI-2SM) ecosystems, enabling designers to scale their implementations to support a wide range of higher-resolution image sensors and displays while keeping power consumption low.

Each MIPI C-PHY lane consists of three lines. Like MIPI D-PHY, MIPI C-PHY also has LP and HS modes. The LP mode has two levels and is mostly used for controlling purposes. The HS mode has three levels, where no two lines share the same level at a given moment. The MIPI C-PHY receiver consists of three differential modules, computing differences between lines, thus each received signal has one of four possible states, resulting in an eye diagram with three eyes (see Figure 9). In HS there is always a transition from one symbol to the next, which facilitates clock-data recovery. These specific features of MIPI C-PHY signaling effectively prevent its generation using a BERT. Figure 17 shows a MIPI C-PHY signal consisting of an LP section (left of the red square) and HS section (right of the red square).

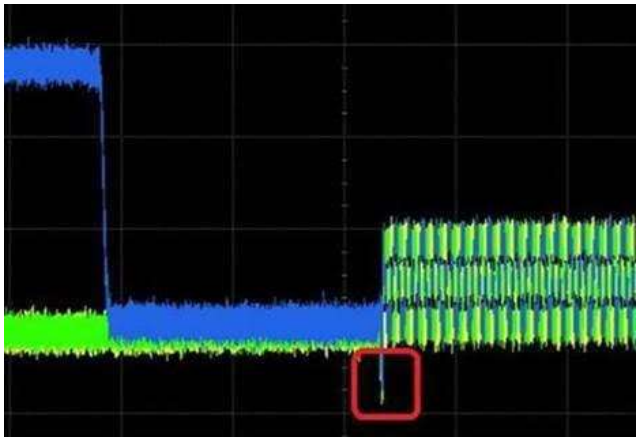

Figure 17. MIPI C-PHY signal: Transition from high-amplitude Low Power (LP) to High Speed (HS) signaling mode. The red square shows a signal artifact resulting from an incorrect adding of impairments at the LP-to-HS transition. Even small mistakes in the waveform generation can cause such artifacts, making the signal non-compliant (incorrect from the specification viewpoint) and unsuitable for testing purposes.

Techniques for generating MIPI C-PHY signals are similar to those described in Section 5.1 for MIPI D-PHY. To generate a one-lane MIPI C-PHY signal with an AWG, one needs three AWG channels. In line with the rule of thumb formulated in Section 4.1, we use sampling rates in the range 13.36–16.25 GSa/s for baud rates up to 4.5 GBd and 26.72–32.5 GSa/s for higher baud rates, up to 9 GBd.

5.3 AWG in interactive receiver testing (HDMI 2.1)

HDMI is a simplex interface for transmitting video and audio data from an HDMI-source device, such as a display controller, to an HDMI-sink, such as a computer monitor, video projector, or digital television.

In contrast to most communication standards, testing the HDMI physical layer is defined in terms of video signals rather than abstract symbol sequences. It includes checks for diverse video modes along with the standard Full HD, 2K and 4K resolution, for example, 3D mode and deep color modes. Although almost all the video modes can be generated with an AWG, it might be complicated to keep waveform calculation short and fit the waveforms into the available AWG memory.

The maximum HDMI 2.1 baud rate is 12 GBd. If the sampling rate of an AWG is limited to 16.25 GSa/s for each channel (which is the case for Keysight AWG M8195A), then at least two AWG channels should be interleaved to generate an HDMI signal of sufficient quality. Since there are four lanes in an HDMI cable (either three data lanes and one clock lane or four data lanes), two M8195A modules are required.

HDMI 2.1 introduces the Fixed Rate Link (FRL) electrical mode, where gap characters are used to achieve the fixed data rate. HDMI 2.1 FRL uses embedded clock, 16b18b encoding, Reed Solomon Parity, etc., but its most important and challenging feature is link training, which requires interaction between sink and source.

Initiating any communication between source and sink requires them to negotiate link properties – such as data rate, number of lanes, and equalization settings – through a duplex low speed I²C-based communication channel called SCDC (Status and Control Data Channel). After some initial handshakes, the sink asks the source to send a link training sequence, which is repeated until suitable link parameters are established. As soon as the link is established, the source transmits a test pattern on all the lanes.

Without interactive link training, FRL testing is not possible. Link training is required at start-up or when a data rate change is necessary. Special registers are defined in the specification to make link training possible on all DUTs, and special link training patterns are defined for this purpose.

Now comes the problem: two types of patterns are used during testing (the FRL link training pattern and FRL test pattern), and the AWG needs to switch instantly between the two on requests from the sink device. Here again, AWG dynamic sequencing is extremely helpful.

5.4 Using an AWG in receiver testing: Guidelines and perspectives

In this section, we address the question of when an AWG is the most suitable instrument for receiver testing.

BERTs are the primary instruments for testing receivers. A BERT consists of a pattern generator that sends test signals to the DUT. In a “loopback mode,” a special test mode, the DUT receiver identifies the data, retimes it, and sends it to the BERT error detector.

A BERT is the preferred tool for symmetric interfaces that use standard signaling; many BERTs support NRZ, PAM3, and PAM4 for applications, including PCIe, SATA, SAS, USB, and most of the OIF-CEI standards.

Compared to BERTs, AWGs are better for more complicated conditions, for example

- a unidirectional link (HDMI) or an asymmetric link (MIPI C-PHY, MIPI D-PHY, MIPI A-PHY), which does not support loopback mode;
- complex signaling, combining signals with different amplitudes and or very different data rates (i.e., combination of High Speed and Low Power signaling in MIPI C-PHY or MIPI D-PHY);
- non-standard modulation schemes not yet supported by existing BERTs (such as PAM8 or PAM16 used in MIPI A-PHY);
- non-standard signaling not supported by BERTs, such as MIPI C-PHY signaling with three levels, Ensemble NRZ signaling (ENRZ), where three bits are transmitted on four correlated wires (×1.5 data rate compared to NRZ), or other Chord^{TM} signaling);
- non-standard transmitter feedforward equalization (e.g., more than 7 taps).

6 Conclusion

In this paper we have discussed how high-speed deep-memory AWGs can be used for generating signals for testing modern serial communication interfaces such as HDMI, MIPI C-PHY, MIPI D-PHY, MIPI A-PHY, etc. We have seen that AWGs are powerful instruments but their effective use is not easy and requires some rules to be followed in the choice of AWG parameters. These rules, especially for the selection of the number of Samples per Unit Interval of the generated signal (SpUI), are seldom stated clearly in the literature. To conclude our paper, let us reiterate them.

1. The higher SpUI, the better the waveform quality. In particular, to generate an X-level signal (e.g., PAM-X) with controllable impairments, SpUI and the number of bits N_{bits} of the AWG vertical resolution should satisfy the following inequality:
$$\text{SpUI} \cdot 2^{N_{\text{bits}}-8} > \text{X}$$
2. To minimize waveform length, try to make SpUI a power of 2, an even number, or at least an integer.
3. Integer SpUI is also beneficial to prevent transition jitter, an unwanted distortion of the waveform.

References

Gentile, K. (2007). Digital pulse-shaping filter basics. Application Note AN-922. Analog Devices, Inc.

Keysight (2015). Fundamentals of Arbitrary Waveform Generation, A high performance AWG primer.

Schmidt, C., Kottke, C., Jungnickel, V., & Freund, R. (2017). High-speed digital-to-analog converter concepts. In *Next-Generation Optical Communication: Components, Sub-Systems, and Systems VI* (Vol. 10130, p. 101300N). International Society for Optics and Photonics.

Tabor Electronics (2021). Effective Number of Bits for Arbitrary Waveform Generators. White Paper.