Siglent SDS800X HD Evaluation

Revision 1.00

Table of Content

Siglent SDS800X HD Evaluation ... 1
Document History ... 4
Introduction ... 5
Basic Information ... 6
Acquisition ... 7
  Bandwidth ... 7
  Pulse Response ... 10
  True Vertical Sensitivity ... 14
  DC Check ... 16
  Peak Detect ... 18
  History & Sequence Mode ... 22
  X-Y ... 25
  Aliasing ... 28
  Noise ... 30
    Noise & Spurs ... 30
    Noise Density (1/f) ... 35
    Noise Density ... 37
    Granular Noise ... 38
Display ... 40
  Vertical Zoom Demo ... 40
  Vertical axis labels ... 42
  Zoom Expectations ... 44
  Zoom Challenge ... 49
  Dots mode ... 54
Trigger ... 61
  Trigger Jitter ... 61
  AC Trigger Coupling ... 62
  Triggering noisy signals ... 64
Measure ... 69
  Measurements ... 69
  Measurement Histograms ... 71
  Deep Measurements ... 72
  Trend Plots ... 76
  Track Plots ... 78
  Counting Pulses ... 79
Cursors ... 84
  Manual Cursors ... 84
  Tracking Cursors ... 84
  Measure Cursors ... 85
Math ... 88
  Identity ... 88
  ERES ... 89
  FFT-Setup ... 89
    FFT-Bandwidth and RBW ... 89
    Setting up the FFT ... 91
  FFT Window Functions ... 92
  FFT Dynamic range ... 97
  FFT Speed ... 100
  Distortion measurements ... 100
Analysis ... 104
  Counter ... 104

Bode Plot at a glance 105
Bode Plot Example 108
SPI Speed Test 110
The 200 Mbps SPI challenge 115
Mask Test 117
Probes 121
PP510 121
Poor Men's Differential Probing 123
System Performance with SP5050A Probe 128
Custom Probe Factors 131
MSO 136
Digital Channels 136
Serial Decoders 151
HAM Test 153
CW Test 153
FM Test 158
AM Test 162
Filter Demo 164
User Comments 168
Fun with Square Waves 168
Pulse Fidelity 178
Outstanding Features 180
Pulse Fidelity versus Anti-Aliasing 181
Figures 183
Tables 186

Document History

Rev. 1.00 Initial Release.

Introduction

These are the results of an evaluation of the Siglent SDS824X HD.

Why the 800 and not the 1000X HD?

I suppose the SDS800X HD will be the most popular offer for hobbyists and small businesses in Siglent’s lineup, eventually replacing the successful SDS1000X-E series.

Even though there are some differences, most of these are comfort features (except for the 50 Ω inputs), hence the SDS800X HD test results should be largely valid for the SDS1000X HD as well.

This is a confirmed list of differences to the SDS1000X HD:

- SDS800X HD has no external trigger input.
- Only the 200 MHz SDS800X HD have 100 Mpts memory, the lower models have only 50 Mpts.
- SDS800X HD has no 50 Ω inputs.
- SDS800X HD doesn’t have the higher quality encoders with detent positions.
- SDS800X HD has fewer serial protocols: CAN-FD and FlexRay are missing.
- SDS800X HD has only 2 USB host ports.
- SDS800X HD has only 7” capacitive touch screen, but at the same resolution 1024 x 600.
- SDS800X HD doesn’t support probe factor detection.
- SDS800X HD doesn’t support Tektronix Mode.
- SDS800X HD doesn’t support Advanced Measurements Display Mode M2.
- SDS800X HD doesn’t support Measurement Histograms Secondary Zoom.
- SDS800X HD has no RTC.
- SDS800X HD supports NTP.

The first impression was very positive. The instrument feels solid, the display appears a bit small, especially for someone used to the 10.1” screens of the 2000 series, yet the resolution is the same and it’s bright and crisp.

Operation feels snappy, it appears to be (at least) on par with the SDS2000X Plus/HD series in this regard.

The fan noise is about the same as in the SDS1104X-E, thus it can be slightly annoying and users will have something to optimize 😉

Boot time is less than 40 seconds in most scenarios.

I have to state in advance that there is a lot of progress when compared to the trusty 1000X-E series – it’s almost a completely different world.

Basic Information

This is just a collection of key-specifications, which might not be entirely clear, or even missing from the datasheet:

SDS800X HD Boot Time: <40s

SDS800X HD codes per screen height: 3840 LSB
SDS800X HD codes per division : 480 LSB

SDS800X HD best true (full resolution) sensitivity: 500 µV/div

Acquisition

Bandwidth

Let’s start with the bandwidth. We would like to get the specified bandwidth even with all channels active, yet we do not want to deal with excessive aliasing.

At first, one single channel at 2 GSa/s:

Fig. 1 SDS824X_HD_FR_2GSa_log

Amplitude drop at 200 MHz is less than 2 dB and actual -3 dB bandwidth is 244 MHz. Frequency response is even a tad better when two channels are in use at 1 GSa/s:

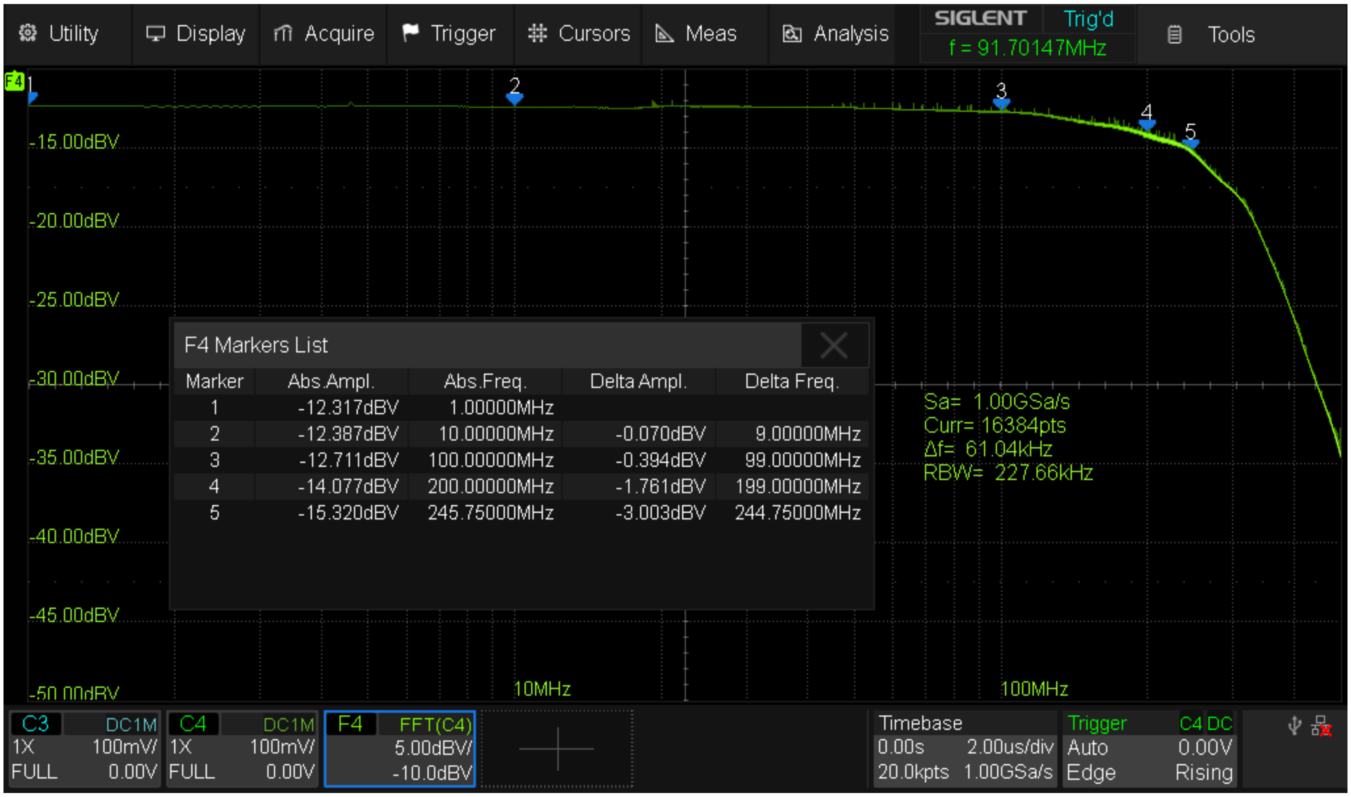

Fig. 2 SDS824X_HD_FR_1GSa_log

Amplitude drop at 200 MHz is <1.8 dB and actual -3 dB bandwidth is 245 MHz. Finally, we look at all four channels in use at only 500 MSa/s:

Fig. 3 SDS824X_HD_FR_500MSa_log

Now the bandwidth is actually limited to the advertised 200 MHz.

Finally the frequency response with 20 MHz bandwidth limiter:

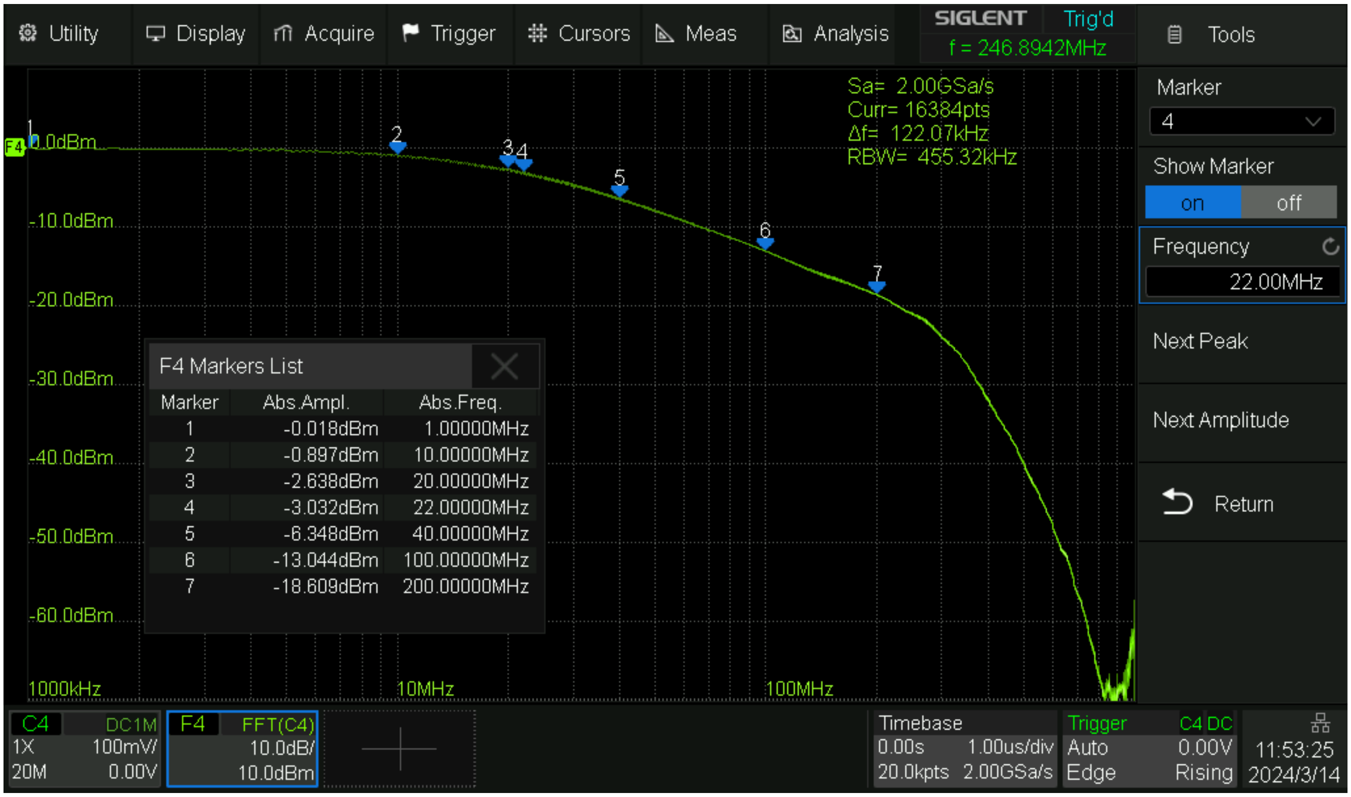

Fig. 4 SDS824X_HD_FR_500MSa_log

Pulse Response

For all these tests, a 10 MHz square wave with 1 ns rise time has been fed into cannel 4.

Let's start with single channel mode and 2 GSa/s:

Fig. 5 SDS824X_HD_PR_2GSa_Zoom_Stop

In stop mode, we get a clear picture of the imperfections of the pulse top even when zoomed in 20x (main window: 100 mV/div, zoom: 5 mV/div). The rise time measurements yield the expected result of ~1.8 ns, which corresponds to 1.5 ns rise time for the SDS824X HD. This is well below the specified 1.8 ns for the 200 MHz model.

In Run mode we can see some modulation because of noise, yet nothing that could not be cured by averaging using a math trace:

Fig. 6 SDS824X_HD_PR_2GSa_Zoom_Avg16

In case you wonder why the imperfections of the pulse top are so pronounced in the previous screenshot, this is simply the price we pay for less than perfect impedance matching when using a scope that lacks 50 Ω inputs. External termination is always a compromise working reasonably well up to 70 MHz at best. Fast edges like the 1 ns rise time in this example occupy 600 MHz bandwidth. The output impedance of the pulse generator isn't perfect 50 ohms either, and both phenomena combined lead to reflections showing up in the first ~16 ns of the pulse.

Of course it can be demonstrated, how better impedance matching improves things. For this I've used a quality 18 GHz cable with two 10 dB 18 GHz Narda in-line attenuators (one at each end) to ensure sufficient attenuation for any reflections between generator and DSO. Because of the 20 dB attenuation in total, I had to increase the generator output level by 20 dB as well. This would have been 6 V amplitude, but at that level, it's rise time is limited to min. 1.2 ns, hence I made do with just 3 V and increased the DSO sensitivity to 50 mV/div, in order to still get the 1 ns rise time:

Fig. 7 SDS824X_HD_PR_2GSa_Zoom_Avg16_Match

With two active channels, the sample rate drops to 1 GSa/s:

Fig. 8 SDS824X_HD_PR_1GSa

The overshoot is more pronounced now (possibly because of additional AA-filtering), yet rise time measurements haven't changed.

With four active channels, the sample rate drops to only 500 MSa/s:

Fig. 9 SDS824X_HD_PR_500MSa

We can see hints on slight reconstruction errors together with rather pronounced Gibbs ears. Rise time measurement is off by hefty 27%, so we can safely state that this configuration is not fit for characterizing pulses with <2 ns rise time.

With a signal rise time of 2ns we can measure 2.6 ns: assuming 1.5 ns rise time for the SDS824X HD, this measurement now is only ~5% off and should be acceptable already. Furthermore, we can use Dots mode to get rid of any reconstruction errors:

Fig. 10 SDS824X_HD_PR_500MSa_2ns_Dots

True Vertical Sensitivity

The SDS800X HD has a specified vertical gain range from 500 µV/div up to 10 V/div. Many contemporary DSOs have similar specs, yet only a small minority of them can provide true 500 µV/div at full resolution. The real sensitivity of many instruments is lower, sometimes significantly so (up to 5 mV/div). As a consequence, anything above the true highest sensitivity is just software zoom and won't provide full resolution anymore. This might be not that much of a problem for a 12-bit DSO, but 8-bit instruments degraded to 6 bits at 1 mV/div could get problematic. On the other hand, most of those instruments also exhibit high noise levels, so the ENOB (Effective Number of Bits) drops below 6 bits at these higher sensitivities anyway.

For the SDS800X HD, I stumbled across the unexpected property of nearly equal noise levels for all vertical sensitivities from 500 µV/div to 5 mV/div:

Noise Density	RBW [Hz]	889,3													
Gain [mV/div]	0,50			1,00			2,00			5,00			10,00		
Freq. [Hz]	Level [dBV]	Level [V]	ND [nV/√Hz]	Level [dBV]	Level [V]	ND [nV/√Hz]	Level [dBV]	Level [V]	ND [nV/√Hz]	Level [dBV]	Level [V]	ND [nV/√Hz]	Level [dBV]	Level [V]	ND [nV/√Hz]
1,0E+3	-103,17	6,95E-6	232,90E-9	-101,93	8,01E-6	268,58E-9	-101,83	8,10E-6	271,60E-9	-103,06	7,03E-6	235,79E-9	-99,19	10,98E-6	368,24E-9
3,0E+3	-103,66	6,56E-6	220,00E-9	-105,69	5,19E-6	174,17E-9	-105,08	5,57E-6	186,93E-9	-105,35	5,40E-6	181,06E-9	-103,12	6,98E-6	234,06E-9
10,0E+3	-112,56	2,36E-6	79,00E-9	-114,59	1,86E-6	62,49E-9	-113,74	2,06E-6	68,93E-9	-113,74	2,06E-6	68,97E-9	-112,36	2,41E-6	80,86E-9
30,0E+3	-126,13	493,86E-9	16,56E-9	-125,99	501,59E-9	16,82E-9	-126,44	476,49E-9	15,98E-9	-125,82	511,45E-9	17,15E-9	-121,22	869,36E-9	29,15E-9
100,0E+3	-133,30	216,30E-9	7,25E-9	-135,37	170,47E-9	5,72E-9	-133,55	210,23E-9	7,05E-9	-132,67	232,59E-9	7,80E-9	-127,63	415,29E-9	13,93E-9
300,0E+3	-141,85	80,85E-9	2,71E-9	-140,45	95,01E-9	3,19E-9	-141,44	84,75E-9	2,84E-9	-139,14	110,36E-9	3,70E-9	-130,92	284,58E-9	9,54E-9
1,0E+6	-140,83	90,89E-9	3,05E-9	-140,80	91,23E-9	3,06E-9	-142,86	71,92E-9	2,41E-9	-141,56	83,59E-9	2,80E-9	-133,60	208,98E-9	7,01E-9
10,0E+6	-142,05	79,00E-9	2,65E-9	-142,66	73,65E-9	2,47E-9	-141,00	89,18E-9	2,99E-9	-141,27	86,39E-9	2,90E-9	-135,29	172,07E-9	5,77E-9

Table 1 SDS824X HD_ND

These numbers are not totally accurate because it proves very difficult to place the markers close to the intended frequency without hitting a minor spur. Consequently, I would think that the noise level is fairly uniform across all the higher sensitivities from 500 µV/div to 5 mV/div and the minima across all measurements would be the best representation of the truth:

Noise 500 µV/div – 5 mV/div :
1 kHz : 232.9 nV/√Hz
3 kHz : 174.2 nV/√Hz
10 kHz : 62.5 nV/√Hz
30 kHz : 16.0 nV/√Hz
100 kHz : 5.7 nV/√Hz
300 kHz : 2.7 nV/√Hz
1 MHz : 2.4 nV/√Hz
10 MHz : 2.5 nV/√Hz

This made me suspicious: does Siglent cheat after all? Are all vertical gain settings below 5 mV/div just fake? First, I’ve checked the raw acquisition data for 500 µV/div vertical gain and found the lowest voltage step to be 1.042 µV.

Time [s]	Value [V]	Delta [V]
-4.0000000000E-08	-4,166667E-05	5,208E-6
-3.9500000000E-08	-4,166667E-05	000,000E+0
-3.9000000000E-08	-4,270833E-05	[b]1,042E-6[/b]

The SDS800X HD has 480 LSB per vertical division (just like the SDS2000X HD), thus 3840 LSB on the visible part of the screen. Since a 12-bit acquisition system provides a total 0f 4096 LSB, there is very little headroom outside the visible screen area.

The interesting part is when we multiply the 1.042 µV resolution with the 480 LSB of one division: 1.042 * 480 ~ 500 µV/div; -> Bingo!

A less accurate, but quicker and simpler method to verify the resolution of the SDS800X HD is using vertical zoom; we can zoom into the noise in dots display mode, thus getting horizontal lines vertically spaced according to the true resolution of the instrument.

At a vertical gain of 500 µV/div and a vertical zoom window at 2 µV/div, we get the following picture:

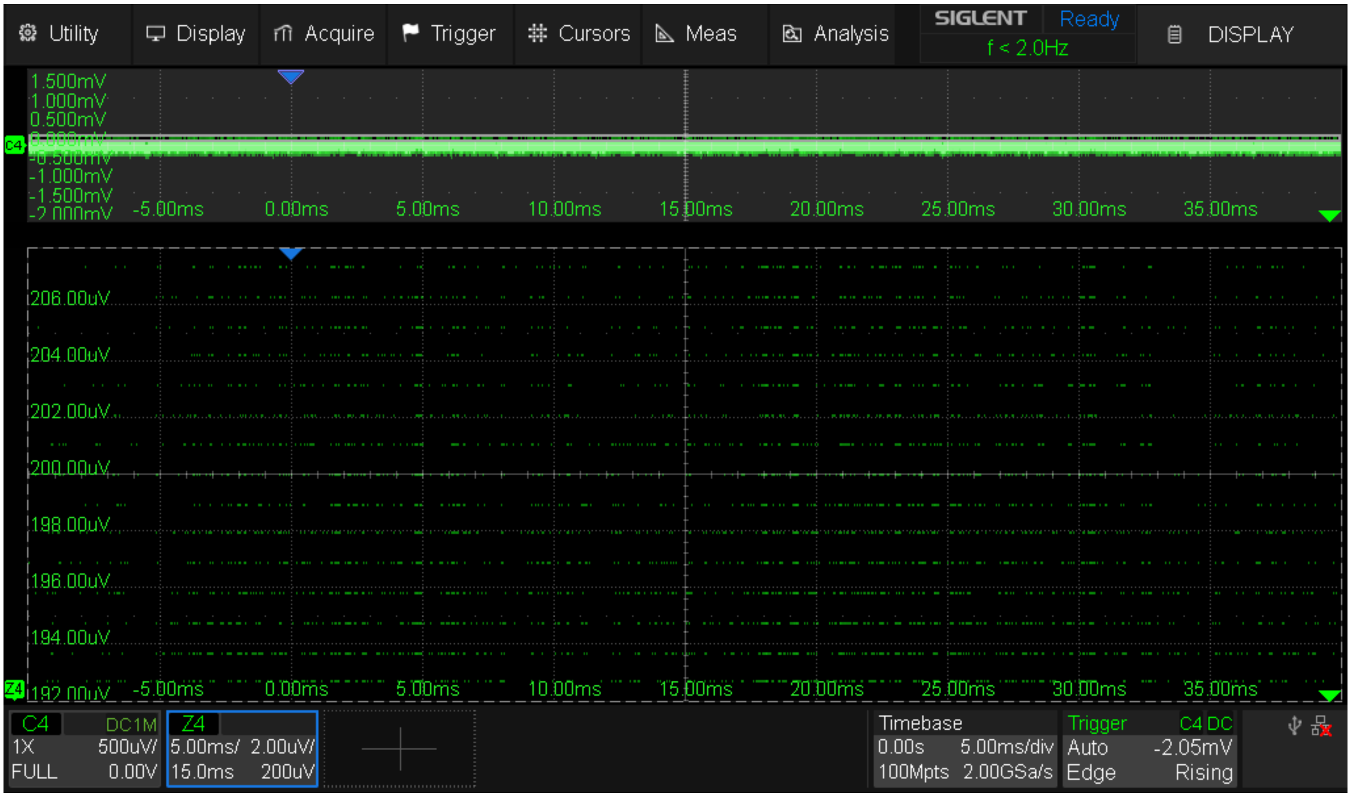

Fig. 11 SDS824X_HD_Resolution_Demo

Since we still have 8 vertical divisions also in the zoom window, the total visible screen height covers 16 µV at 2 µV/div. We can count 15 horizontal lines, hence 16 steps and can conclude that each step has to be close to one microvolt.

Verdict: Siglent don't cheat. The uniform noise level at and below 5 mV/div is just a property of the integrated PGA (Programmable Gain Amplifier) used in this instrument.

DC Check

One of the advantages of a 12-bit DSO should be not only high resolution, but also good accuracy. The SDS800X HD has a typical error of 0.5% at vertical gain settings from 5 mV/div up to 10 V/div, thus entering 3.5-digit DMM territory.

Here are some checks with a 6 V DC "signal".

First the uninspired way: all default settings with the trace position at 0 V at the vertical center of the screen – we need 2 V/div vertical gain in this scenario:

Fig. 12 SDS824X_HD_DC_6V_G2V_0.42%

The Result is 6.025 V, this is a deviation of almost 0.42%. Now let’s try something different instead:

Fig. 13 SDS824X_HD_DC_6V_G1V_0.29%

Now we've tried to better utilize the available dynamic range. We are able to get the entire range from 0 to 6 volts on the screen with a vertical gain of only 1 V/div, by dialing in an offset compensation of 3 V. The result tells us that we 're on the right track<. Just 0.29% deviation!

Of course we can do even better. Who says that the zero volts position need to be visible if we want to measure 6 volts? Our last attempt to accurately measure 6 volts uses a vertical gain of 100 mV/div together with an offset compensation of 6 volts. The resulting signal should be ideally zero. Deviation with regard to the vertical screen center.

The automatic measurements (always look at the mean value!) show 6.00485 V now, this is equivalent to a deviation of 0.081%!

Fig. 14 SDS824X_HD_DC_6V_G100mV_0.08%

In the end it had been demonstrated, that with the right method, the accuracy of a voltage measurement can approach that of an average 4-½ digit DMM.

Peak Detect

The peak detection capability of the SDS800X HD is specified as 2 ns. Let's have a closer look at that.

First a 2 ns wide pulse with 300 mV amplitude and 500 ps rise time in normal acquisition mode at sufficient sample rate (2 GSa/s):

Fig. 15 SDS824X_HD_Pulse_W2ns_RT500ps_2GSa_Norm_Zoom

It can be seen that such a narrow pulse is already a bit too much for a 200 (244) MHz oscilloscope; the amplitude has already dropped a bit and pulse width measurement isn't quite accurate either. As expected, the rise time measurement approaches the scope's own rise time.

With all these shortcomings, we still get a fairly stable picture – look at the main window and the peak and standard deviations in the measurement statistics.

In the screenshot above, the time base was at 5 ms/div and the sample memory was already at its maximum of 100 Mpts; slowing down the time base any further will inevitably lower the sample rate:

Fig. 16 SDS824X_HD_Pulse_W2ns_RT500ps_100MSa_Norm_Zoom

At 100 ms/div and 100 Mpts record length the sample rate has to be decimated to just 100 MSa/s – far too slow for capturing a 2 ns wide pulse. As a consequence, many pulses get lost. In the main window we would expect to see about 1000 pulses at a pulse repetition rate of 1 kHz, but there are actually much less and the amplitudes vary wildly.

This isn’t a very realistic scenario; not many engineers would try to watch 2 ns wide pulses at a time base of 100 ms/div and have to use 2 million times zoom to watch the pulse details. Yet this is where Peak Detect acquisition mode comes into play:

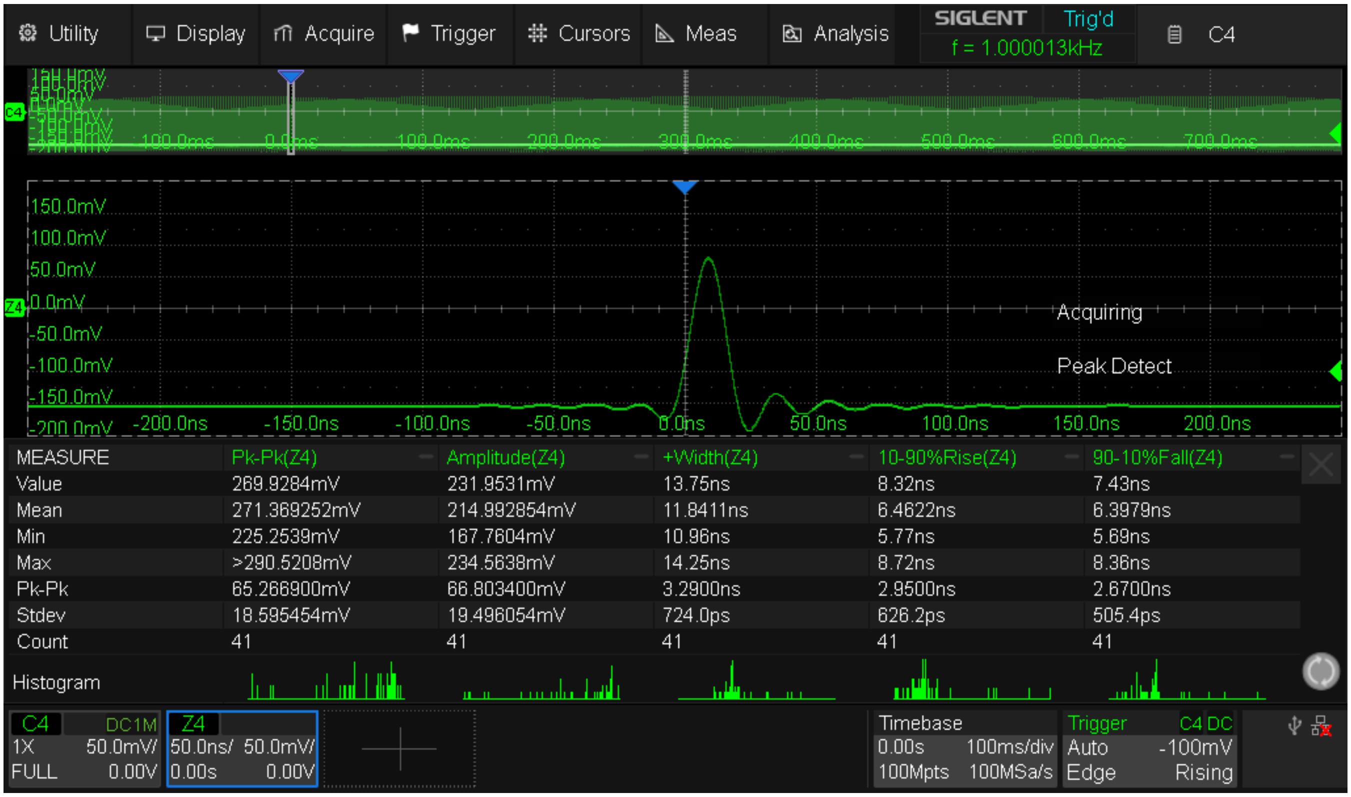

Fig. 17 SDS824X_HD_Pulse_W2ns_RT500ps_100MSa_Peak_Zoom

The main window now shows all the pulses; the amplitudes still vary a bit, but at least we don't miss any pulses anymore. Pulse shape has nothing to do with reality anymore and measurements yield just house numbers. This should be a clear warning to not use Peak Detect for anything serious, as any math and measurements on such waveforms are of artistical value at best.

All that Peak Detect really can do is to hint on any pulses within the record.

Of course, peak detection works for even narrower pulses just as well. This is not because the specification is not correct, but the simple fact that a 244 MHz DSO like the SDS824X HD simply cannot process even faster pulses:

Fig. 18 SDS824X_HD_Pulse_W1ns_RT500ps_2GSa_Norm_Zoom

This is now a 1 ns wide pulse at maximum sample rate of 2 GSa/s. The amplitude is still 300 mV, yet the SDS824X HD cannot cope with it anymore and the amplitude measurement result has dropped to just 173 mV. The pulse width is still measured as 1.8 ns, so the relative slowness of the frontend widens shorter pulses at the expense of amplitude, hence makes an even faster peak detection unnecessary.

History & Sequence Mode

Inspired by the complaint here:

https://www.eevblog.com/forum/testgear/rigol-hdo1000-and-hdo4000-12bit-oscilloscopes-launched-in-china/msg5269170/#msg5269170

I've tried to replicate the fairly moderate test scenario described by forum member Egonotto. For this, the SDS800X HD doesn't need any special mode; the always active background history can handle that:

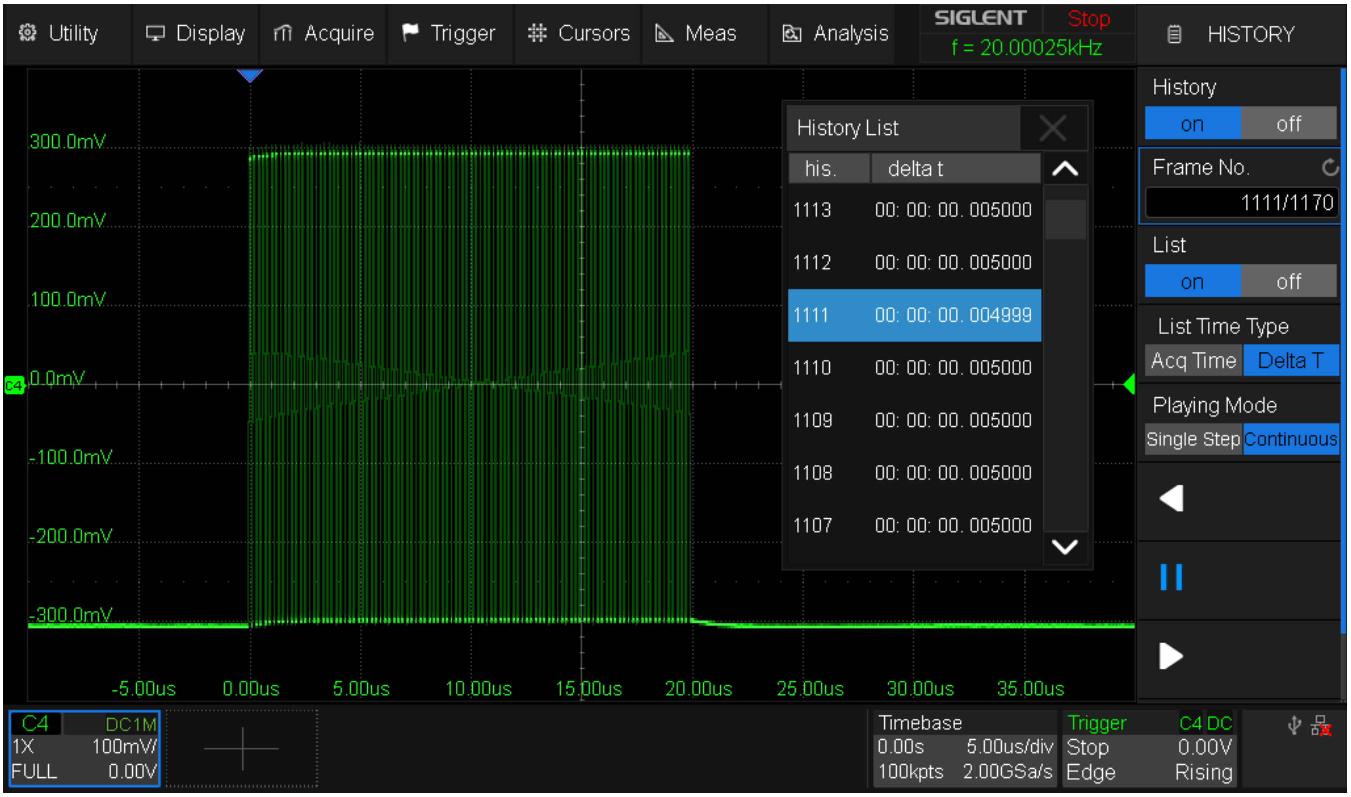

Fig. 19 SDS824X_HD_5ms_Hist

The test signal is a burst packet, 20 µs long, consisting of 100 pulses. The repetition interval (burst period) was 5 ms for this test.

At 5 µs/div, the SDS824X HD takes an average of 650 µs/frame and a maximum of <2 ms/frame. The screenshot shows the History List displaying the time delta between the packets. It is 5 ms throughout, with the occasional 4.999 ms because of the not so accurate time base of the SDS800 (25 ppm vs. 1 ppm in the SDS2000X Plus/HD series).

For event recording, we'd rather use the dedicated Sequence mode. This provides a constant 52 µs/trigger @ 5 µs/div, hence can capture a burst period of 100 µs without a single missing frame:

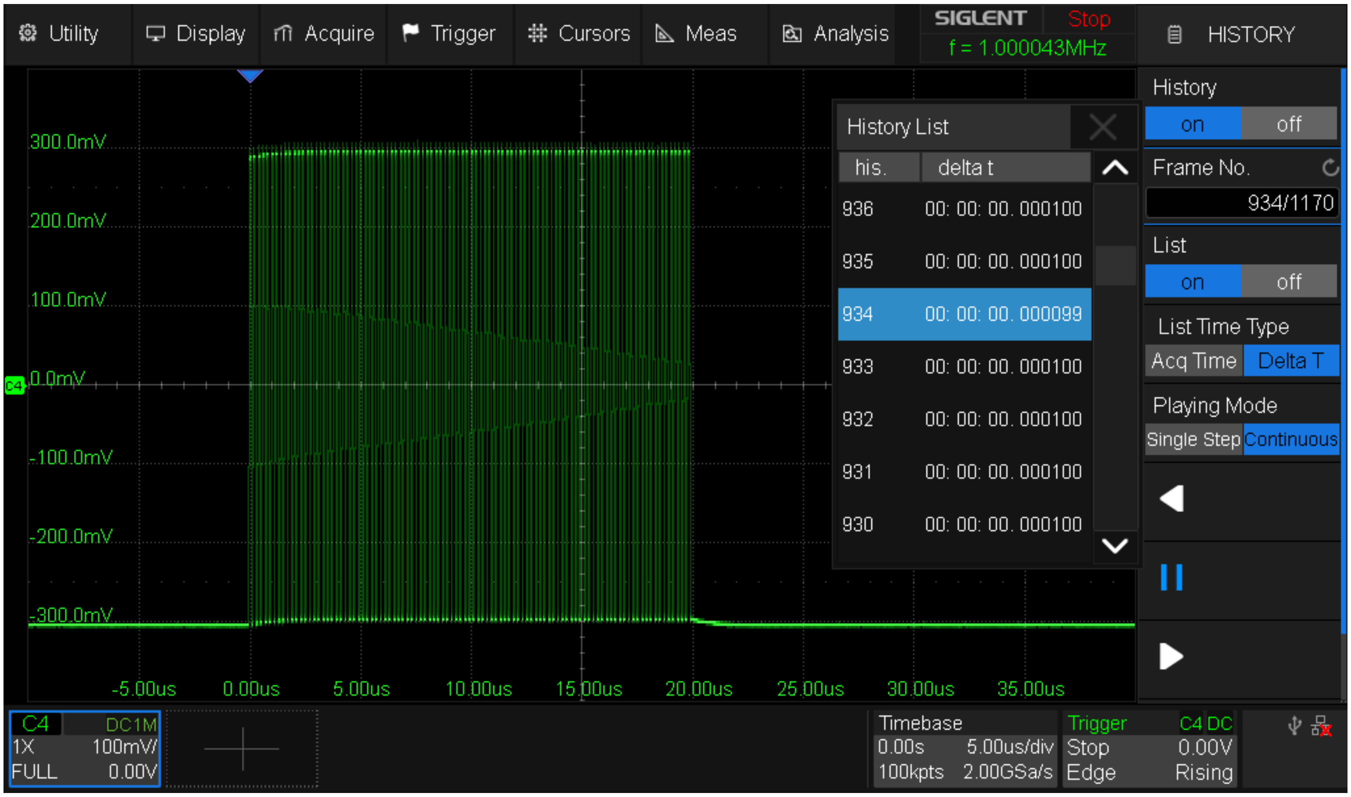

Fig. 20 SDS824X_HD_100us_Seq

The screenshot shows the History List displaying the time delta between the packets. It is 100 µs throughout, with the occasional 99 µs because of the not so accurate time base of the SDS800.

For complete information, here are my measurements for the trigger rates during normal use with vector and dots display mode as well as sequence recording from the fastest time base of 1 ns/div up to 100 µs/div. The input signal was not optimized for this test.

				Vector		Dots		Sequence	
TB [s/div]	RecLen [pts]	Tot. Mem [pts]	Frames [-]	time/frm [s]	Rate [frm/s]	time/frm [s]	Rate [frm/s]	time/frm [s]	Rate [frm/s]
1,00E-9	20,00E+0	1,60E+6	80000	124,45E-6	8,04E+3	124,29E-6	8,05E+3	32E-6	31,56E+3
2,00E-9	40,00E+0	3,20E+6	80000	78,57E-6	12,73E+3	72,64E-6	13,77E+3	16E-6	60,76E+3
5,00E-9	100,00E+0	8,00E+6	80000	60,49E-6	16,53E+3	44,40E-6	22,52E+3	7E-6	136,18E+3
10,00E-9	200,00E+0	16,00E+6	80000	54,15E-6	18,47E+3	34,18E-6	29,25E+3	4E-6	231,84E+3
20,00E-9	400,00E+0	32,00E+6	80000	52,54E-6	19,03E+3	29,62E-6	33,76E+3	3E-6	358,79E+3
50,00E-9	1,00E+3	45,98E+6	45976	34,22E-6	29,22E+3	8,43E-6	118,69E+3	2E-6	506,48E+3
100,00E-9	2,00E+3	91,95E+6	45976	46,40E-6	21,55E+3	14,06E-6	71,12E+3	2E-6	506,45E+3
200,00E-9	4,00E+3	92,84E+6	23209	74,99E-6	13,33E+3	66,01E-6	15,15E+3	3E-6	308,11E+3
500,00E-9	10,00E+3	103,70E+6	10370	144,08E-6	6,94E+3	136,76E-6	7,31E+3	6E-6	155,13E+3
1,00E-6	20,00E+3	110,02E+6	5501	233,08E-6	4,29E+3	226,93E-6	4,41E+3	12E-6	86,42E+3
2,00E-6	40,00E+3	117,00E+6	2925	412,59E-6	2,42E+3	408,07E-6	2,45E+3	21E-6	47,25E+3
5,00E-6	100,00E+3	117,00E+6	1170	641,09E-6	1,56E+3	643,44E-6	1,55E+3	52E-6	19,29E+3
10,00E-6	200,00E+3	117,60E+6	588	1,24E-3	807,51E+0	1,24E-3	807,51E+0	102E-6	9,78E+3
20,00E-6	400,00E+3	118,00E+6	295	1,72E-3	580,56E+0	1,73E-3	576,69E+0	202E-6	4,95E+3
50,00E-6	1,00E+6	117,00E+6	117	3,15E-3	317,73E+0	3,11E-3	321,58E+0	502E-6	1,99E+3
100,00E-6	2,00E+6	116,00E+6	58	5,64E-3	177,38E+0	5,64E-3	177,38E+0	994E-6	1,01E+3

Table 2 SDS824X HD Trigger rate

X-Y

As usual, X-Y mode is hardware accelerated with high waveform update rates and intensity or color grading.

An important property is the waveform update rate in X-Y acquisition mode and it shows that at faster time bases (≤2 µs/div) the trigger rate is even higher in X-Y mode than the corresponding 2-channel Y-t mode. The table below shows the trigger rates for various time base settings from 100 ns/div up to 1 ms/div and compares the trigger rates in regular dual channel Y-t mode to the X-Y mode:

Mode	X-Y		Display	default		SR [Sa/s]	1,0E+9	
TB [s/div]	Signal [Hz]	Trig. Y-t [Hz]	Trig. X-Y [Hz]	RecLen [pts]	time/frm Y-t [s]	Rate Y-t [frm/s]	time/frm X-Y [s]	Rate X-Y [frm/s]
100,00E-9	2,00E+6	17,40E+3	63,30E+3	1,00E+3	57,47E-6	**17400,0**	15,80E-6	**63300,0**
200,00E-9	1,00E+6	15,16E+3	51,10E+3	2,00E+3	65,96E-6	**15160,0**	19,57E-6	**51100,0**
500,00E-9	500,00E+3	10,80E+3	23,20E+3	5,00E+3	92,59E-6	**10800,0**	43,10E-6	**23200,0**
1,00E-6	200,00E+3	7,68E+3	12,09E+3	10,00E+3	130,21E-6	**7680,0**	82,71E-6	**12090,0**
2,00E-6	100,00E+3	4,83E+3	5,67E+3	20,00E+3	207,04E-6	**4830,0**	176,37E-6	**5670,0**
5,00E-6	50,00E+3	2,87E+3	2,45E+3	50,00E+3	348,43E-6	**2870,0**	408,16E-6	**2450,0**
10,00E-6	20,00E+3	1,64E+3	1,21E+3	100,00E+3	609,76E-6	**1640,0**	826,45E-6	**1210,0**
20,00E-6	10,00E+3	787,00E+0	546,00E+0	200,00E+3	1,27E-3	**787,0**	1,83E-3	**546,0**
50,00E-6	5,00E+3	384,00E+0	260,00E+0	500,00E+3	2,60E-3	**384,0**	3,85E-3	**260,0**
100,00E-6	2,00E+3	194,00E+0	112,00E+0	1,00E+6	5,15E-3	**194,0**	8,93E-3	**112,0**
200,00E-6	1,00E+3	28,70E+0	28,70E+0	2,00E+6	34,84E-3	**28,7**	34,84E-3	**28,7**
500,00E-6	500,00E+0	28,70E+0	19,20E+0	5,00E+6	34,84E-3	**28,7**	52,08E-3	**19,2**
1,00E-3	200,00E+0	19,23E+0	11,49E+0	10,00E+6	52,00E-3	**19,2**	87,03E-3	**11,5**

Table 3 SDS824X HD_XY_UpdateRate

The maximum speed at 100 ns/div was more than 63000 updates per second and X-Y mode limits the time base, so that it cannot get any faster than that. Up to 1 µs/div, the update speed is always greater than 10000 per second and from there it scales as expected, i.e. the trigger speed is inversely proportional to the record length.

The figures stated above are valid for the full sample rate of 1 GSa/s, which also means record lengths of e.g. 10 Mpts at 1 ms/div. In other words, these numbers represent the worst case and X-Y operation could be accelerated by limiting the record length (thus also reducing the sample rate).

I want to show some examples, which also demonstrate the intensity and color grading. First a familiar Lissajous figure, and then some I/Q waveform patterns at 1 Mbps, which can also serve as a speed demonstration because of their complexity.

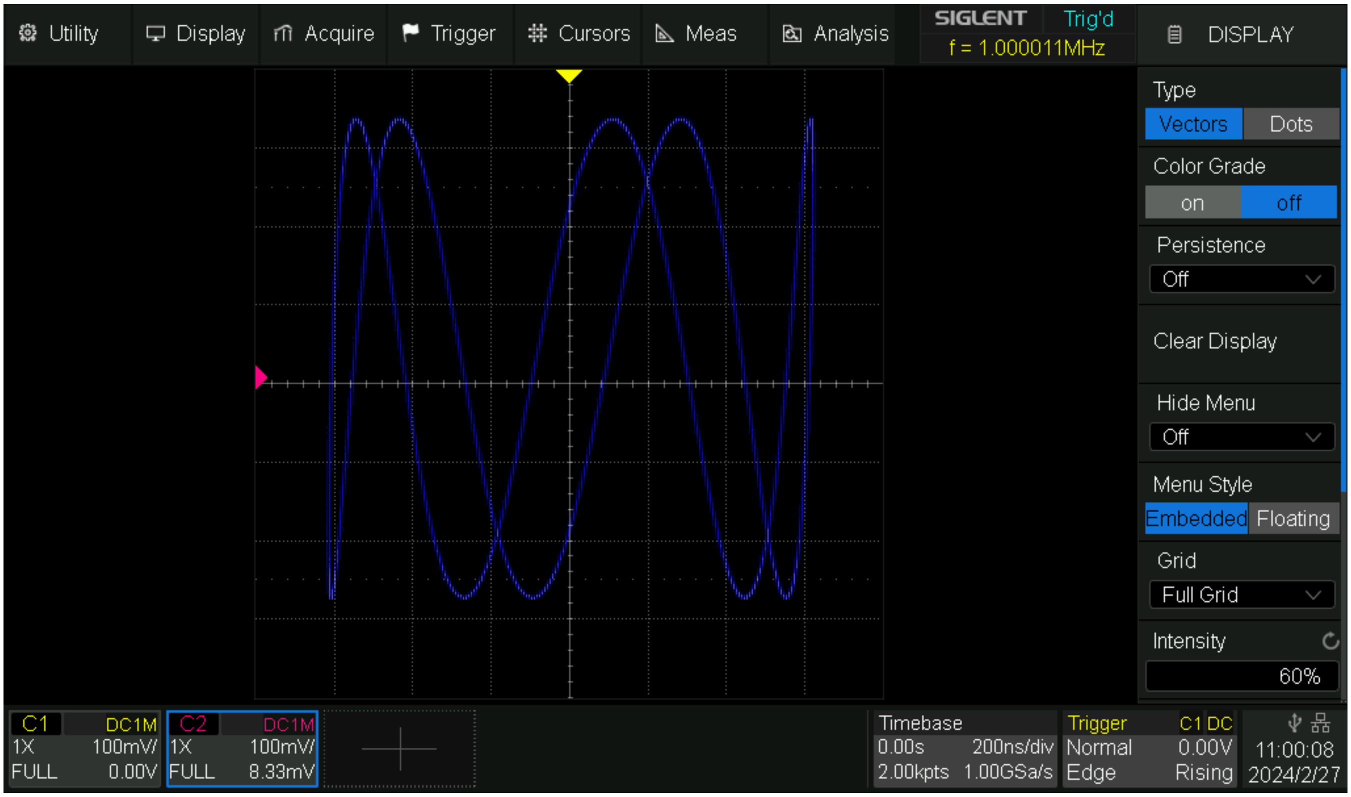

Fig. 21 SDS824X HD_Sine_1MHz_5MHz_45deg_IG

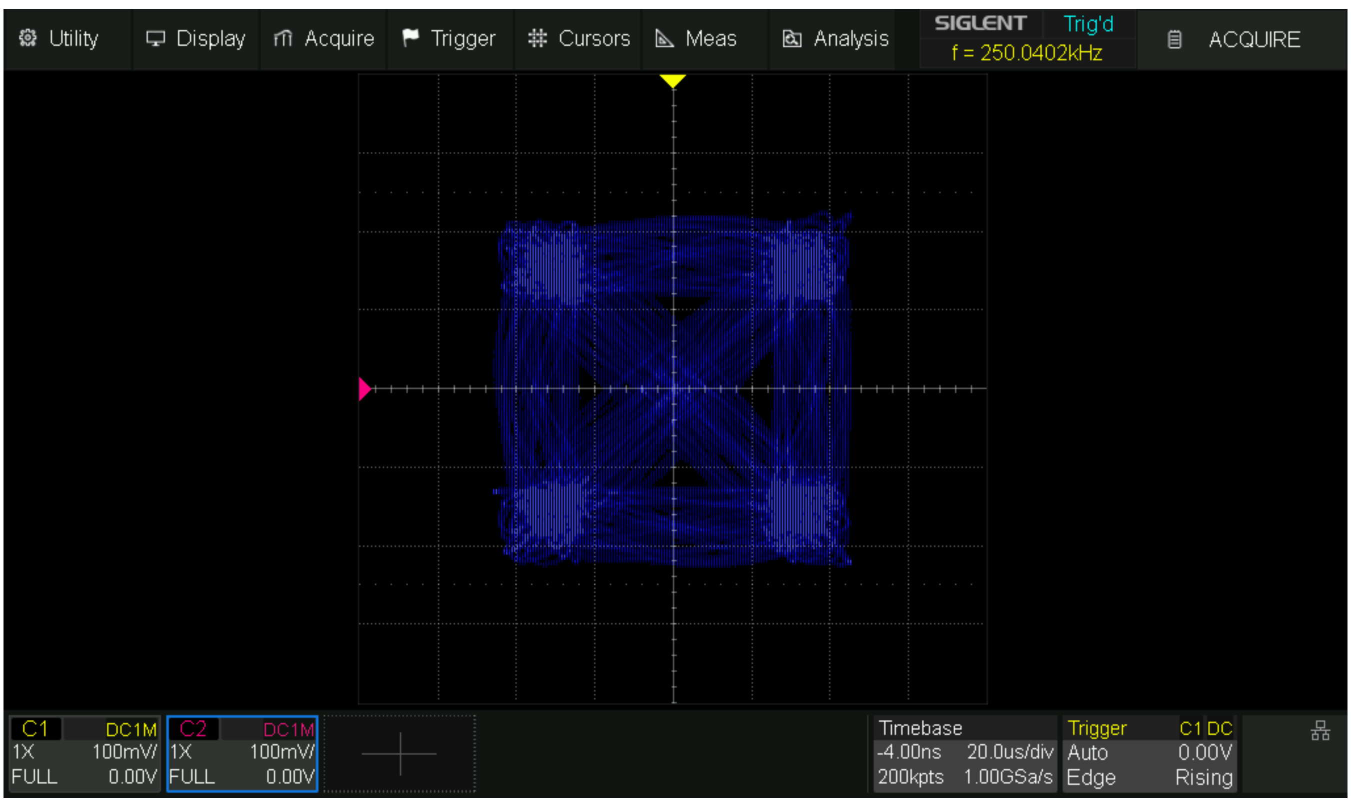

Fig. 22 SDS824X HD_QPSK_1Mbps_CG

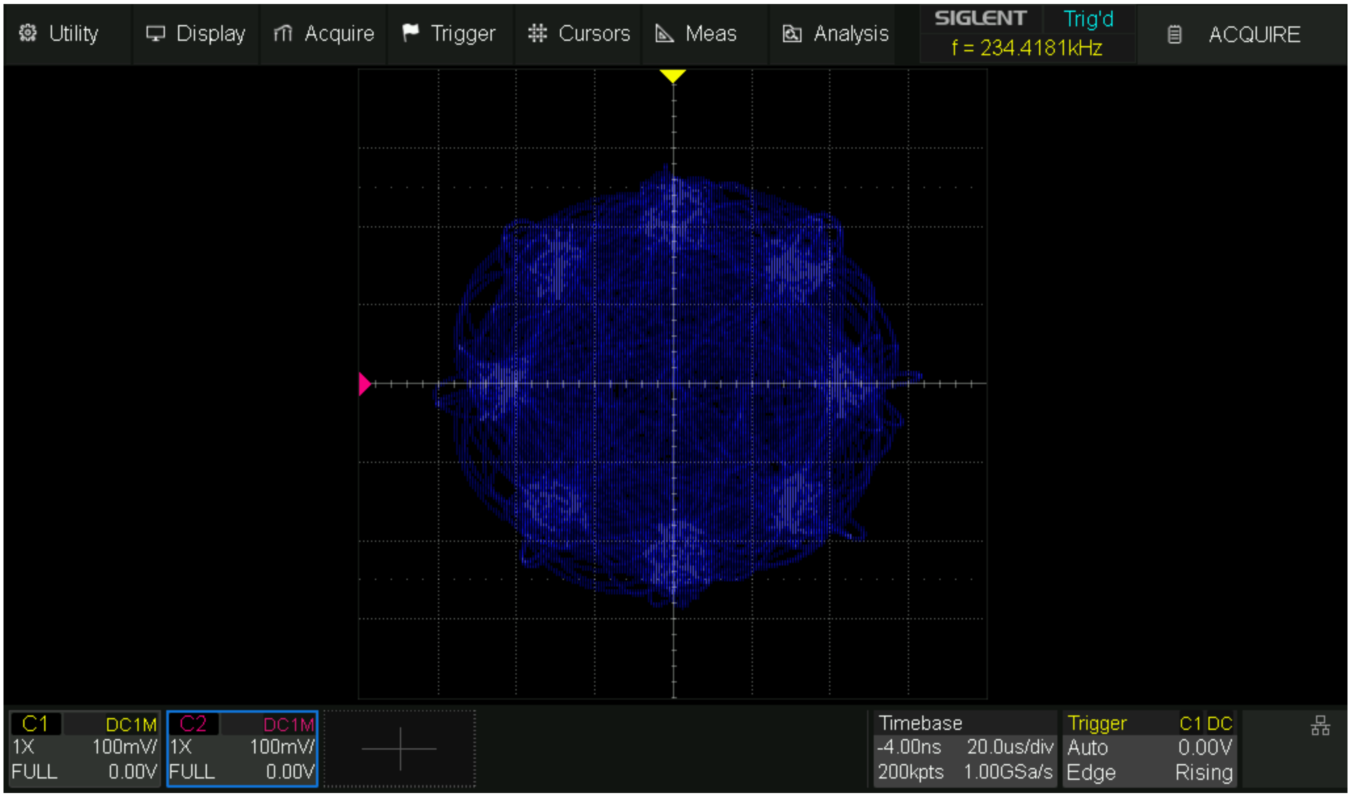

Fig. 23 SDS824X HD_8PSK_1Mbps

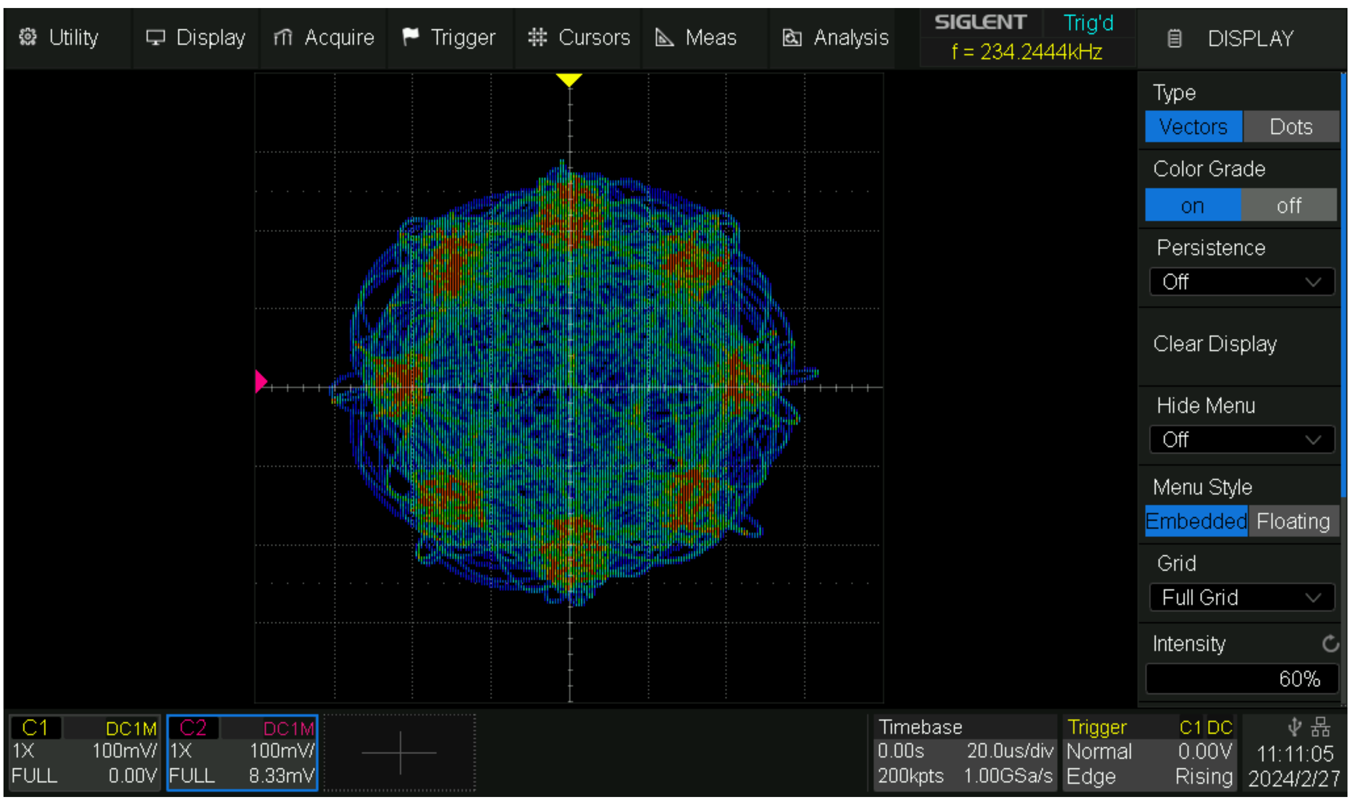

Fig. 24 SDS824X HD_D8PSK_1Mbps_CG

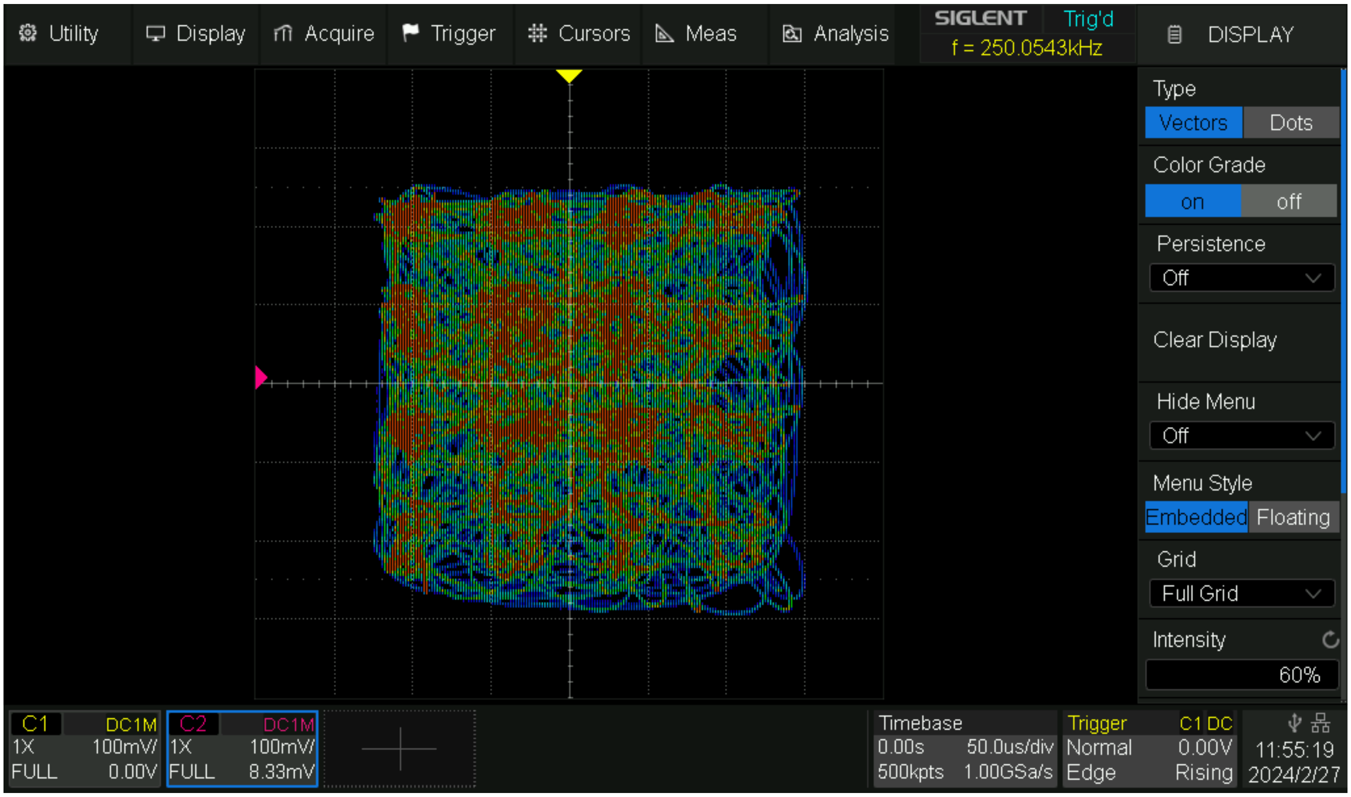

Fig. 25 SDS824X HD_16QAM_1Mbps_CG

Aliasing

Let's have a look at aliasing with only one single channel active and 2 GSa/s.

Amplitude drop at 200 MHz is less than 2 dB and actual -3 dB bandwidth is 250 MHz. We can also see that in this configuration we have a very high protection against aliasing: >86 dB attenuation at the Nyquist frequency of 1 GHz is way more than enough even for a 12-bit acquisition system.

Fig. 26 SDS824X_HD_FR_2GSa

Here is the aliasing situation with two channels active and 1 GSa/s:

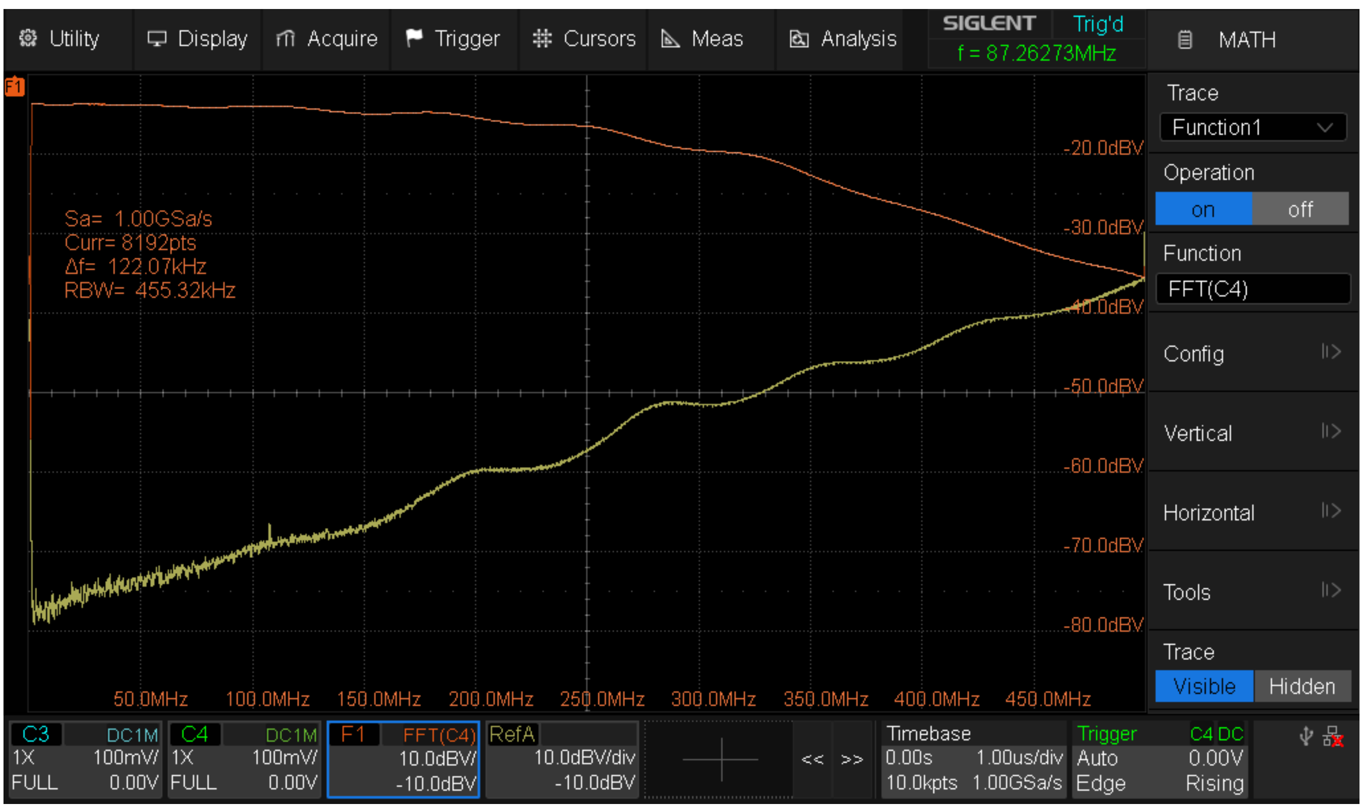

Fig. 27 SDS824X_HD_Aliasing_1GSa

Math trace F1 (orange) shows the frequency response in the first Nyquist zone (0-500 MHz), whereas reference trace A (yellow) represents the 2nd Nyquist zone (500 MHz to 1 GHz).

Reference Level is about -13 dBV and the attenuation at the Nyquist frequency of 500 MHz is only ~23 dB, but is dropping reasonably fast at even higher frequencies. We get >-40 dBc at 750 MHz. For any practical signals applied to a DSO in this class the spectrum shouldn't be that aggressive, hence users will rarely experience aliasing artifacts.

Finally, the aliasing with more than two channels active, when the sample rate drops to just 500 MSa/s:

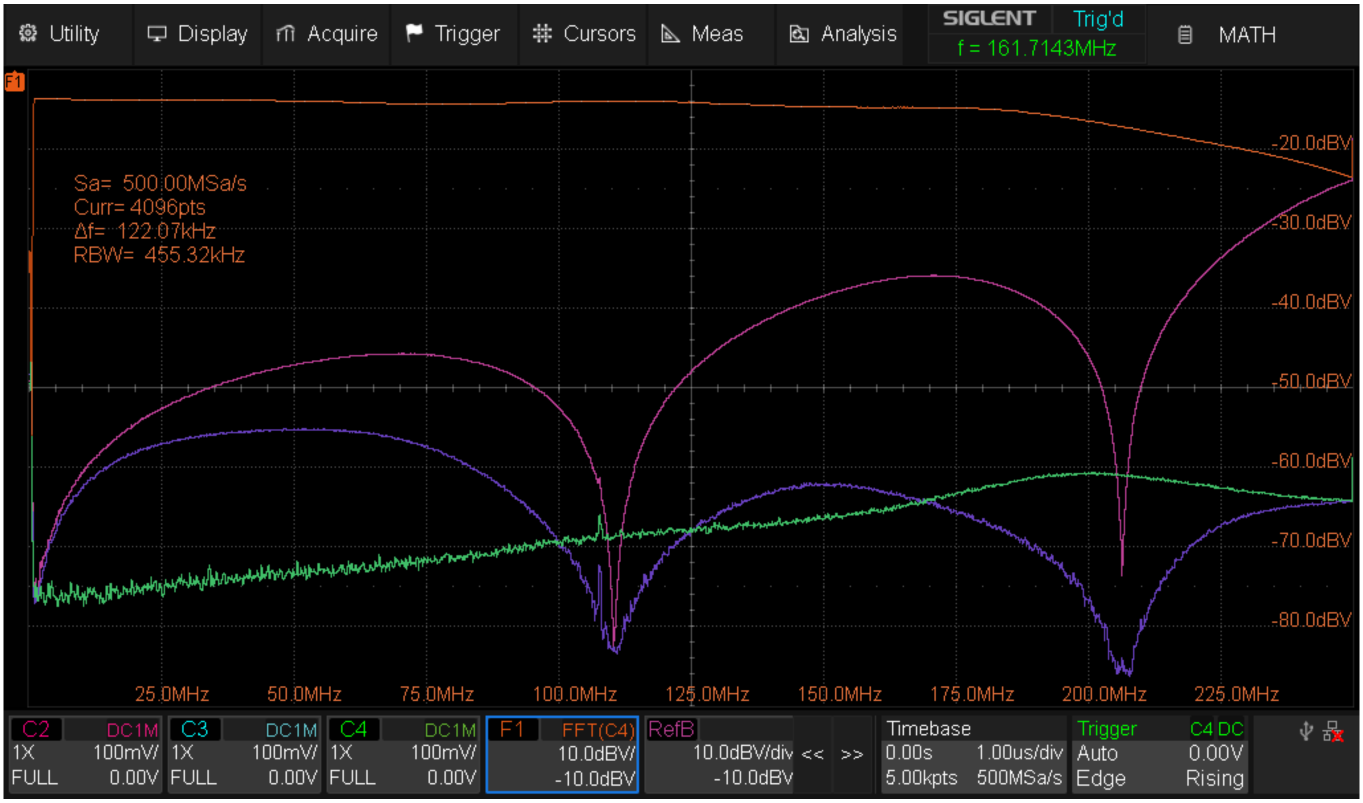

Fig. 28 SDS824X_HD_Aliasing_500MSa

Math trace F1 (orange) shows the frequency response in the first Nyquist zone (0-250 MHz), whereas reference trace B (magenta) represents the 2nd Nyquist zone (250 - 500 MHz). Reference trace C (violet) represents the 3rd Nyquist zone (500 - 750 MHz) and finally reference trace D (green) plots the 4th Nyquist zone (750 MHz – 1 GHz). Reference Level is about -13 dBV and the attenuation at the Nyquist frequency of 250 MHz is only ~11 dB, but is dropping pretty fast beyond that. We get -34 dBc at 325 MHz.

Noise

Noise & Spurs

This is a demonstration of the noise with all channels active, where the bandwidth is limited to true 200 MHz.

The noise is shown for various conditions:

Ch.1: input open, 200 MHz bandwidth;
Ch.2: input open, 20 MHz bandwidth;
Ch.3: input 50 ohm terminated, 200 MHz bandwidth;
Ch.4: input 50 ohm terminated, 20 MHz bandwidth;

Because of the pronounced 1/f characteristic of the frontend noise below about 300 kHz, the results strongly depend on the lower bandwidth limit. Let’s start with 100 kHz:

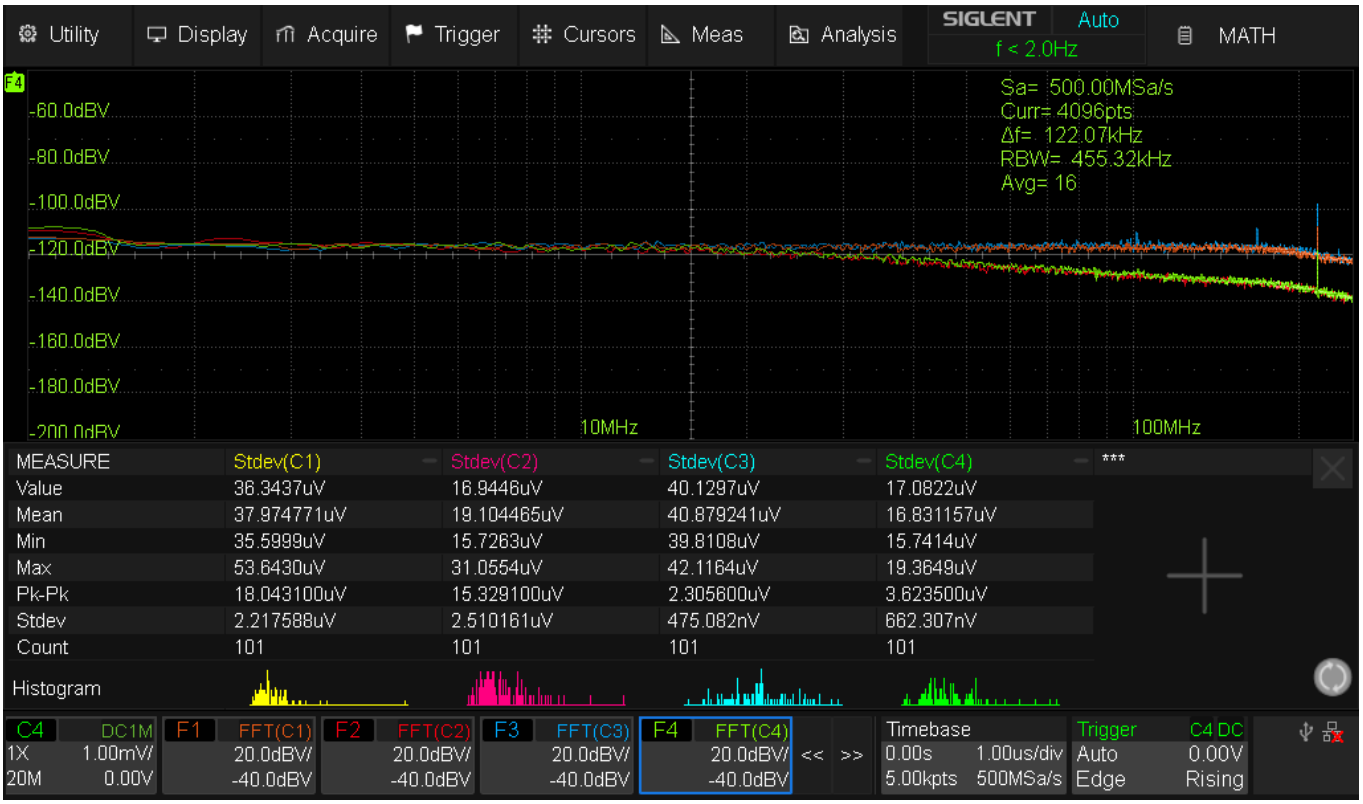

Fig. 29 SDS824X_HD_Noise_100kHz-200MHz_4Ch

Compare this with 10 kHz lower bandwidth limit:

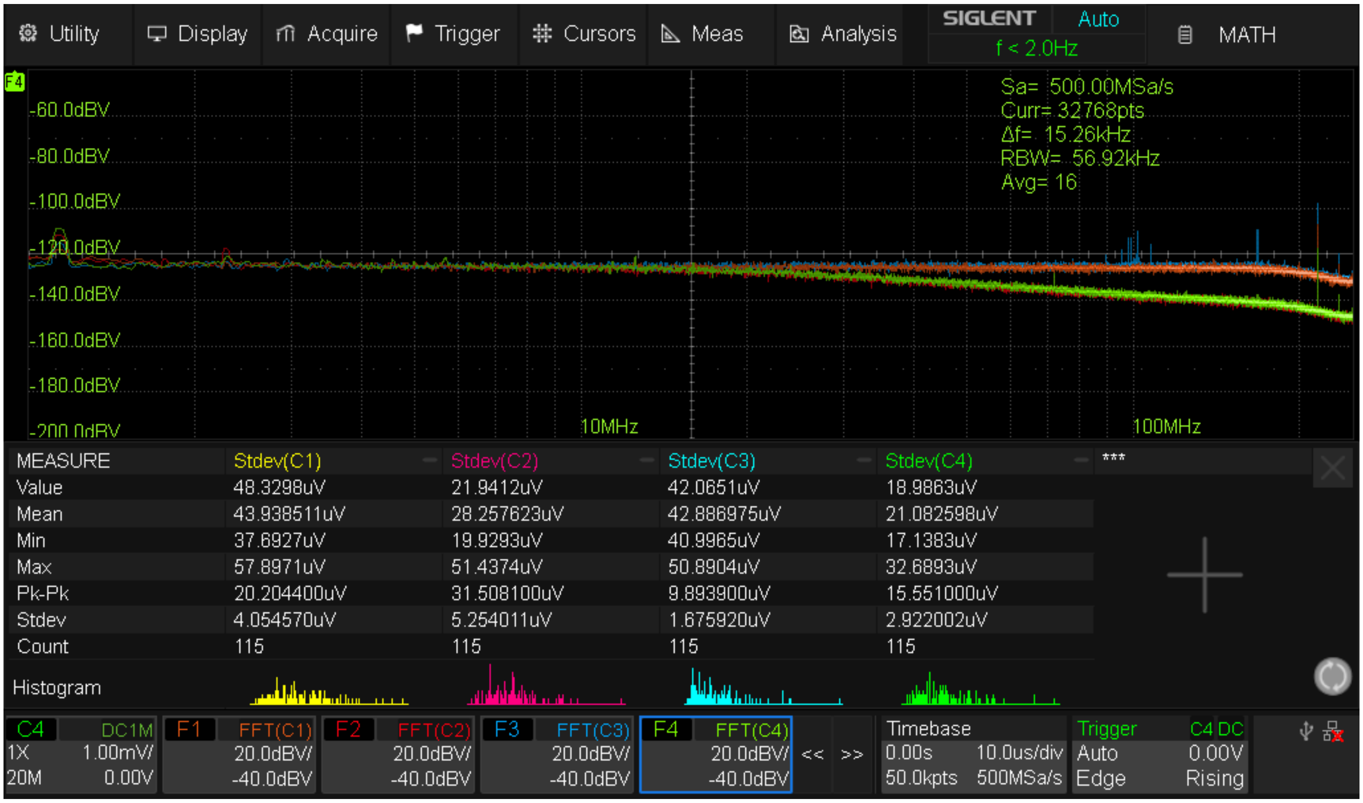

Fig. 30 SDS824X_HD_Noise_10kHz-200MHz_4Ch

1 kHz lower bandwidth limit:

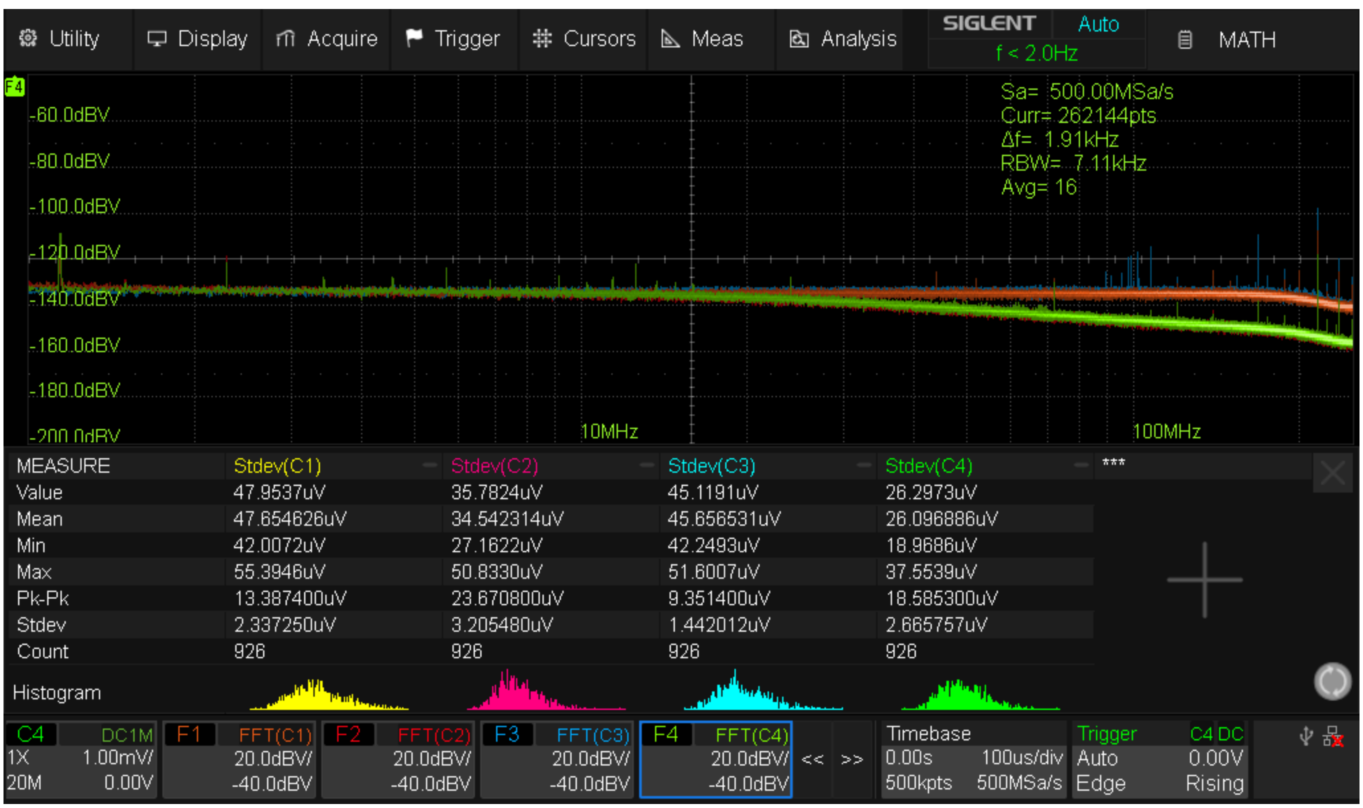

Fig. 31 SDS824X_HD_Noise_1kHz-200MHz_4Ch

Finally, 200 Hz lower bandwidth limit:

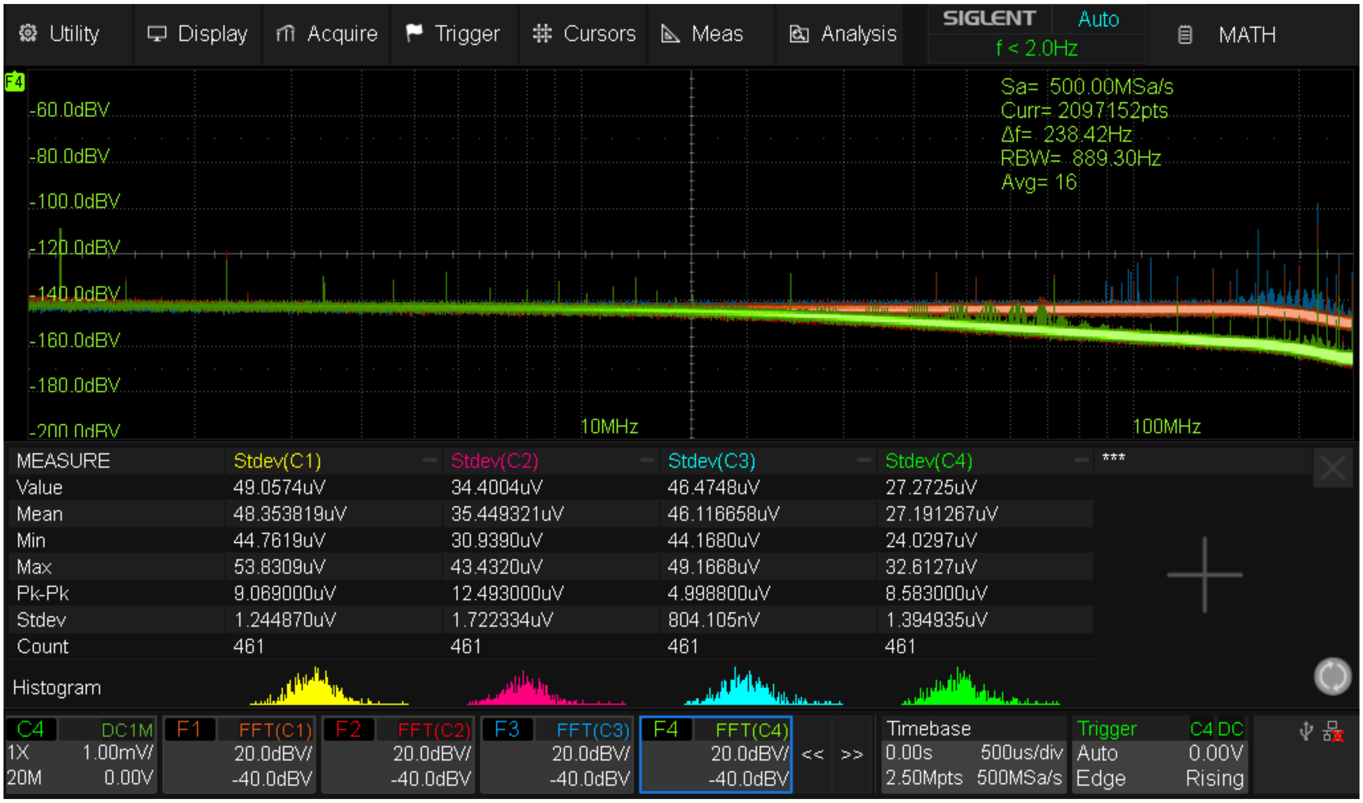

Fig. 32 SDS824X_HD_Noise_200Hz-200MHz_4Ch

Here you can see the noise characteristic up to 25 MHz at a RBW of ~90 Hz. In order to prevent aliasing taking effect, the noise measurement has been done on a 20 MHz bandwidth limited input channel. Since this bandwidth limiter is only first order, a digital 30 MHz lowpass filter has been added.

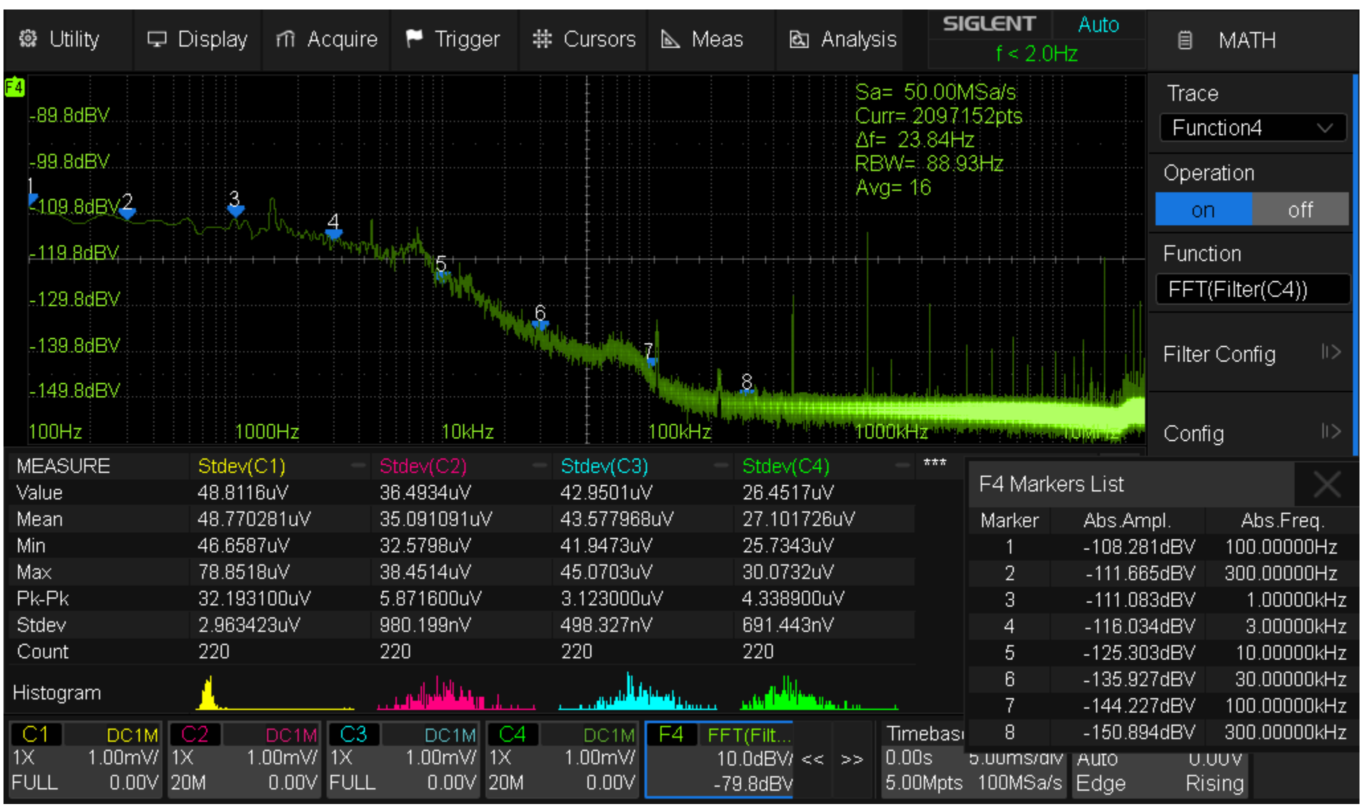

Fig. 33 SDS824X_HD_Noise_20Hz-25MHz_4Ch_F30M

The Noise characteristic is a combination of the 1/f noise of the MOSFET input buffer in the HF path and the FET OpAmp in the LF path, which is fed with a heavily attenuated signal that has to be amplified again before the recombination of both paths.

We can clearly see this in the previous screenshot:

At 300 kHz, the measured noise level is -150.894 dBV, this corresponds to a noise density of just 3 nV/√Hz at a RBW of 89 Hz. The following table shows the complete measurements:

300 kHz:	-150.894 dBV	3.0 nV/√Hz
100 kHz:	-144.227 dBV	6.5 nV/√Hz
30 kHz:	-135.927 dBV	16.9 nV/√Hz
10 kHz:	-125.303 dBV	57.5 nV/√Hz
3 kHz:	-116.034 dBV	167.2 nV/√Hz
1 kHz:	-111.083 dBV	295.6 nV/√Hz
300 Hz:	-111.665 dBV	276.5 nV/√Hz
100 Hz:	-108.281 dBV	408.2 nV/√Hz

At 10 MHz, the noise density has dropped to about 2.4 nV/√Hz.

Finally let’s have a look at the spurious signals (CH.4, 50 Ω termination, 20 MHz bandwidth limit):

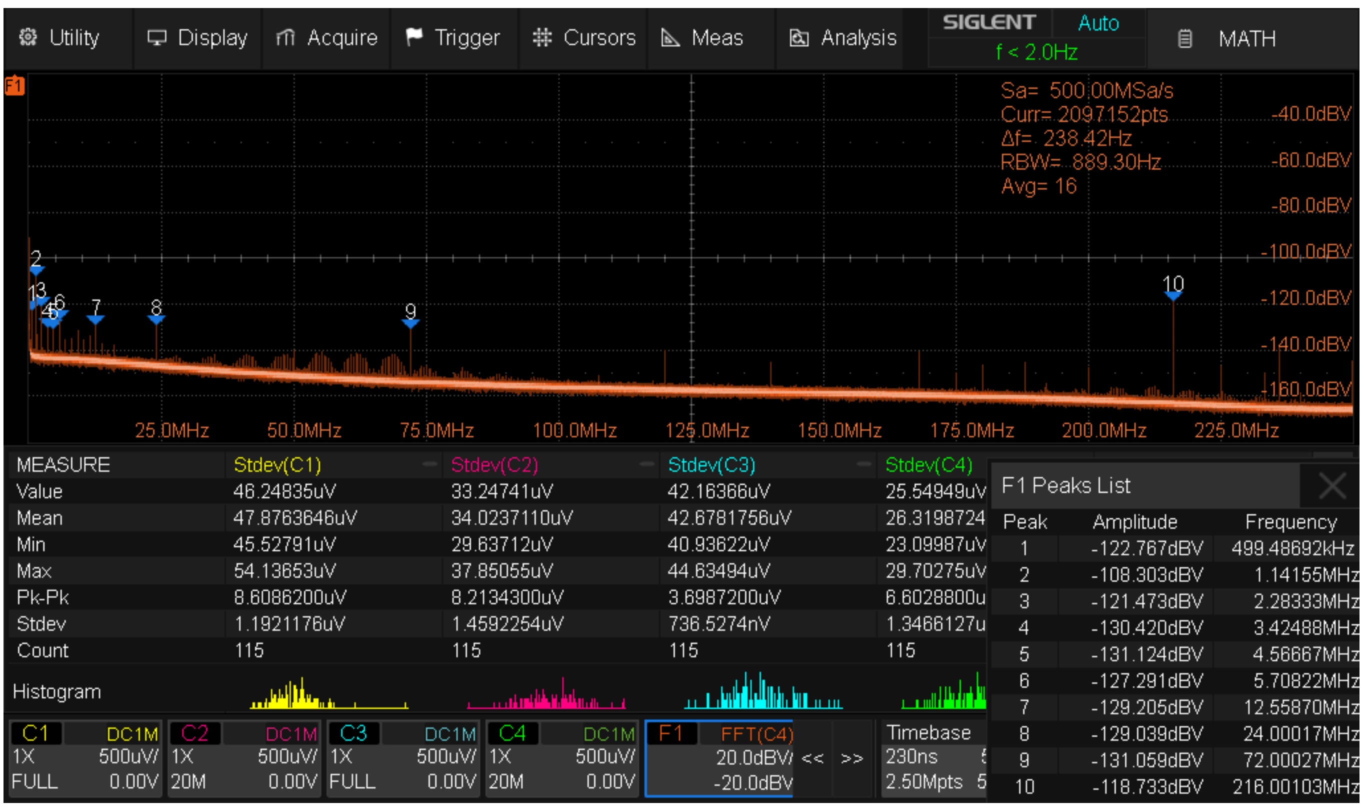

Fig. 34 SDS824X_HD_Spurs_200Hz-200MHz_4Ch

The Peaks List shows the 10 strongest spurious signals, where all are at or below 1 µV$_{RMS}$, except for a single spur at 1.14155 MHz, which is 3.85 µV$_{RMS}$. This is exceptionally good, especially in this class.

Noise Density (1/f)

In my previous test I've used a constant sample rate of 100 MSa/s, which allowed a 2 Mpts FFT with an effective FFT-sample rate of 50 MSa/s and Δf = 23.84 Hz. This was required, since the FFT in the SDS800X HD is limited to 2 Mpts max. and I wanted to measure the 1/f noise down to at least 100 Hz. As a consequence, I had to set up the FFT in a way that I get a frequency step (Δf) well below that.

Of course, if we want any accuracy in the spectrum plot, the Flattop window has to be used, and the RBW is Δf * 3.73 in case of Siglent's version of the Flattop Window.

Because of the low sample rate of just 100 MSa/s for the acquisition, there will inevitably be aliasing, folding back all the noise above 50 MHz to the first Nyquist zone. Then there will be even more aliasing because the FFT introduces one more decimation step, from 100 to 50 MSa/s. The latter could be countered by a digital filter, but it doesn't make that much of a difference anymore.

All this does not matter much as long as we are mainly interested in the 1/f noise below about 300 kHz, because it is much stronger than the high frequency noise anyway.

Now we want to see the real noise density up to 10 MHz without any aliasing spoiling our measurements. For this we can activate all channels, thus reducing the input bandwidth to a well-defined 200 MHz and engage the 20 MHz bandwidth limiter on top of that, so that we can be absolutely sure that there will be no aliasing products of any significance affecting the measurement at 10 MHz.

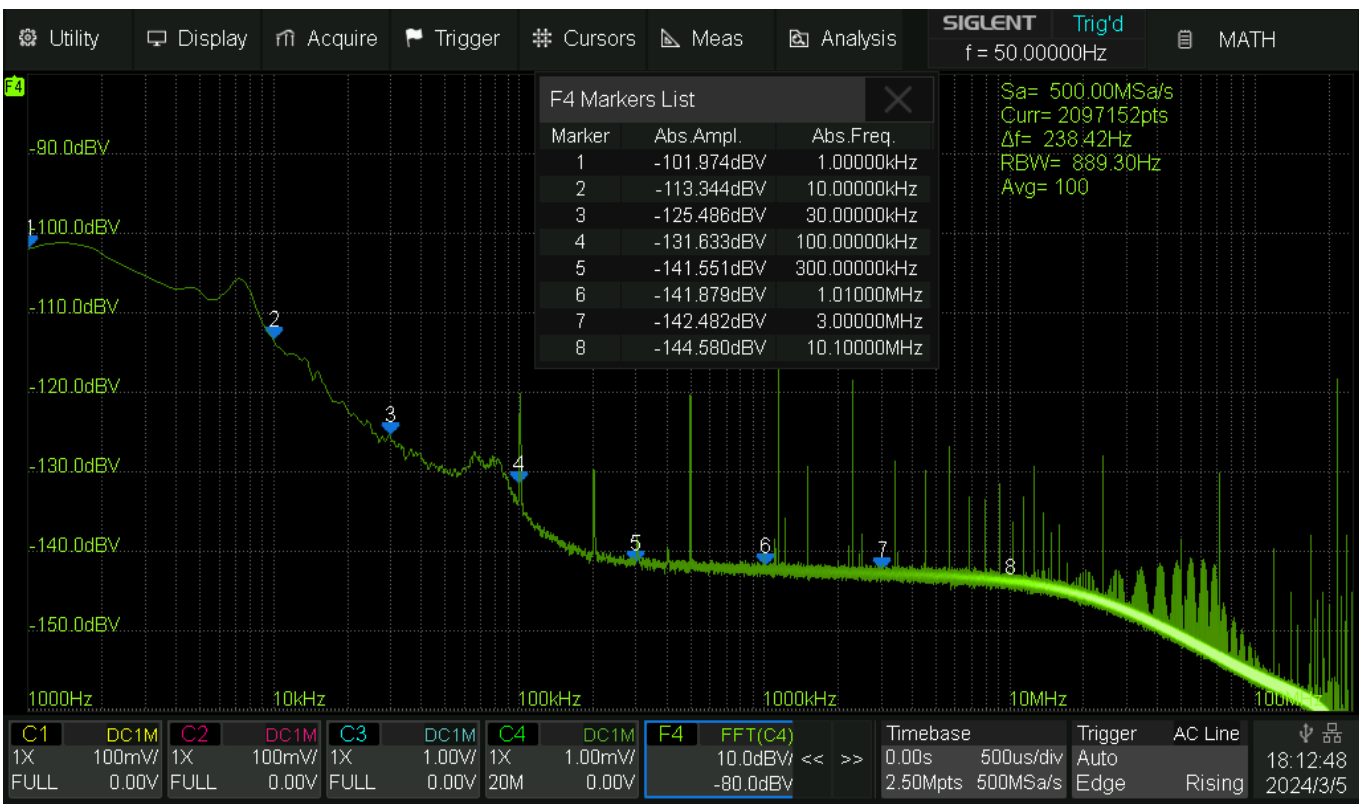

Fig. 35 SDS824X HD_ND_1mV_20MHz_500MSa

Calculation for 10 MHz: -144.58 dBV = 59 nV_{RMS}.
The noise density at this point is 59 nV / √889.3 Hz = 59 nV / 29.8 = 1.98 nV/√Hz;

Here is the complete table:

10 MHz:	-144.58 dBV	2.0 nV/√Hz
3 MHz:	-142.48 dBV	2.5 nV/√Hz
1 MHz:	-141.88 dBV	2.7 nV/√Hz
300 kHz:	-141.55 dBV	2.8 nV/√Hz
100 kHz:	-131.63 dBV	8.8 nV/√Hz
30 kHz:	-125.49 dBV	17.8 nV/√Hz
10 kHz:	-113.34 dBV	72.2 nV/√Hz
1 kHz:	-101.97 dBV	267.2 nV/√Hz

With a noise density near 2 nV/√Hz, the Siglent SDS824 X HD beats most of the competition at higher frequencies, whereas the 1/f noise is nothing to write home about, but that has to do with the special split path input buffer design with its enormous offset compensation capability (±8 V starting at only 10.2 mV/div!).

PS: Of course, the above measurement was flawed, because the 20 MHz bandwidth limiter affects the 10 MHz measurement. The actual noise density, measured without bandwidth limit at 10 MHz is 2.4 nV/√Hz, just as it was stated in the first test.

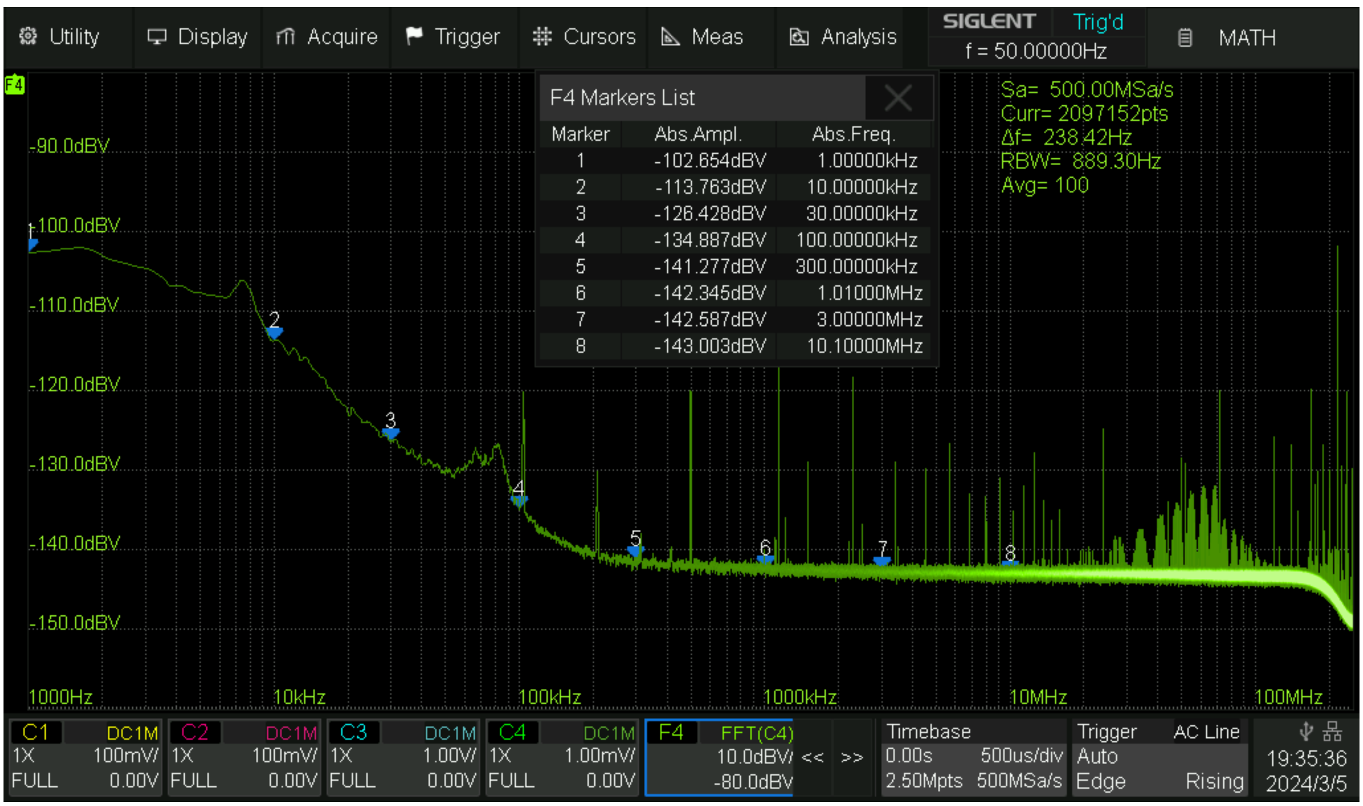

Fig. 36 SDS824X HD_ND_1mV_200MHz_500MSa

Here is the updated noise density table:

10 MHz:	-144.58 dBV	2.4 nV/√Hz
3 MHz:	-142.48 dBV	2.5 nV/√Hz
1 MHz:	-141.88 dBV	2.6 nV/√Hz
300 kHz:	-141.55 dBV	2.9 nV/√Hz
100 kHz:	-131.63 dBV	6.0 nV/√Hz
30 kHz:	-125.49 dBV	16.0 nV/√Hz
10 kHz:	-113.34 dBV	68.8 nV/√Hz
1 kHz:	-101.97 dBV	247.0 nV/√Hz

A noise density of <2.4 nV/√Hz is still one of the best in the industry.

Attached is the binary data file for this measurement.

SDS824X_HD_Binary_C4_3.7z
Channel 4, 1 mV/div, 50 ohms terminated;
500 µs/div, 2.5 Mpts, 500 MSa/s;
Full Bandwidth;

Noise Density

This time I've decided to put not so much weight on the 1/f noise at really low frequencies, but do a flawless measurement where we can rule out any aliasing artefacts affecting the numbers.

I used only a single channel (Ch. 4), hence a sample rate of 2 GSa/s, which permits a 2 Mpts FFT with an effective FFT-sample rate of 2 GSa/s and Δf = 953.67 Hz, resulting in 3.56 kHz RBW with the Flattop window. This allows us to measure the 1/f noise down to at least 10 kHz and guarantees full accuracy up to 1 GHz.

Now we want to measure the real noise density up to 100 MHz without any aliasing spoiling our results.

Fig. 37 SDS824X_HD_ND_2GSa_1mV

Calculation for 10 MHz (I've used 9.9 MHz to escape a micro-spur): -137.35 dBV = 135.68 nV_{RMS}.

The noise density at this point is 135.68 nV / √3560 Hz = 135.68 nV / 59.66 = 2.27 nV/√Hz;

Here is the complete table:

100 MHz:	-137.73 dBV	2.18 nV/√Hz
10 MHz:	-137.35 dBV	2.27 nV/√Hz
3 MHz:	-136.64 dBV	2.47 nV/√Hz
1 MHz:	-136.54 dBV	2.50 nV/√Hz
300 kHz:	-135.01 dBV	2.98 nV/√Hz
100 kHz:	-128.72 dBV	6.14 nV/√Hz
30 kHz:	-120.57 dBV	15.70 nV/√Hz
10 kHz:	-104.51 dBV	99.67 nV/√Hz

We are getting pretty close to 2 nV/√Hz at frequencies of 10 MHz and higher.

Granular Noise

This time we shall have a look at the granular noise of the 12-bit SDS800X HD. At high sensitivities like 1 mV/div, we cannot expect much of an advantage from the 12 bits, but at low sensitivities like 50 mV/div and higher, the 12 bits should clearly give us a benefit.

I used only a single channel (Ch. 4), hence a sample rate of 2 GSa/s, which permits a 2 Mpts FFT with an effective FFT-sample rate of 2 GSa/s and Δf = 953.67 Hz, resulting in 3.56 kHz RBW with the Flattop window. This allows us to measure the noise down to at least 10 kHz and guarantees full accuracy up to 1 GHz.

Fig. 38 SDS824X_HD_ND_2GSa_1V

There's little point in calculating the traditional noise density (which would have to be specified in hundreds of nanovolts or even single digit microvolts per √Hz), but we should consider the full-scale value of +9 dBV in this test scenario.

We get a SNR of >110 dB at 100 MHz with a RBW of 3.56 kHz.

You can download a 100 Mpts binary file for this test scenario:

SDS824X HD
Channel 4, 1V/div, 50 ohms terminated;
5 ms/div, 100 Mpts, 2 GSa/s;
Full bandwidth = 245 MHz;

SDS824X HD_Noise_1V_1ms_2GSa_245MHz

Display

Vertical Zoom Demo

Vertical zoom can suffer from noise, if high zoom factors are used. Some demonstrations use bandwidth limits to reduce noise when zooming in. It is in the responsibility of the user then to make sure that no relevant high frequency detail gets lost by this.

In general, the question remains: what if we need to look at higher frequencies? A 200 MHz 12-bit DSO should be able to demonstrate a resolution advantage with 200 MHz bandwidth signals just as well…

Here is the signal mix: a 1 MHz 600 mV_{PP} sine with a 200 MHz 10 mV_{PP} sine riding on it:

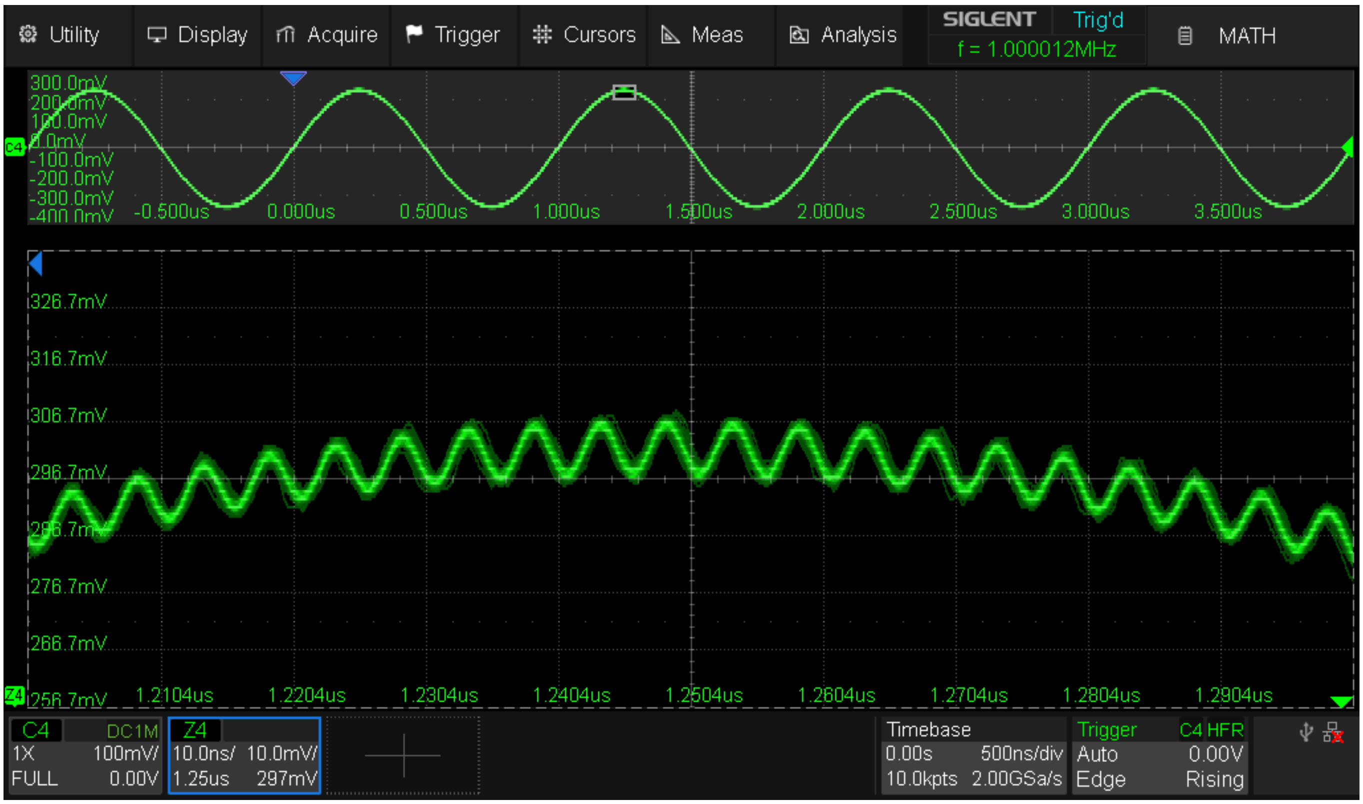

Fig. 39 SDS824X_HD_VZ10x_Run

I've chosen straight 10 mV/div for the zoom window, i.e. a ten times magnification. The superimposed waveform is a little noisy, yet clearly visible.

This is run mode. In stop mode, we can see that all the noise is rather low frequency and lowering the bandwidth wouldn't help anyway:

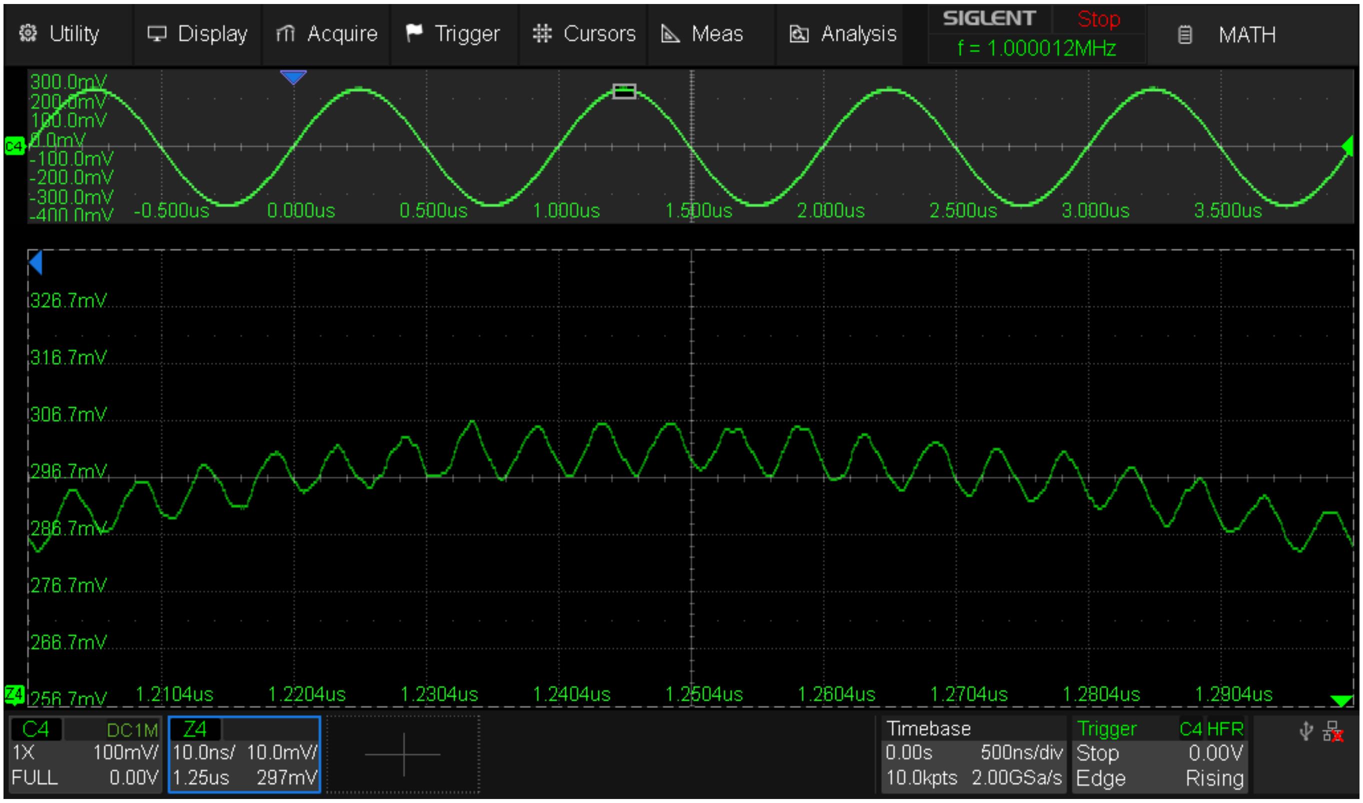

Fig. 40 SDS824X_HD_VZ10x_Stop

In stop mode we basically get a clean waveform with some distortion. Yet this is just 12 bits without any additional tricks.

We can use the average math function to get rid of noise and modulation:

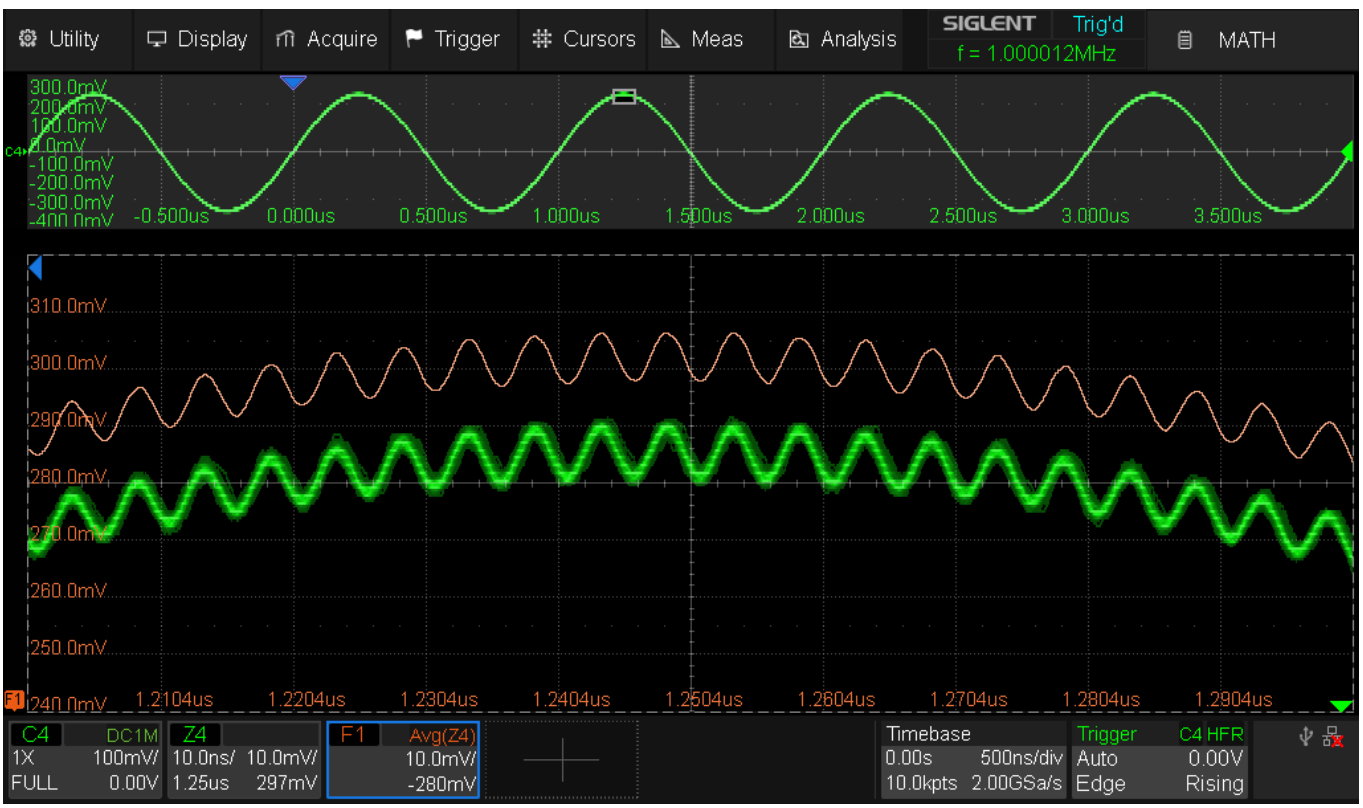

Fig. 41 SDS824X_HD_VZ10x_Avg16

16 times average (Math trace F1) is enough to get the waveform pretty clean also in run mode. The implicit resolution enhancement of this measure is 4 bits, so that the DSO is effectively working with 16-bit data now.

Vertical axis labels

The new kids in town like the SDS800X HD also bring new features: apart from the logarithmic frequency axis for the FFT, as demonstrated in the noise measurements, we also got a selectable vertical label position.

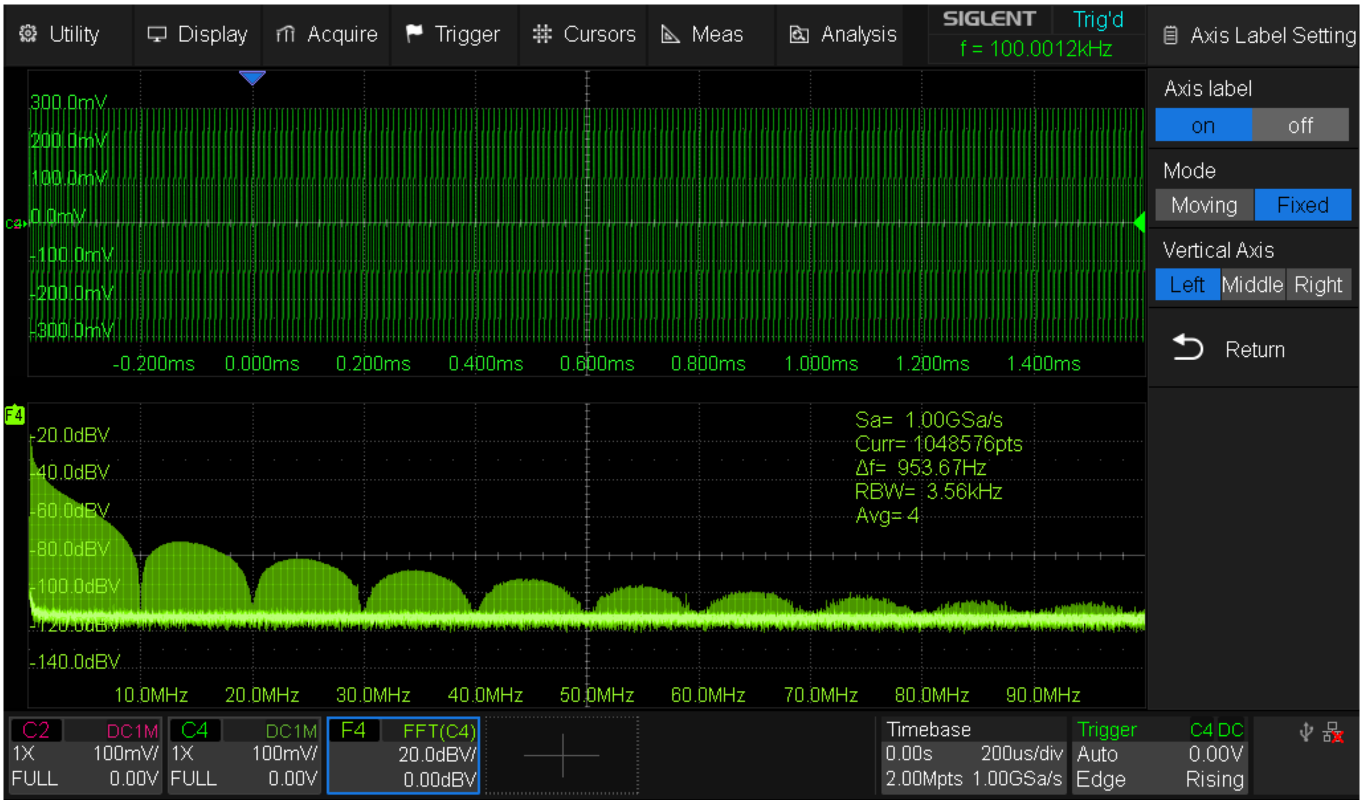

Fig. 42 SDS824X_HD_VLabel_left

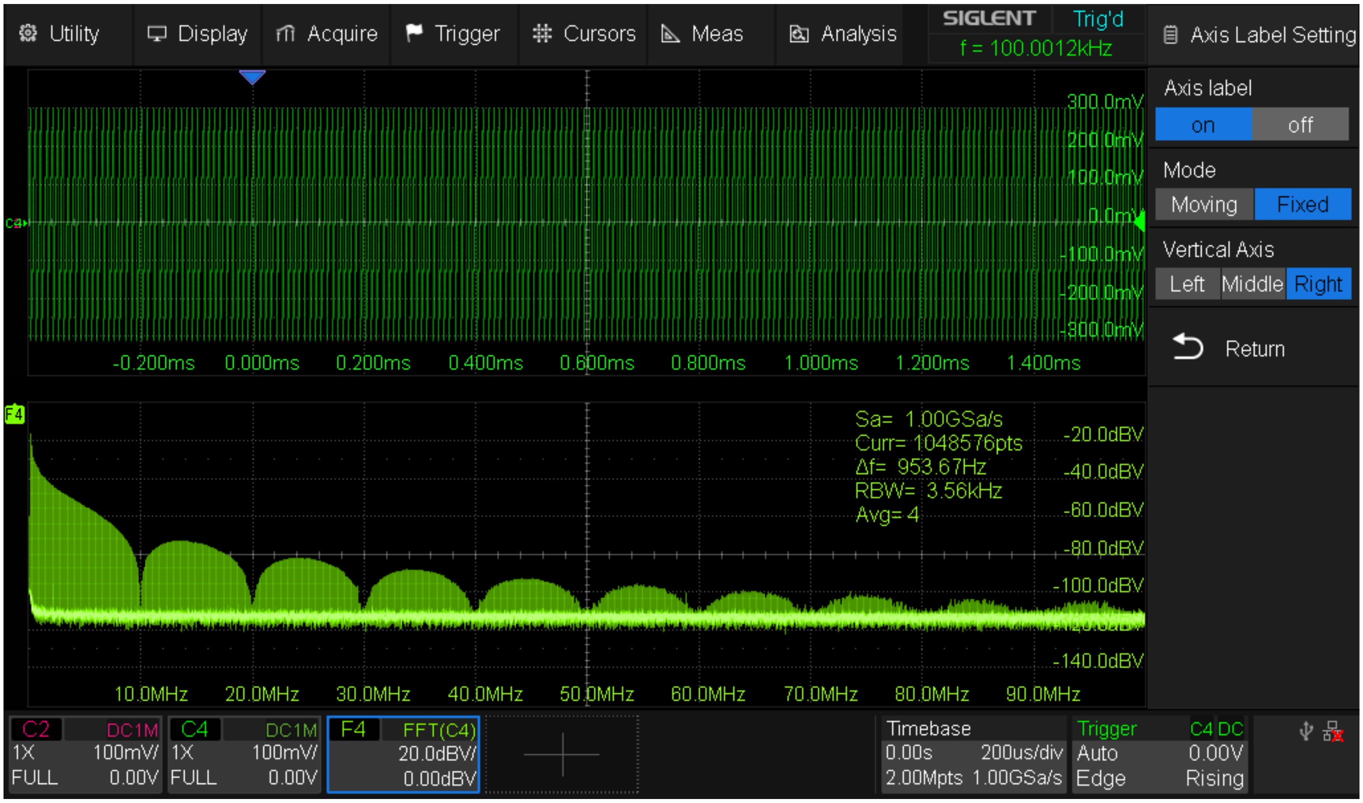

Fig. 43 SDS824X_HD_VLabel_right

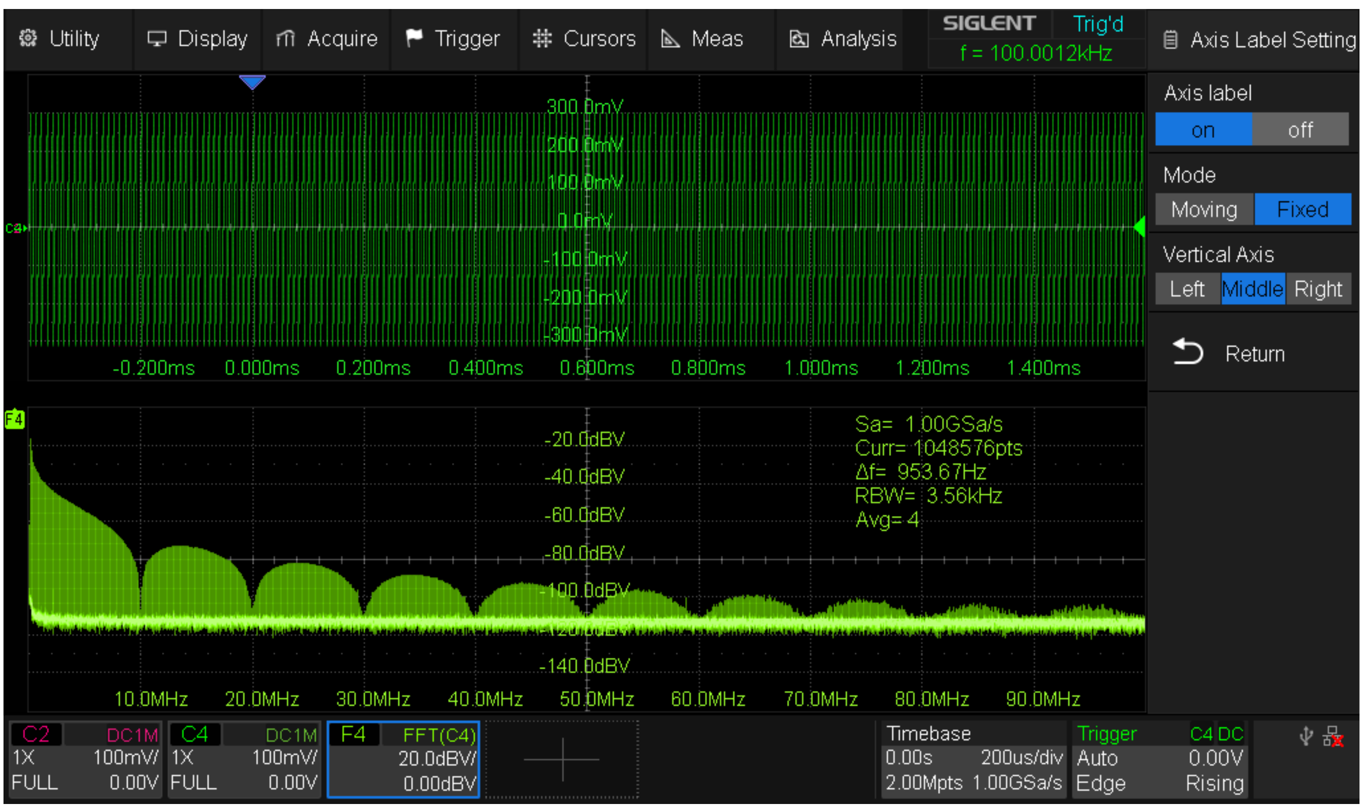

Fig. 44 SDS824X_HD_VLabel_center

Zoom Expectations

When using a 12-bit DSO, some people tend to get enthusiastic and expect miracles. Maybe we all should come back to earth and ask ourselves what we can realistically expect.

Consider a signal that has to be viewed at a vertical gain of 1 V/div in order to fit on the screen:

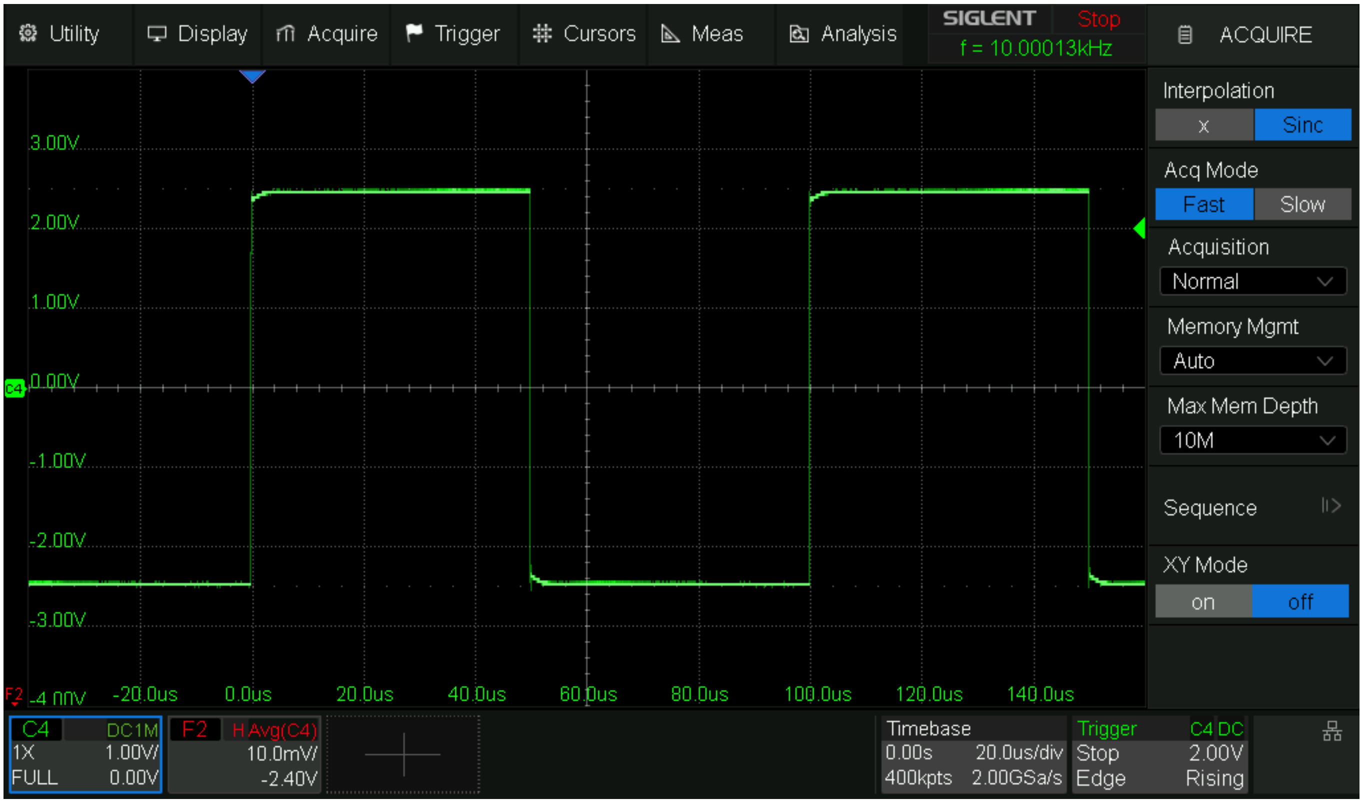

Fig. 45 SDS824X HD_PR_H50ns_Stop

Now we want to zoom in, e.g. 5x:

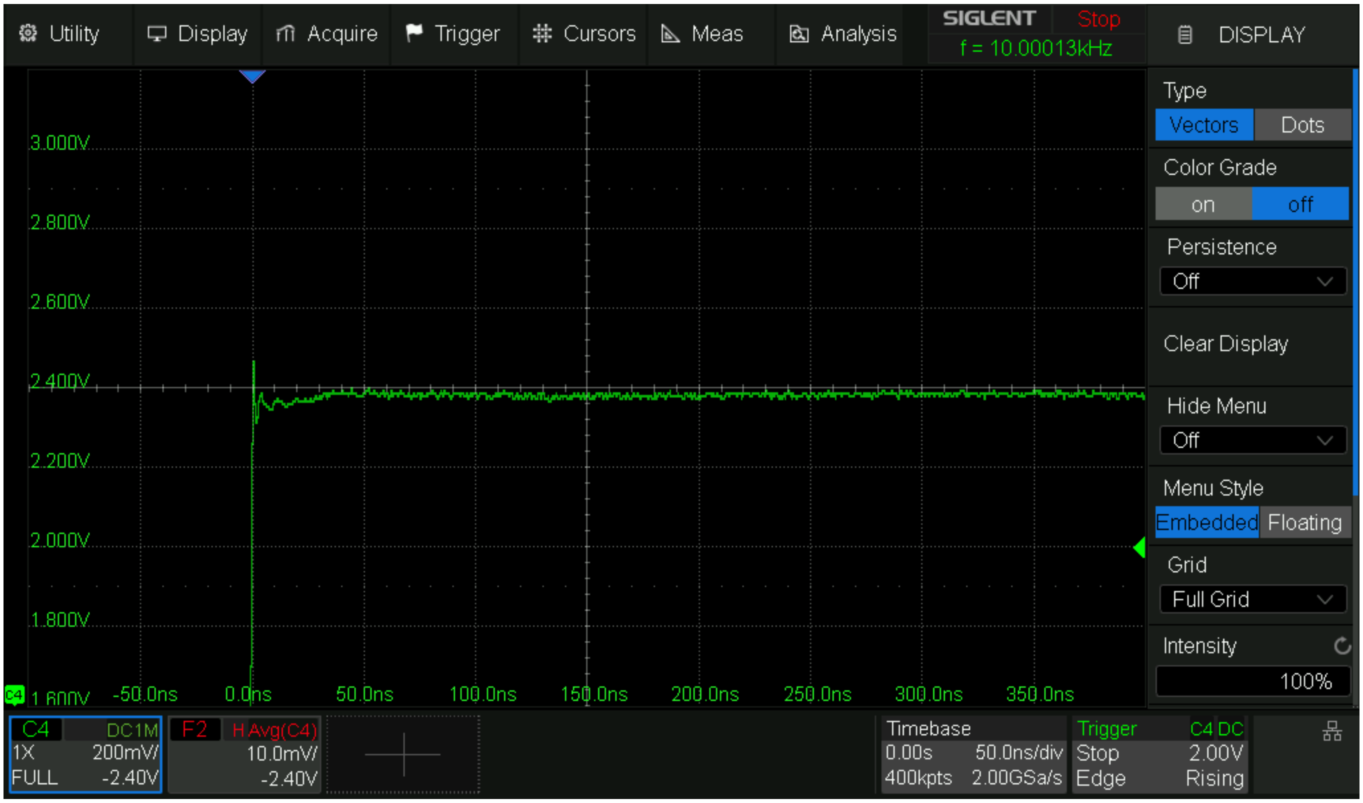

Fig. 46 SDS824X HD_PR_H50ns_Stop_Zoom5

We already see a bit of noise creeping in. Apart from that, we should determine how much zoom is feasible, before we try to zoom in any further:

Just like the SDS800X HD, an SDS1000X HD will have 480 LSB (aka codes) per division. Consequently, 20x zoom is about the sensible limit, because then we get 24 LSB per division – and with this, there would still be a chance to see something meaningful. 20x zoom means 1 V / 20 = 50 mV/div:

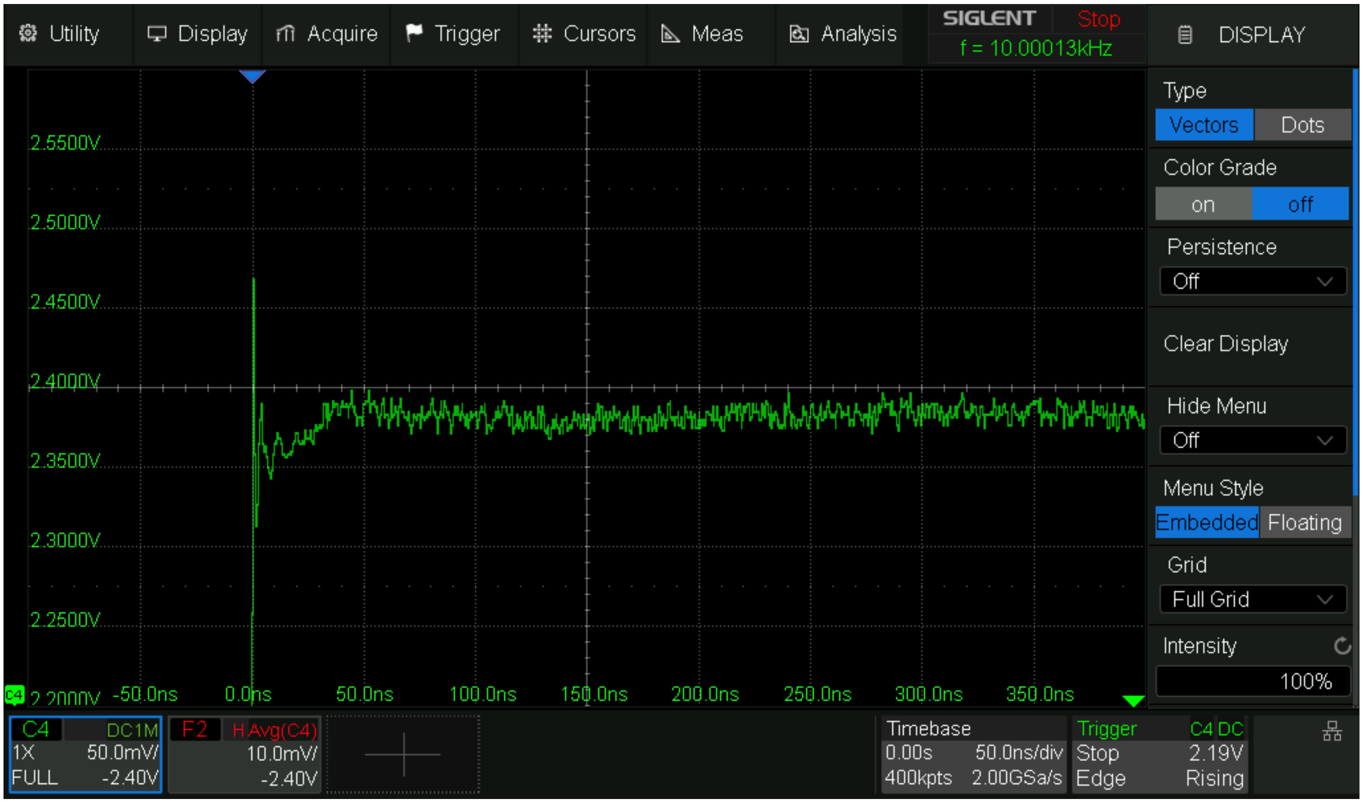

Fig. 47 SDS824X HD_PR_H50ns_Stop_Zoom20

Now the noise is stronger and it already gets hard to spot any details. Yet we can take it to the extreme and try 100x zoom:

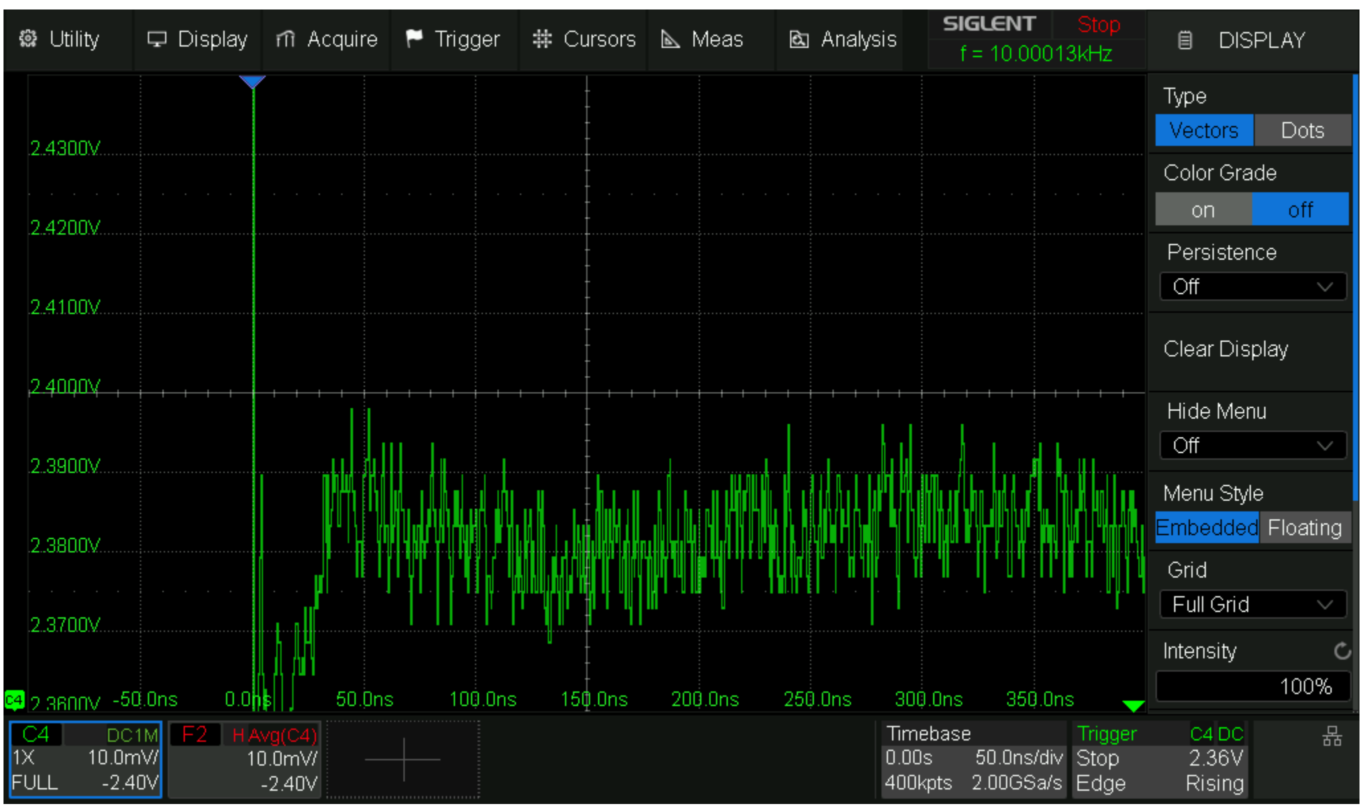

Fig. 48 SDS824X HD_PR_H50ns_Stop_Zoom100

We are now at 4.8 codes per division and what we see is just (mostly granular) noise – all this has nothing to do with the real signal anymore. Most obvious when viewing it in Dots mode.

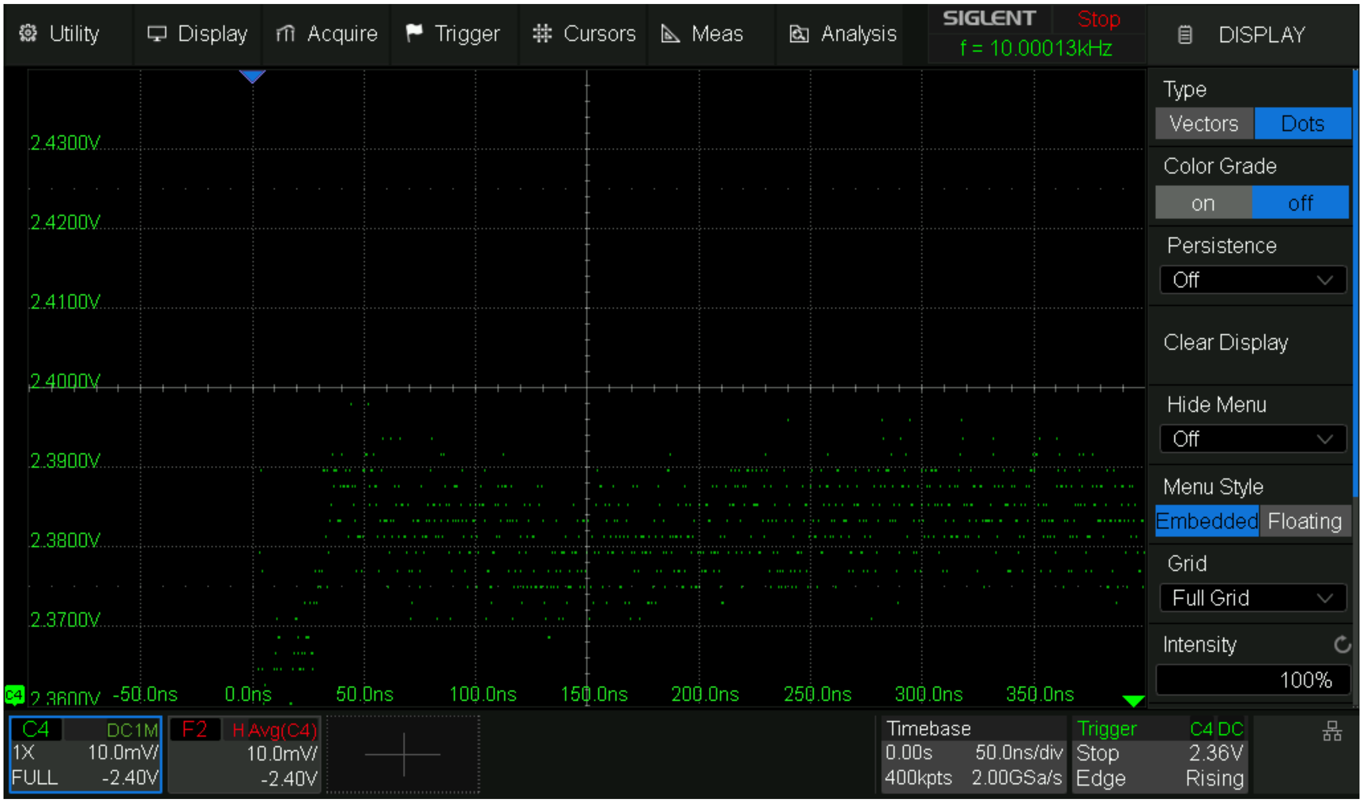

Fig. 49 SDS824X HD_PR_H50ns_Stop_Zoom100_Dots

We can take it one step further and zoom in horizontally as well (while still in Vectors mode), so that it becomes less obvious that we're actually looking at pure noise:

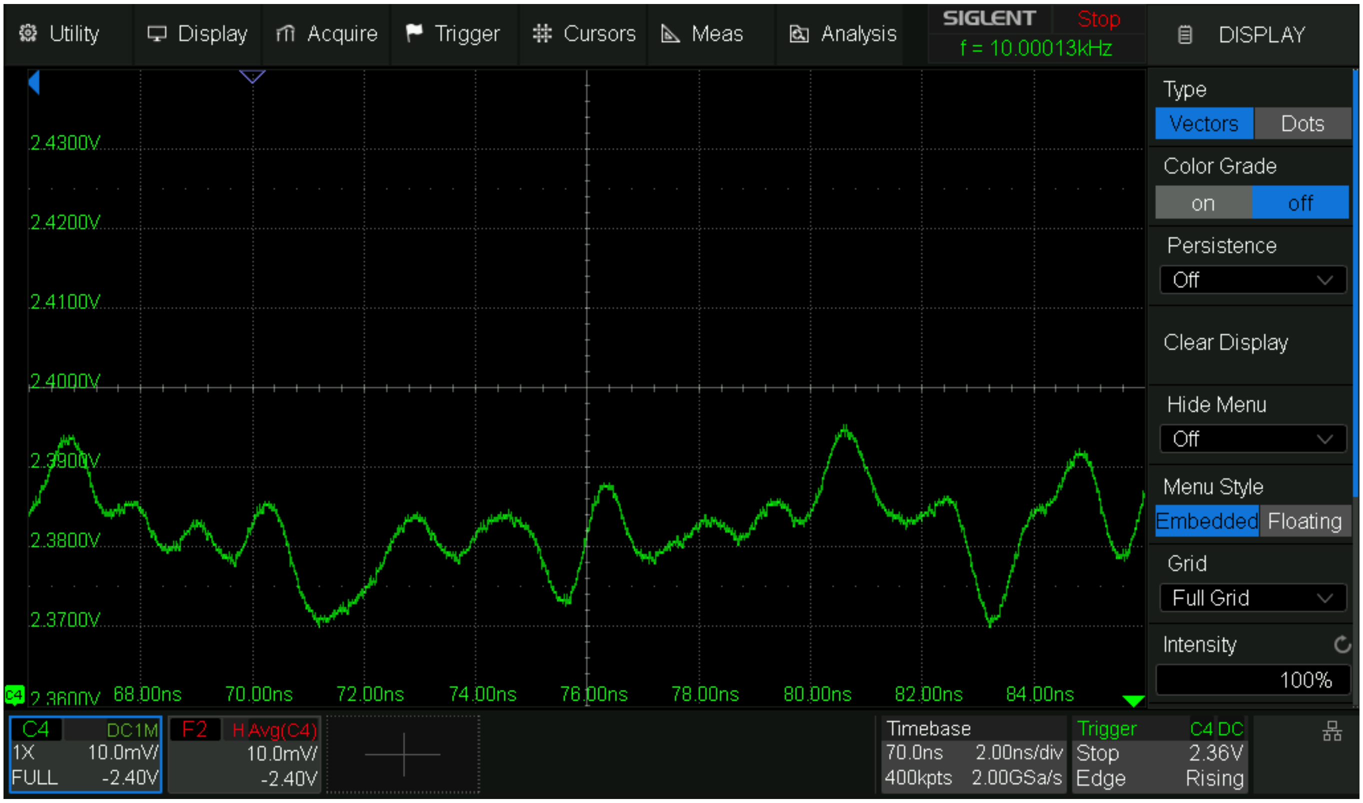

Fig. 50 SDS824X HD_PR_H2ns_Stop_Zoom100

As can be seen, the SDS800X HD doesn't show any Sinc artifacts, yet the much more important insight should be that it is completely irrelevant what a DSO shows at such extreme zoom levels, where we see nothing but noise anyway.

Of course, we could also do it the correct way. Whenever we need to use some extreme zoom, then we also need a means to

a) Increase the vertical resolution
b) Reduce the noise

The tool of choice for repetitive signals is Averaging, because it increases the vertical resolution, suppresses any modulation (hence also noise) and doesn't affect the bandwidth.

Even at the extreme 100x zoom setting we can get the following:

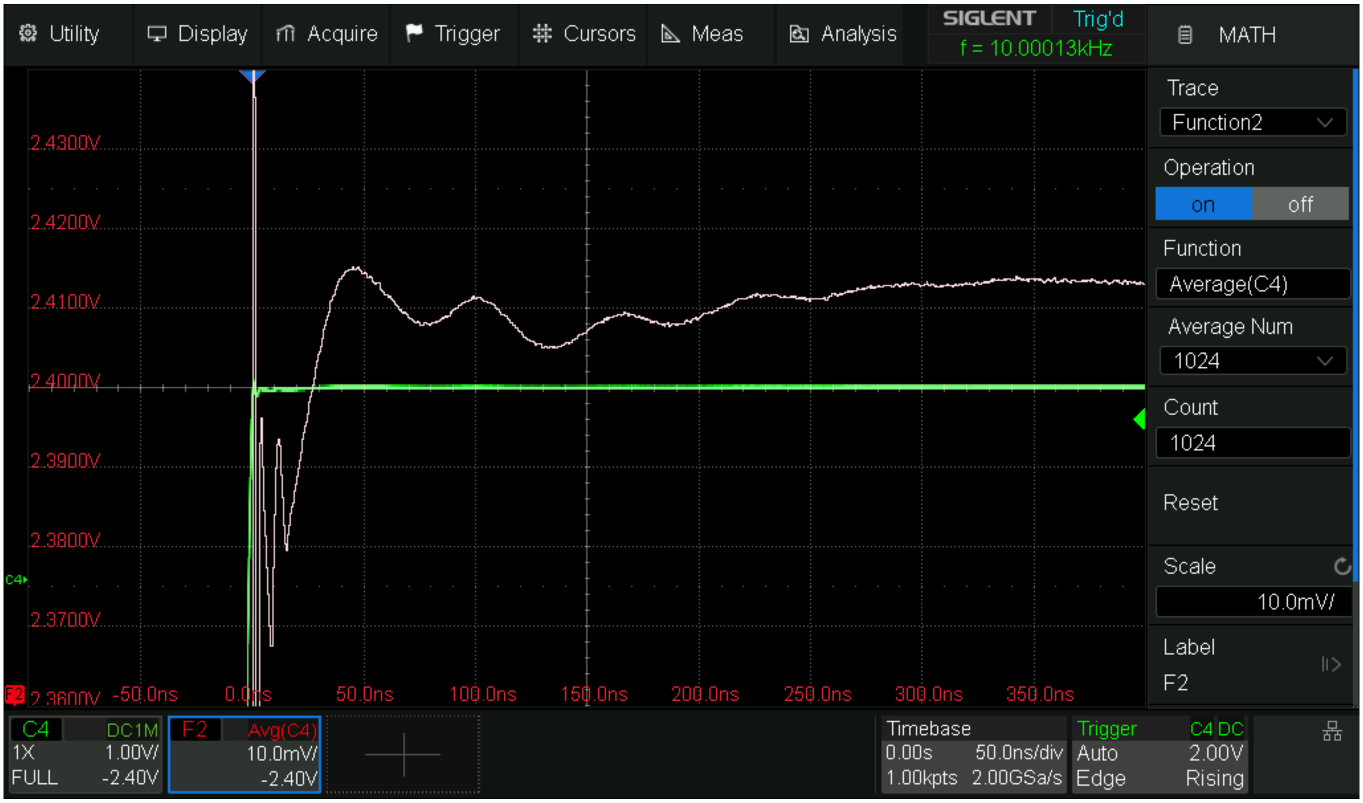

Fig. 51 SDS824X HD_PR_H50ns_Stop_Zoom100_Avg1024

We leave the input channel at its original gain of 1 V/div, but set the final time base while still in Run mode and set up a math trace with averaging, which can be displayed at any vertical scale, i.e. also 10 mV/div for a 100x zoom. When the desired number of averages has been processed, we can stop the acquisition, yet this is not required, as the input channel is left unchanged anyway. In this example it is 1024 averages shown at 10 mV/div, hence a 100x zoom again.

Now compare this with the previous screenshots. This one now has much less curves and kinks than even the pointless capture at 2 ns/div before, proving that it's just nonsense (=noise) what we get at such zoom levels without proper averaging.

Of course, 1024 averages are quite a lot. In theory, it would enhance the resolution by 10 bits, making for a total of 22 bits. The current platform doesn't support sample data and digital signal processing results at more than 16 bits resolution, so this is what we get as soon as we use 16x or higher averaging. The ENOB on the other hand benefits a lot more from this, even though it's limited to 16 bits as well. But it starts at just 8.4 bit according to the data sheet, and 1024x averaging will increase this by 5 bits for a total of ~13.4 bits.

Zoom Challenge

Some folks have the need for a high dynamic range, i.e. the ability to inspect small details in a signal. To accomplish this, they usually increase the vertical gain of the DSO and use the position control to center the region of interest on the screen. This way, even 8-bit oscilloscopes can display some detail – as long as the signal distortions, caused by overdriving the oscilloscope frontend, don't affect the displayed portion of the signal too much. The distortions are especially bad with general purpose oscilloscopes, as they use the well-known split path input buffer with its problematic overload recovery behavior.

Now let’s examine our options with the Siglent SDS800X HD.

First, we could try to use the traditional technique in overloading the scope. Without too much thinking, we can just connect a strong signal and then “zoom in” by increasing the vertical gain of the oscilloscope.

In the following example we have a 2 Mbps PRBS-signal with 3 V amplitude connected directly, hence a 1x probe factor applies.

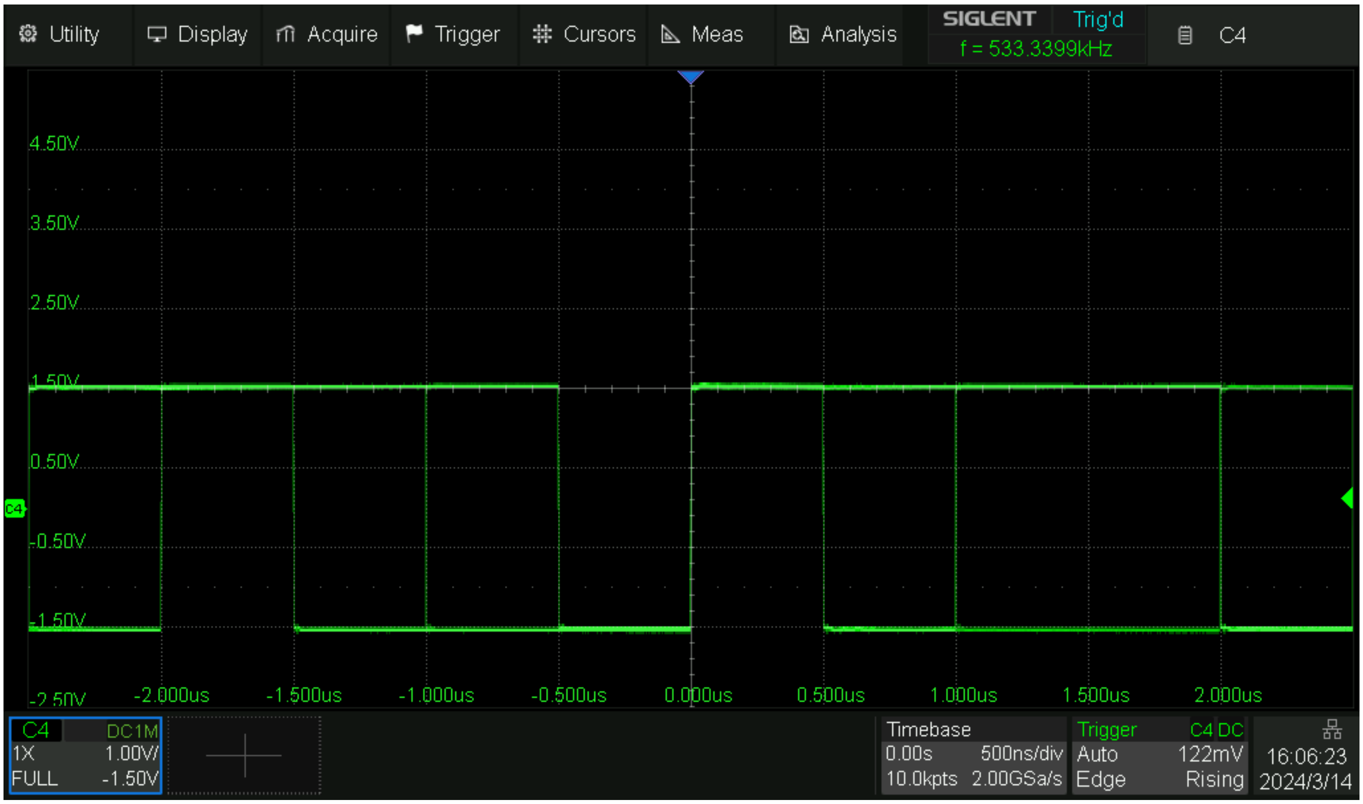

Fig. 52 SDS824X HD_PRBS-4_A3V_V1V_P1

Now we try to take a closer look at the pulse tops and increase the sensitivity. This works reasonably well down to 200 mV/div, but at 100 mV/div we hear a relay clicking and the signal gets distorted:

Fig. 53 SDS824X HD_PRBS-4_A3V_V100mV_P1

With a distorted signal like this, it makes no sense to try to look at any details in the signal. So, this obviously is the wrong approach.

For most applications, it is not the overload recovery of the semiconductor devices, like clamping diodes and transistors, which cause the problem. The overload recovery time of these devices is usually in the low (or even sub-) nanoseconds and is only really of concern in multi-GHz instruments.

Our problem is the clamping in the split-path input buffer, which causes clean clipping in the LF-path, but a differentiation of the waveform in the HF-path. When the clipped LF-path is recombined again with the both offset- and phase-shifted HF-path, the result is heavily distorted and has little similarity with the original signal.

Knowing all this, we are able to find a solution: just don't drive the input buffer so hard that the clamps get activated. Keep the input signal well below 1 V_{PP} by using 100x probes if necessary. This also has the advantage of a much lower capacitive load at the probe tip and the low noise of the SDS824X HD makes the use of x100 probes unproblematic.

The next screenshot demonstrates a 1 MHz square wave with 5V amplitude and a 10 mV_{PP} 40 MHz sine riding on it, using a ten times probe.

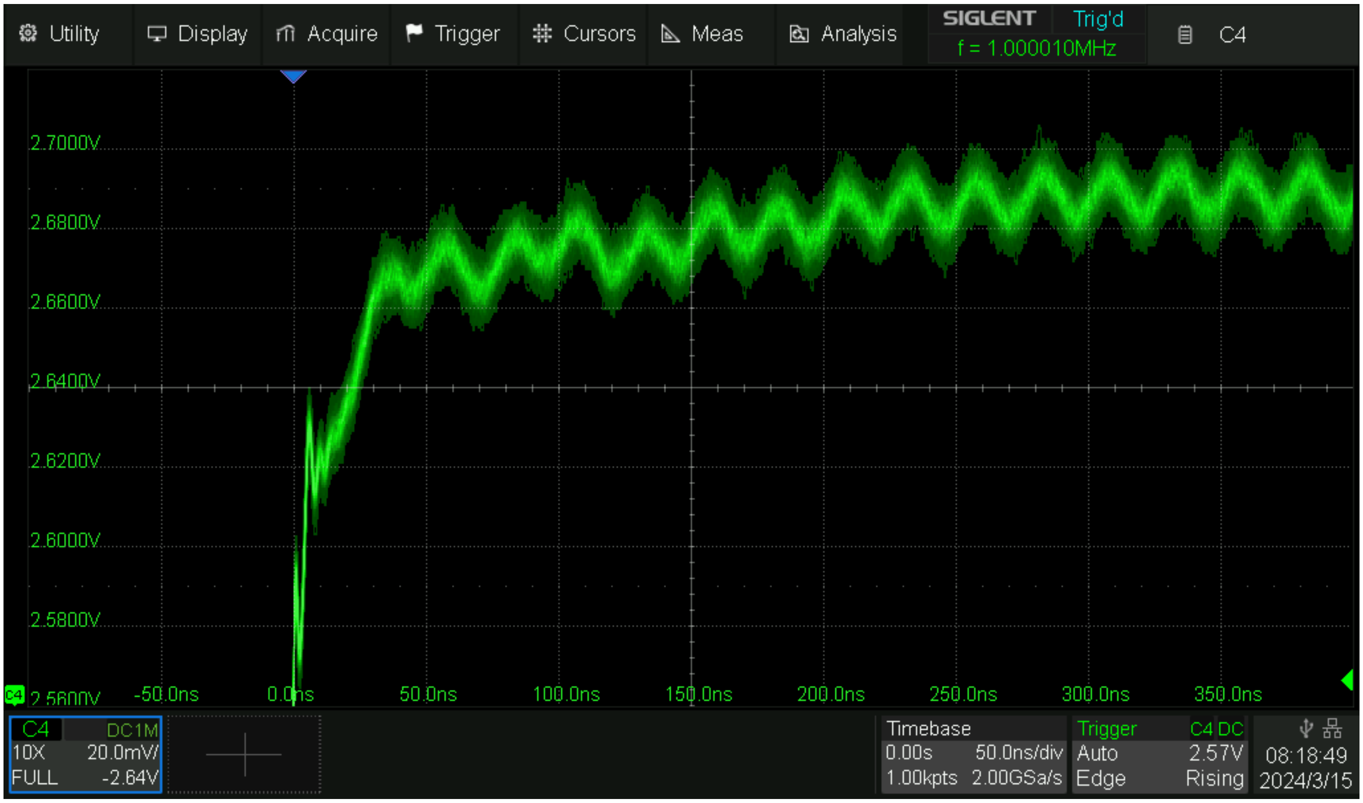

Fig. 54 SDS824X HD_OVD_5V_10mV_P10

Yes, the trace is noisy. It would be much better if we could use the 20 MHz bandwidth limiter – but unfortunately, this would also affect the 40 MHz signal we are interested in. Averaging would help a lot, but we want to be able to watch dynamic signals, hence it is not an option either.

We can still see the 40 MHz sine clear enough to know it is there – and that for a signal amplitude ratio of 1:500! That's what a low noise high resolution DSO can do for you…

There might be situations, when we just cannot get that low – maybe because the signal levels are so high that the output of even x100 probes would still exceed ~500 mV_{PP}. Then a combination of (moderately!) overdriving the scope and vertical zoom could be the best solution.

Consider a 1 MHz Square wave with 5 V amplitude – maybe as output of a x100 probe, so the original signal would be 500V - unbelievable, isn't it? It could be some 625 watt transmitter – but these wouldn't output a square wave and hopefully there wouldn't be any subtle signal details to observe, which would not be better analyzed by using the FFT, but I digress…

Here is that familiar 40 MHz sine wave again, riding on the square wave:

Fig. 55 SDS824X HD_Ref_5V_10mVpp

First step is to increase the vertical gain, i.e. dial in lower numbers, just before the relay would click. We could use the fine adjust to get 102 mV/div (because this is the highest gain we can get without changing the attenuator setting), but this shouldn't be necessary for now. We finally end up with 200 mV/div:

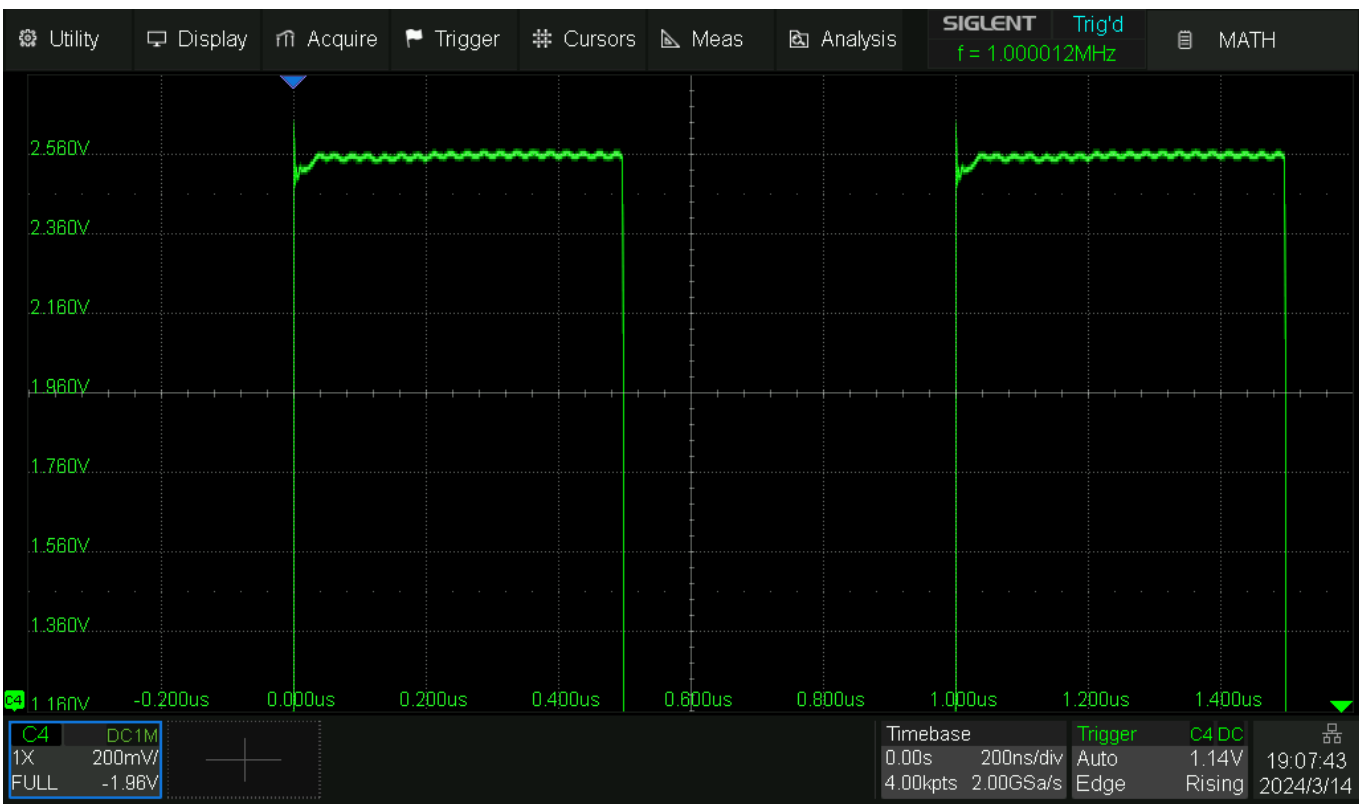

Fig. 56 SDS824X HD_OVD_limit_5V_10mVpp

We can already see the little wiggles on the top of the square, it is much smaller than the overshoot and ringing at the rising edge. Yet now we engage the Zoom mode to take a closer look:

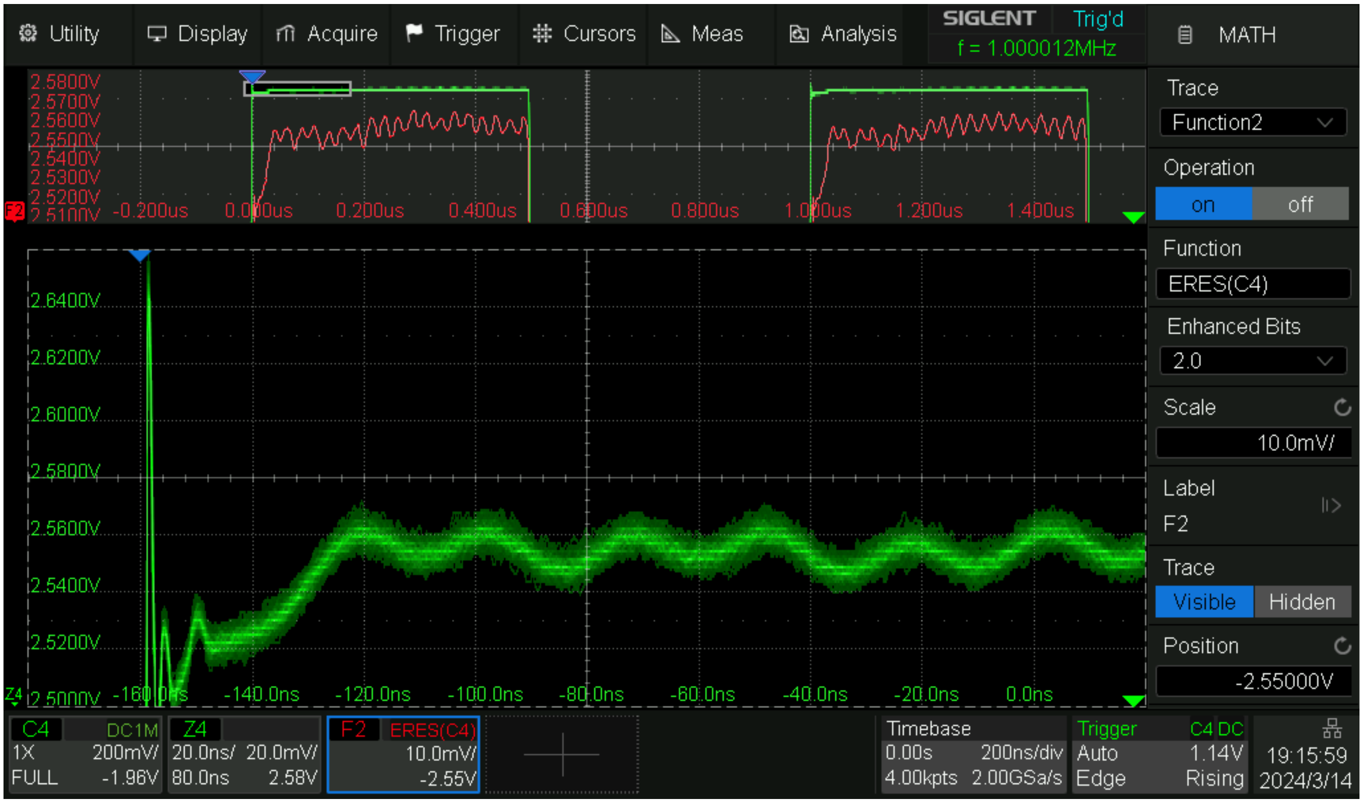

Fig. 57 SDS824X HD_OVD_limit_5V_10mVpp_Z20mV_ERES2.0

The above screenshot demonstrates two things: first is the ERES2.0 math trace in the main window, that lets us look at the 40 MHz sine at 10 mV/div. It is ugly, because ERES cannot get rid of the 1/f noise, so there's no use displaying it in more detail in the zoom window. But secondly, we have the regular trace in the zoom window at 20 mV/div, which is at least as clear as the overdrive zoom before.

One more time it should be remembered that we have a signal ratio of 500:1 here.

Dots mode

Not every DSO has it, but Dots display mode is essential whenever the DSO gets near the limits of the sample theorem, hence signal reconstruction – or even acquisition itself – appears flawed.

As an example, consider a 12 MHz square wave with fairly moderate 3 ns rise time, hence a perfectly adequate signal for a 200 MHz oscilloscope like the SDS824X HD.

Fig. 58 SDS824X HD_Square_12MHz_3ns_1GSa_Vect

The screenshot above shows the standard use case: Auto memory with at least 10 Mpts max. record length, Sin(x)/x reconstruction, Vector display mode, no Color grading, Persistence off.

In 2 channel mode, where we get a sample rate of 1 GSa/s without aggressive AA-filter, the waveform looks pretty good. With these settings, it would be pretty hard to provoke major reconstruction errors or even aliasing on a deep memory DSO like the SDS824X HD. Yet there might still be situations where we can't get a sufficient real-time sample rate. To demonstrate this, we can use the Constant Sample Rate setting instead of Auto Memory.

As a first step, let's reduce the sample rate to 250 MSa/s, which many would consider still adequate for a 12 MHz signal:

Fig. 59 SDS824X HD_Square_12MHz_3ns_250MSa_Vect

Even though the fundamental frequency of the signal is just 12 MHz and the Nyquist frequency (125 MHz) is more than ten times higher, we still get to see massive reconstruction artefacts and aliasing already. So much for the sometimes mentioned “rule of thumb” which suggests that a bandwidth five times the repetition frequency of a square wave would be adequate…

Let’s take this one step further and set the sample rate to 100 MSa/s:

Fig. 60 SDS824X HD_Square_12MHz_3ns_100MSa_Vect

With the previous settings, we still got something remotely similar to a square wave. We go one step further and reduce the sample rate to 50 MSa/s:

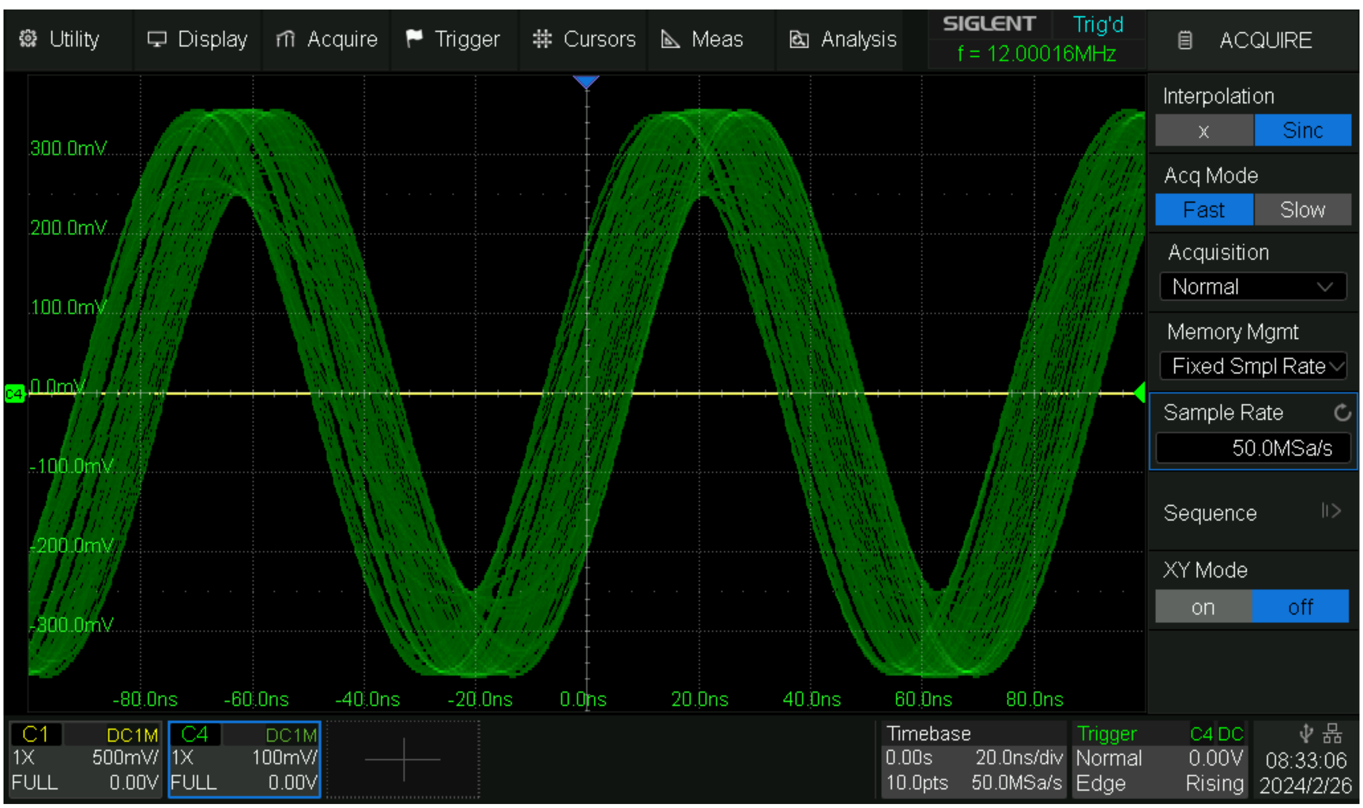

Fig. 61 SDS824X HD_Square_12MHz_3ns_50MSa_Vect

Now we finally got a pure sine wave with lots of amplitude modulation and jitter – certainly not a very good representation of the original waveform anymore. We still want to take it to the extreme and reduce the sample rate even further to 20 MSa/s, thus violating Nyquist even for the fundamental frequency:

Fig. 62 SDS824X HD_Square_12MHz_3ns_20MSa_Vect

This last screenshot needs not be commented, except for the fact, that the SDS824X HD won't let us use the original time base of 20 ns/div with such a low sample rate anymore. Tthe DSO has automatically switched to 50 ns/div, so that we get at least a total of 10 samples per record.

Anyway, this is not the end – after all we've got the Dots display mode up our sleeves:

Fig. 63 SDS824X HD_Square_12MHz_3ns_20MSa_Dots

Yes, with only 1 point per division (10 points for the whole record!), there is no contiguous trace and the rendering is a bit dim. Yet nothing that could not be improved by a little Persistence time:

Fig. 64 SDS824X HD_Square_12MHz_3ns_20MSa_Dots_P1

What we get now is a perfect visual representation of the original signal – within the bounds of the 244 MHz bandwidth, that is – despite the effective sample rate of only 20 MSa/s.

The trigger path is completely separate and always works at the maximum sample rate for the current channel configuration. In my example, there were two active channels and sample rate was 1 GSa/s. This data stream is fed to the trigger engine directly, but might get decimated before it is stored into sample memory, according to max. record length and time base settings.

Dots display mode can only work as long as the trigger remains stable; this is the reason why this cannot replace ETS (Equivalent Time Sampling) or RIS (Random Interleaved Sampling), i.e. we can never display signals that would violate the Nyquist criterion in the trigger path.

Trigger

Trigger Jitter

The datasheet specifies the trigger jitter as <100 ps. This doesn't sound great, considering the SDS2000X HD, where the specification is <10 ps RMS (and it has been measured as 2.02 ps actually).

Now let's measure this using a 200 MHz sine signal from an OCXO-driven AWG (SDG7102A), fed into channels 2 and 4 of the SDS824X HD via a 12.4 GHz resistive power splitter. This way we can observe the jitter in the trigger channel as well as a not triggered channel.

The high quality 200 MHz sine signal has been chosen for its low inherent jitter – after all we want to characterize the DSO and not the signal source.

We need to utilize a measurement gate, because the T@M measurement considers all rising edges in the record, whereas we only want to measure the first one.

Fig. 65 SDS824X_HD_Trigger Jitter

At a time base of 1 ns/div, we cannot see any jitter in the triggered as well as the non-triggered channel after more than 10 minutes at infinite persistence.

The jitter measurements are as follows:
Triggered channel: 28.6 ps pk-pk, 4.852 ps rms;
Un-triggered channel: 28.9 ps pk-pk, 4.652 ps rms;
Skew Ch.2-Ch.4: 20.9 ps pk-pk, 3.39 ps rms;

While this is about twice as much as the SDS2000X HD, it is still very respectable and miles ahead of older designs with analog trigger system (none of Siglent's X and A series).

AC Trigger Coupling

Most of us use DC coupling for the trigger almost all the time, and there is not much to talk about it, other than that it works just as it should. We rather want to examine AC trigger coupling now.

Why and when would we need AC coupling for the trigger at all? Usually, we make that choice for the channel input and if we select AC coupling there, the trigger will inevitably be AC coupled as well. So, there we already have the answer – we have the opportunity to force the trigger into AC coupling, even when the corresponding input channel is DC coupled. This can be useful for AC signals that have a DC offset that we want to watch on the screen. The offset might change with time and we still don't want to lose triggering

AC trigger coupling does not display a trigger level indicator, simply because it would need to closely follow even a fast-changing signal offset, thus might be rather distracting instead of beneficial.

The following test uses a 200 mV_{PP} 100 ns wide pulse at 1 MHz repetition frequency that is superimposed on a 600mV_{PP} sine wave at 100mHz, which acts as a variable DC offset here. As if this weren't enough, this signal has a fixed DC offset of -6V on top of that, which needs to be removed by means of the vertical position control and the trigger level adjusted accordingly. Infinite persistence is used to give a hint what is going on.

With DC trigger coupling, triggering would only occur about 1/3 of the total time in this scenario and even then, the horizontal position would only be reasonably stable because of the short 1 ns rise time of the pulse edges. A signal with slower edges would move horizontally as well, because of the permanently changing trigger level (relative to the AC portion of the input signal).

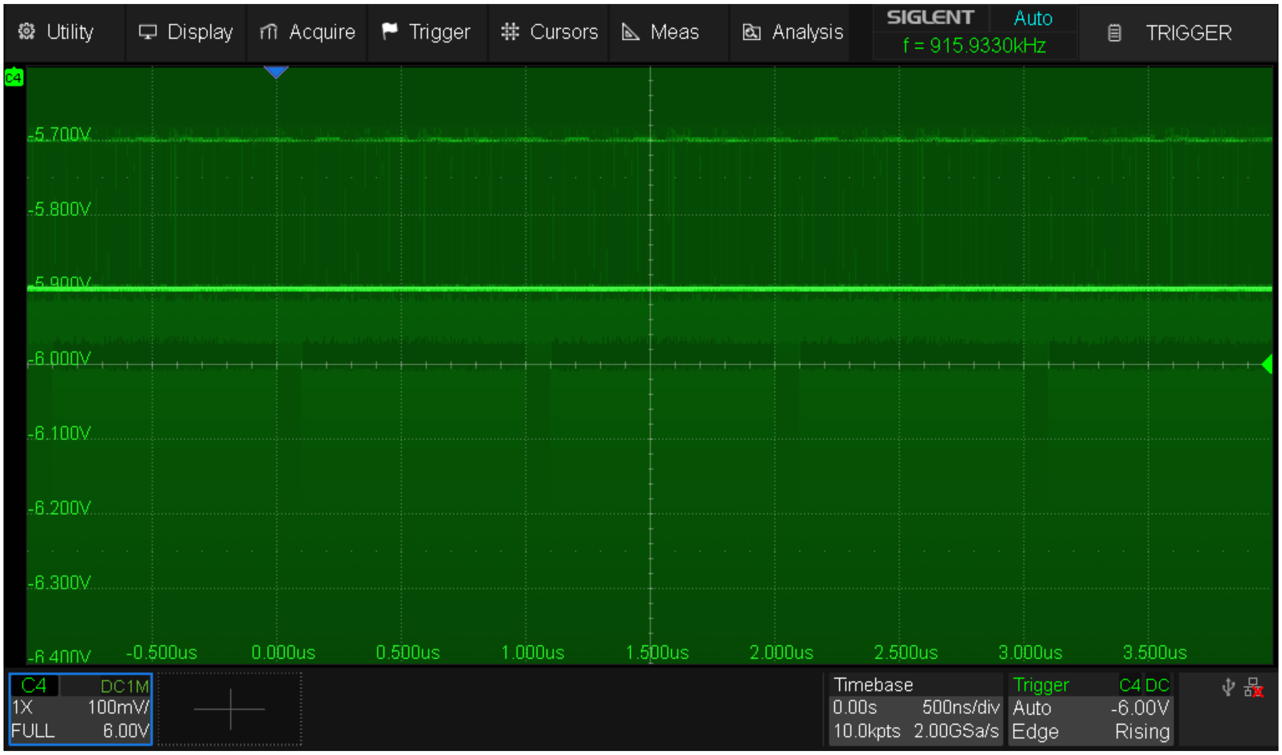

Fig. 66 SDS824X_HD_Trigger_DC_VarOffset

It's totally different if we use AC trigger coupling. When using Auto-Set by pushing the trigger level control, the trigger level is set to 50% of the signal amplitude. With this, triggering occurs always at the same point on the X-axis, no matter what the DC offset or low frequency instantaneous signal level is. The waveform constantly changes its vertical position on the screen, but remains stable on the time axis – and even more important, the signal is triggered continuously.

Fig. 67 SDS824X_HD_Trigger_AC_VarOffset

We’ve heard complaints about DSOs that prove unable to maintain an AC- or LFRJ-trigger without some additional jitter. Here’s a test with AC trigger coupling and a 6 ns wide pulse with 1 MHz repetition frequency. The screenshot has been taken after several minutes with infinite persistence.

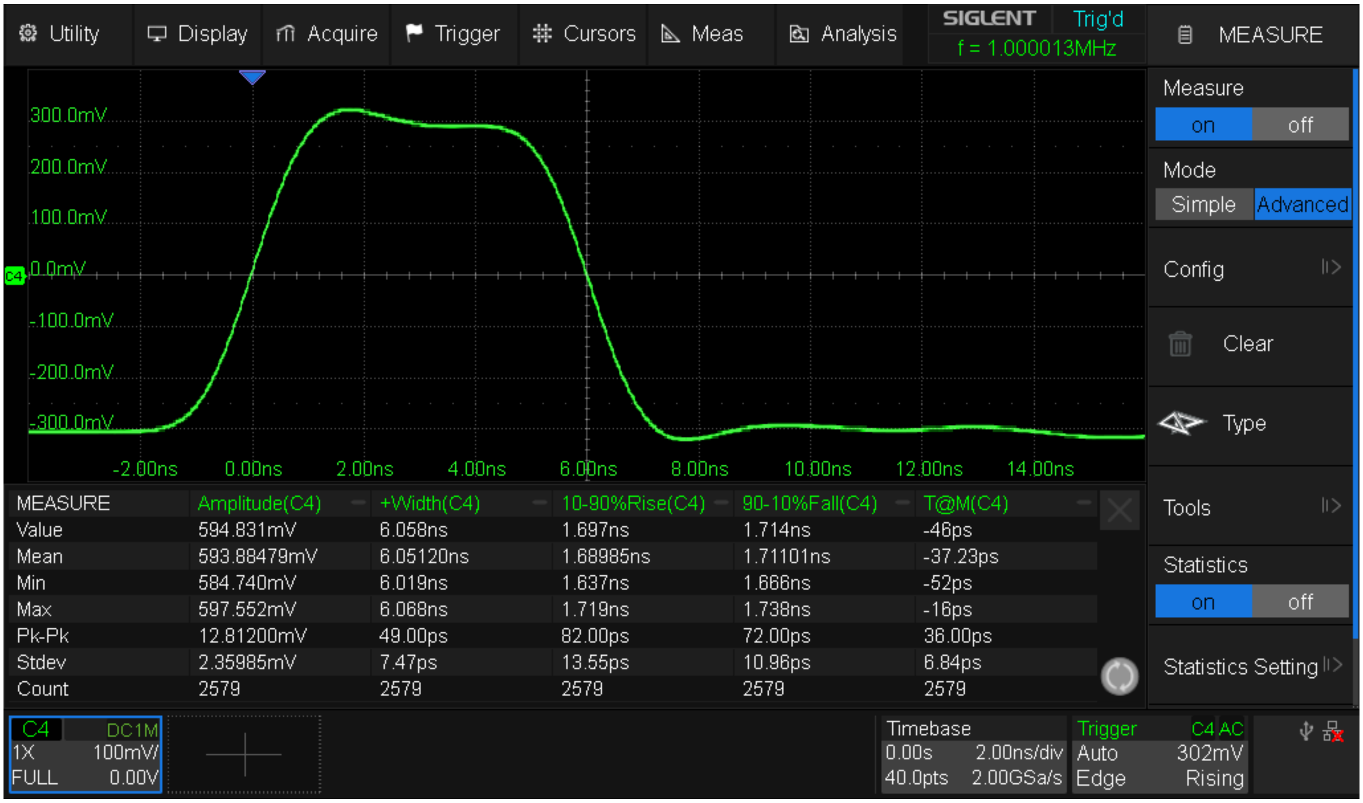

Fig. 68 SDS824X_HD_Trigger_AC_Jitter

Jitter can be measured as 36.0 ps peak to peak and 6.84 ps RMS; not quite as good as the former test with the 200 MHz sine signal, yet certainly not bad either – and also reveals that the pulse generator (SDS7102A) produces quite stable signals, even when fast edges are involved.

Triggering noisy signals

Of course, for serious measurements, our goal should be to find a reliable, stable and noise-free trigger source with sufficient amplitude. Sometimes this is not available and we need to try the next best thing by getting a stable triggering also from less ideal sources.

Let’s assume a low frequency signal with high frequency spikes (maybe from fast logic circuits nearby) superimposed. To simulate this, a 600 mV_{PP} 1 kHz sine wave has a 10 mV_{PP} 3.254 MHz pulse train (10 ns wide pulses) riding on it. Standard DC trigger doesn’t work with such a signal, but HF-reject coupling does:

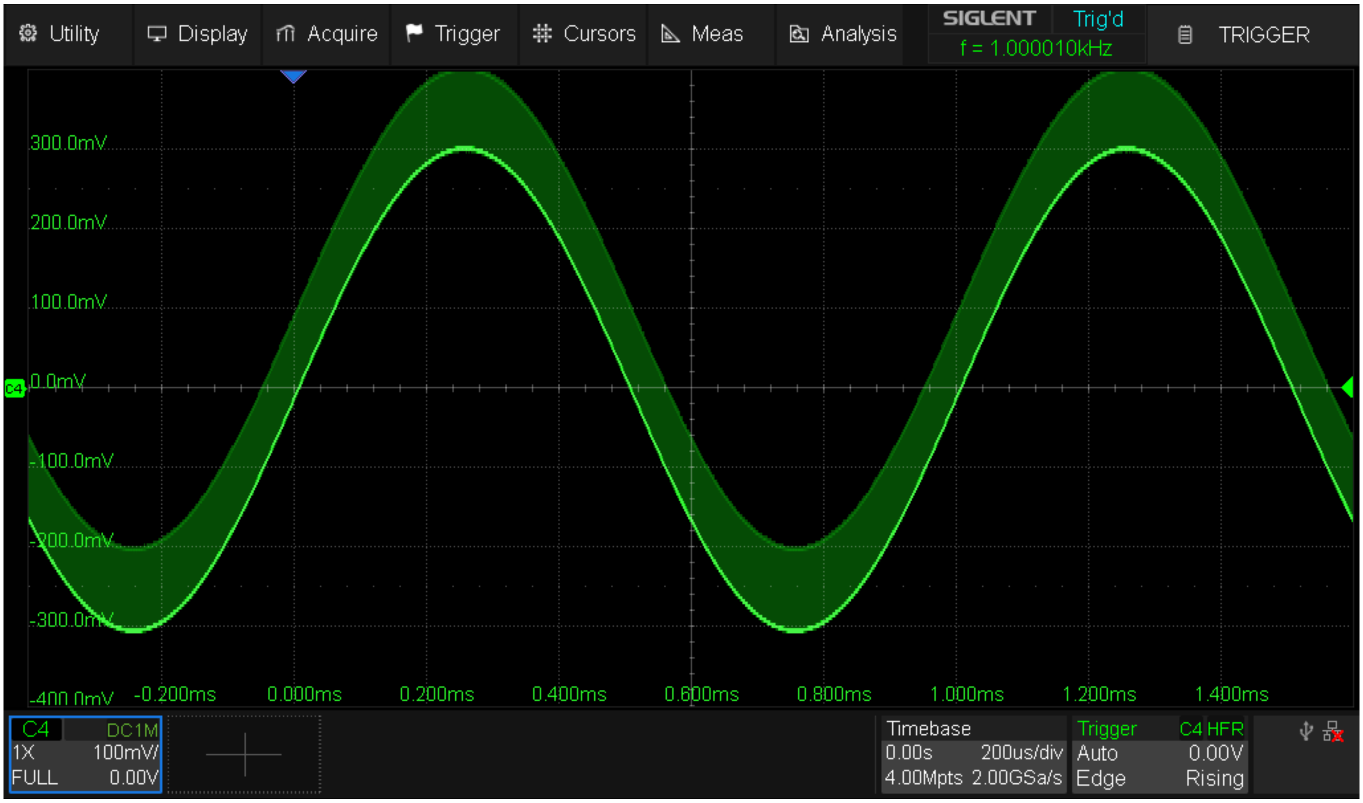

Fig. 69 SDS824X_HD_Trig_Spikes_HFRJ

Just for fun, we can do the opposite thing and use LF-reject trigger coupling. This triggers stably on the pulses, yet because of the high waveform update rate the screen is full of traces from the 1 kHz sine.

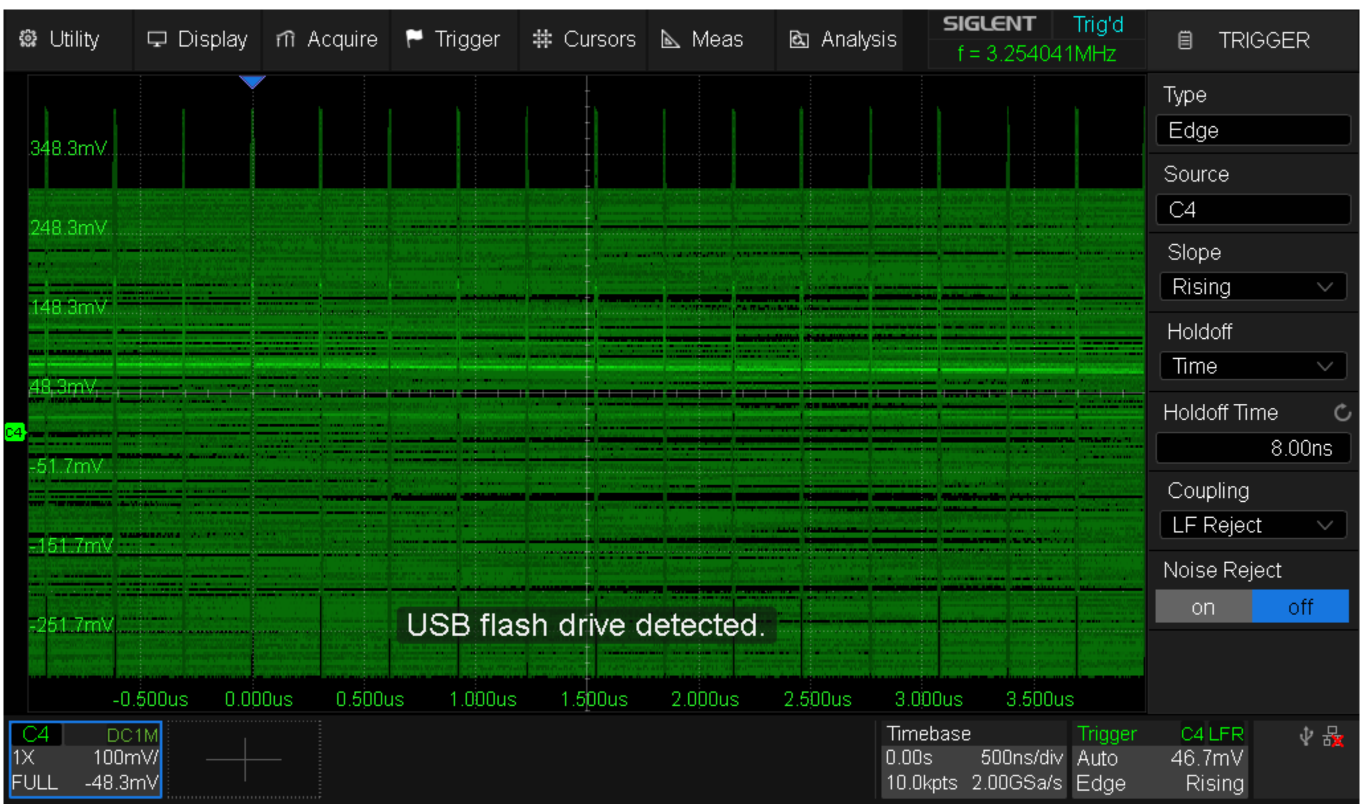

Fig. 70 SDS824X_HD_Trig_Spikes_LFRJ

After stopping the acquisition, we can closely inspect the last record. Even better, we can look up all the previous acquisitions in the history:

Fig. 71 SDS824X_HD_Trig_Spikes_LFRJ_Hist1

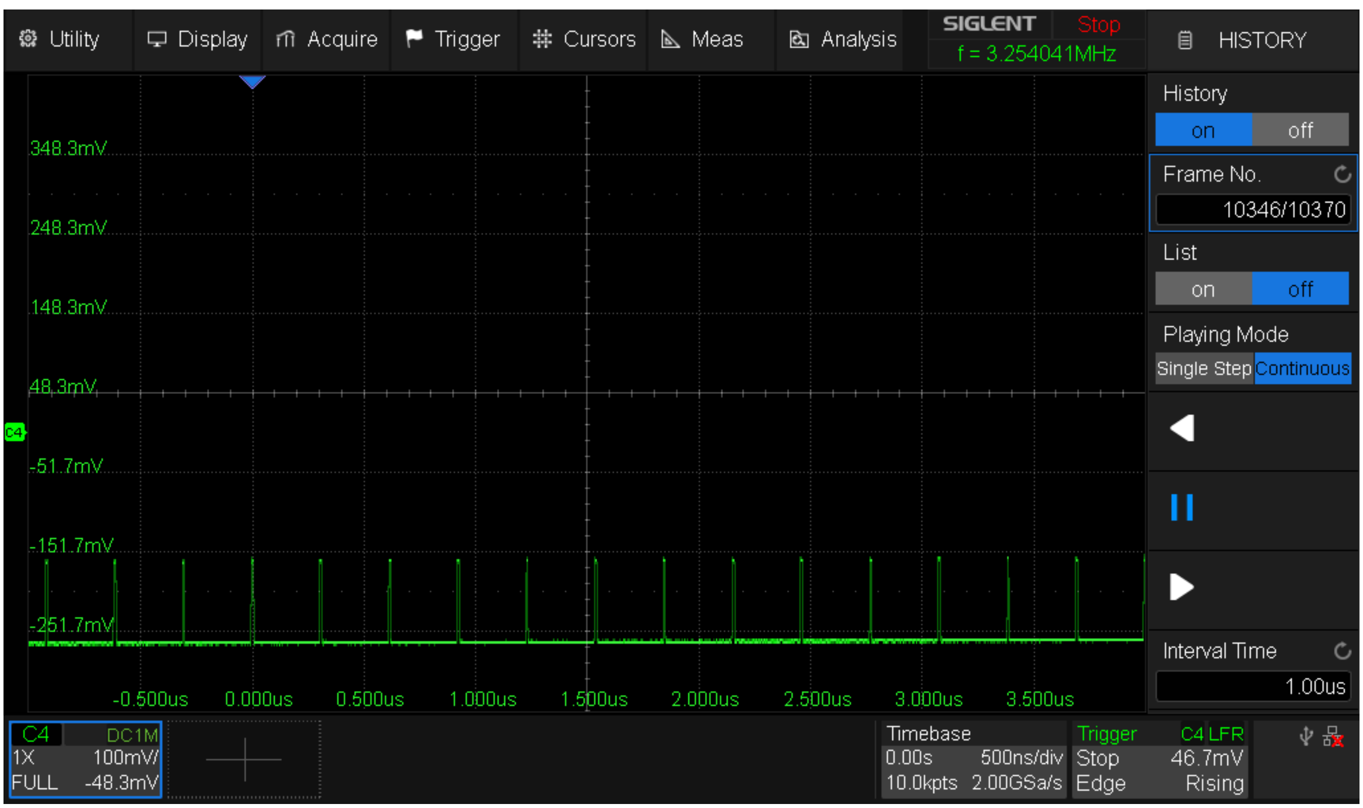

Fig. 72 SDS824X_HD_Trig_Spikes_LFRJ_Hist2

Another common situation is just a noisy signal like this:

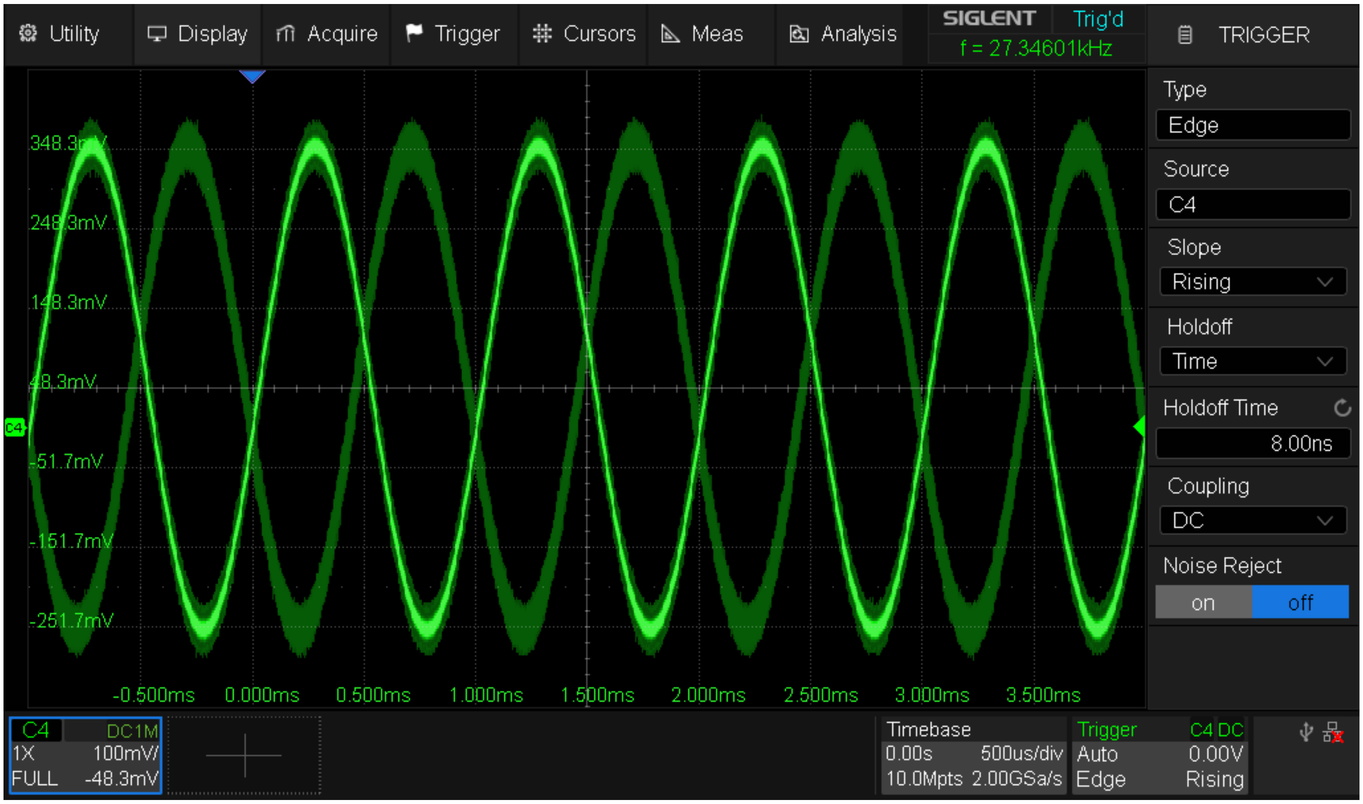

Fig. 73 SDS824X_HD_Trig_Noise_DC

Infinite persistence has been used to visualize the unstable signal phase.

We can still trigger on such a signal by simply lowering the trigger bandwidth, i.e. using HFRJ:

Fig. 74 SDS824X_HD_Trig_Noise_HFRJ

Alternatively, the Noise Reject switch in the trigger settings will increase the trigger hysteresis, thus making it immune to noise (within reason).

Fig. 75 SDS824X_HD_Trig_Noise_DC_NRJ

Measure

Measurements

There are the simple measurements, where we can define an arbitrary set of up to all 52 measurements that are related to a single channel. This set can subsequently be applied to any Input-, Zoom-, Math- or Reference-Channel. That's also one disadvantage of the simple measurements – the set is restricted to a single channel at a time. The other drawback is the total lack of statistics in this mode.

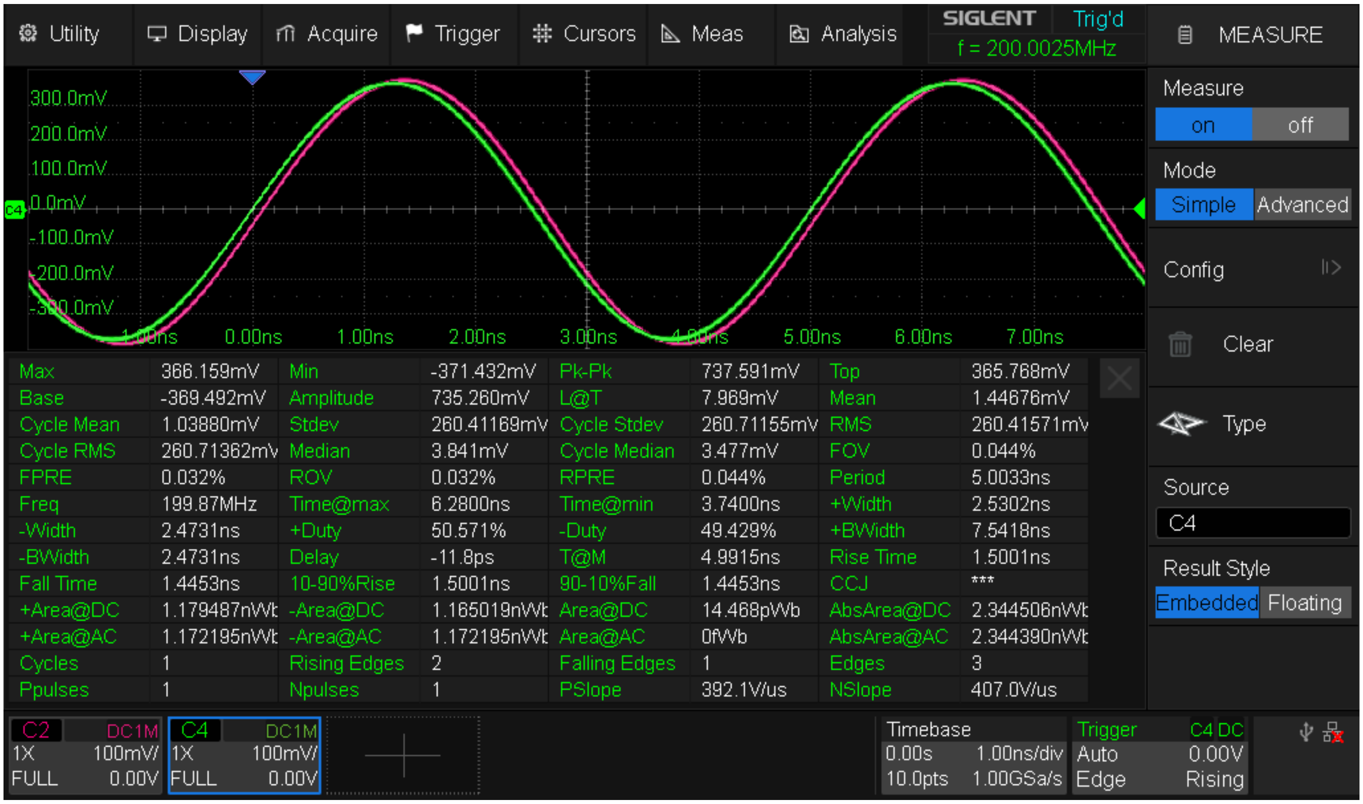

Fig. 76 SDS824X_HD_Measure_Simple_Embedded

A high number of measurements might be desirable at times, yet it takes up a lot of screen space. Siglent has now introduced a new display option for measurements; "Floating":

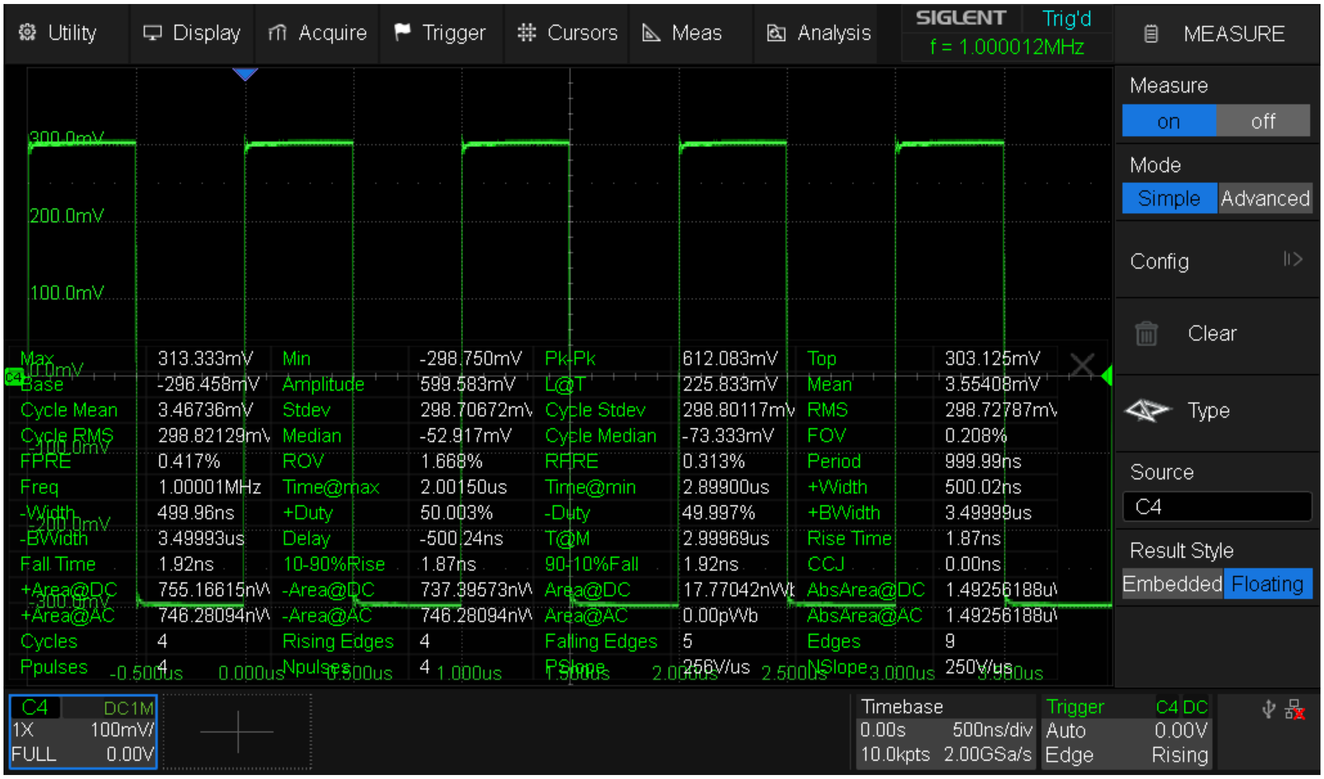

Fig. 77 SDS824X_HD_Measure_Simple_Floating

This is a transparent overlay which might help to better utilize the screen space in certain situations.

We also got the Advanced Measurements., but unfortunately only mode A; mode B is available only in its bigger (and more expensive) siblings:

Fig. 78 SDS824X_HD_Measure_Advanced_Embedded

We can have statistics and also enable the little Histicons (History Icons) as in the screenshot above, we can also mix various channels, yet we are restricted to only 5 measurements at a time.

We can have floating measurements in this mode as well, yet it might be a good idea to turn the axis labels off to avoid text collisions:

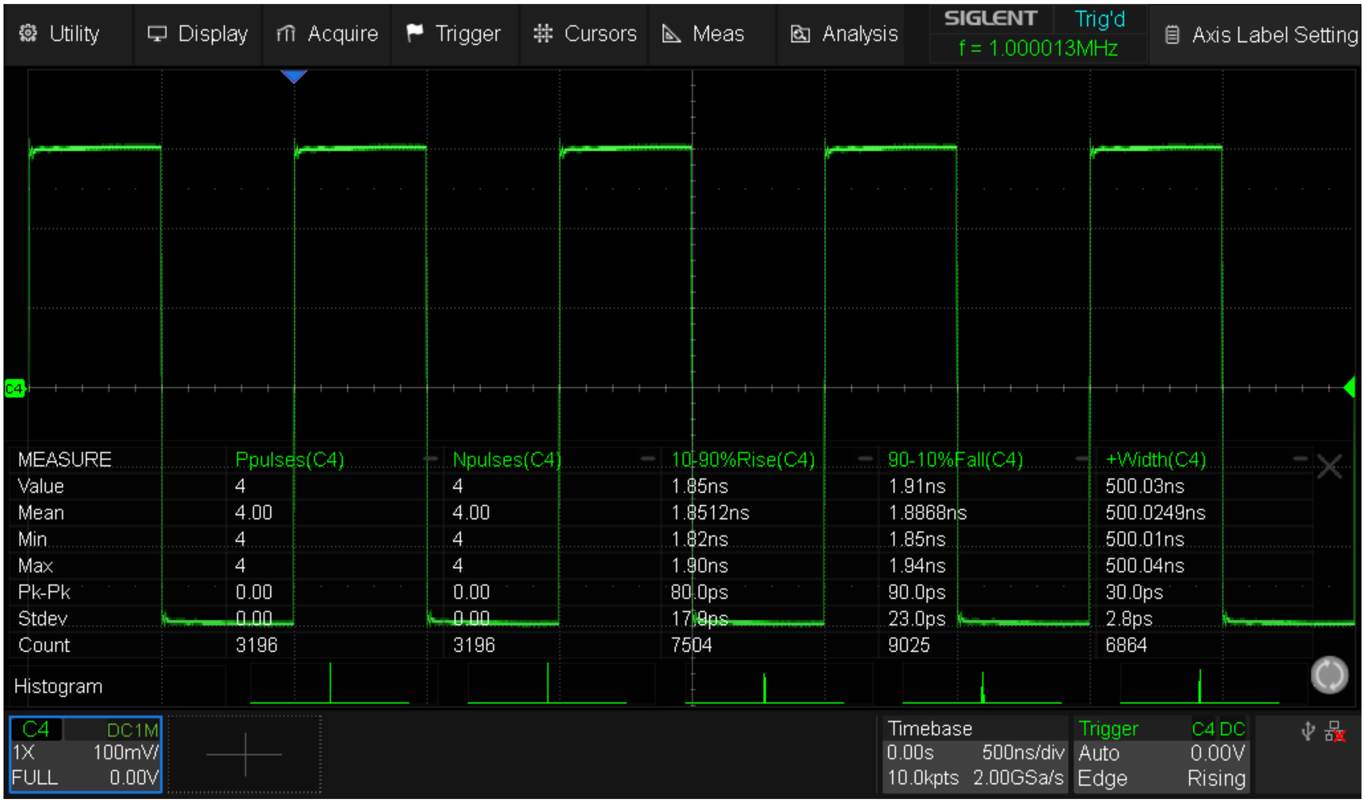

Fig. 79 SDS824X_HD_Measure_Advanced_Floating

Measurement Histograms

The small Histicons in the advanced measurements statistics can be enlarged by simply clicking or tapping on them. This opens a separate window with a more detailed version of the histogram. The last measurement result is marked by a small red dot in the histogram, so one can watch the buildup of the histogram even when there is already a lot of data collected and histogram bars don't visibly change anymore.

Fig. 80 SDS824X_HD_Measure_Advanced_Histogram

Deep Measurements

Here's some demonstration of a common exercise which cannot be solved without deep measurements.

Imagine a 16-bit PWM based on 20 MHz clock frequency. This results in a rather slow 304 Hz PWM signal that can resolve 65536 different levels of duty cycle. To analyze this, we should be able to have accurate time measurements with at least 0.001525878 % (15 ppm) resolution

A very similar demonstration has been published for the 8-bit SDS2354X Plus a while ago already (reply #4336):

https://www.eevblog.com/forum/testgear/siglent-sds2000x-plus-coming/msg5183499/#msg5183499

Let's see how the SDS800X HD fares.

Time base is set to 500 µs/div, so that we can capture at least one full PWM period.

At 2 GSa/s, this results in 10 Mpts record length.

The PWM signal for this test has 1 ns rise time and 0.001% resolution for the duty cycle.

First the lowest at 0.001 %:

Fig. 81 SDS824X_HD_Duty_0.001

Near full scale at 99.999 %:

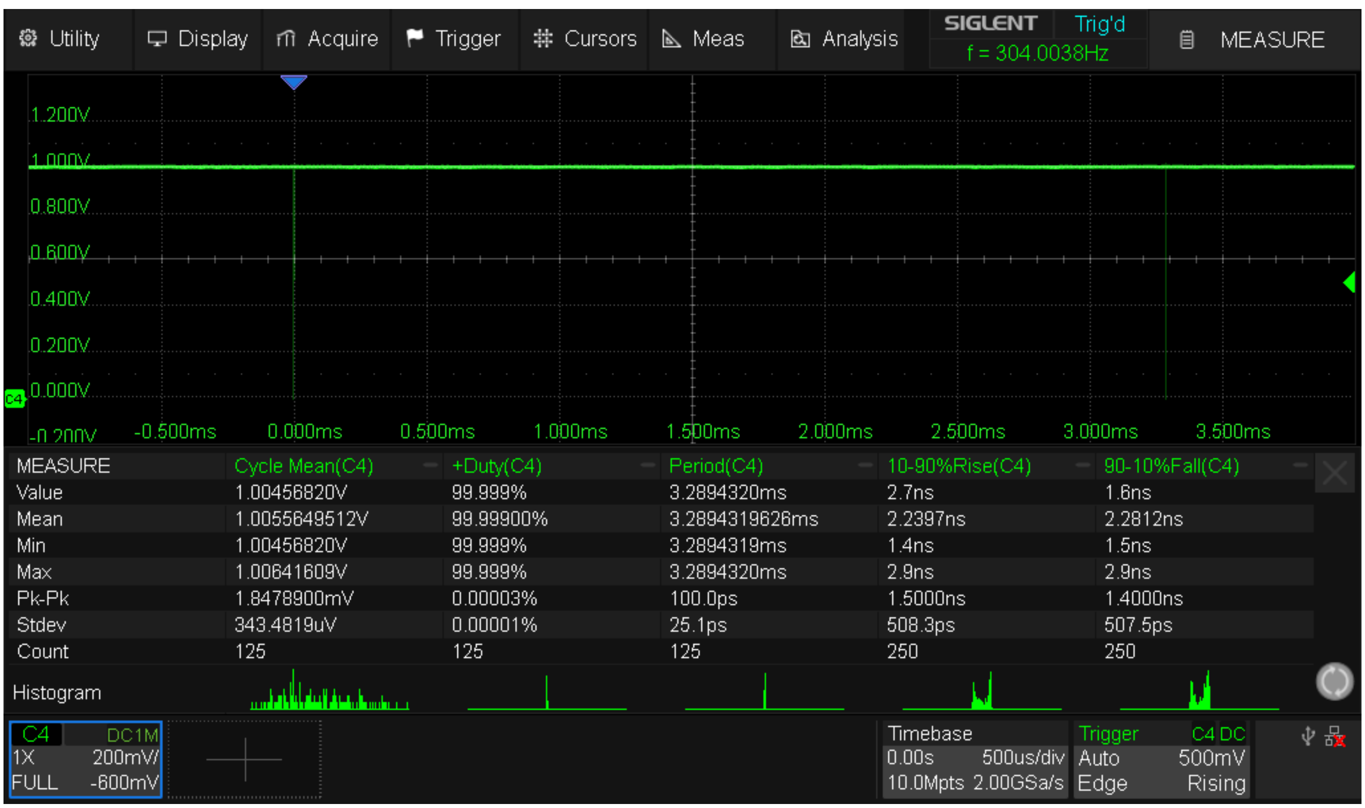

Fig. 82 SDS824X_HD_Duty_99.999

Half scale at 50.000 %:

Fig. 83 SDS824X_HD_Duty_50.000

Finally, one step higher at 50.001 %:

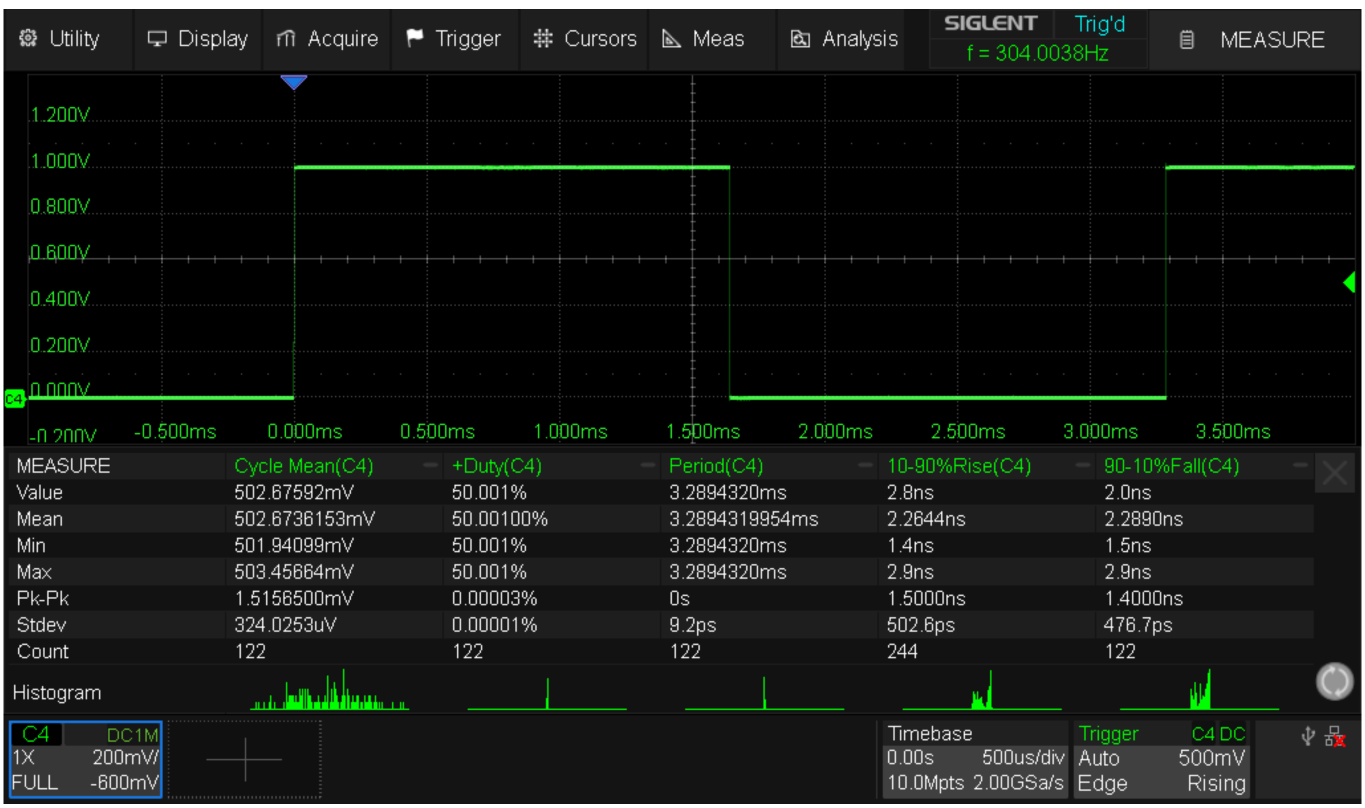

Fig. 84 SDS824X_HD_Duty_50.001

The duty cycle measurement is spot on and stable, even though not quite as impressive as the SDS2000X Plus (look at the peak-peak and standard deviation).

The period measurement is fairly stable too, with a standard deviation of only 9.2 ps. Peak deviation can only be measured in 100 ps steps, which is ten times more than the SDS2000X Plus. Quite obviously the peak deviation was well below 100 ps, hence gets reported as 0 s.

The Cycle Mean measurements gives an approximation of the resulting voltage level. It is far less precise than the duty cycle measurement though. No wonder – even a 12-bit DSO is still no precision bench DMM, hence measurement resolutions of ~15 µV are not going to be stable – this also shows in the standard deviation of ~324 µV – which is about ten times as much as with the SDS2000X Plus, but that's only because the amplitude was ten times lower for the test back then (and the required resolution there would have been ~1.5 µV).

Finally, the rise and fall times are about as (in)accurate as can be realistically expected at 2 GSa/s. The rather high peak deviation of ~1.4 ns already hints on the averaging of many individual measurements to get the final result. Once again, we see an advantage of the SDS2000X Plus, even though the sample rate is the same. Yet this is not only about the physical sample rate, but also interpolation strategies, which require massive HW support and might be a bit simpler in the SDS800X HD. The higher bandwidth of the SDS2000X plus is helpful in this case as well.

Of course we can always get full accuracy for one local detail like the rise time by using zoom trace measurements:

Fig. 85 SDS824X_HD_Duty_50.001_Rise

Now the rise time measurement result is much closer to the truth. The key for this is to use a time-base faster than 50 ns/div in the zoom window. Now the Sinc reconstruction generates additional data points, thus increasing time resolution and reducing the standard deviation of the measurement to just ~6 ps, which is in turn even better than the SDS2000X Plus.

Trend Plots

Let's start with the more familiar (to most) Trend charts. As the name suggests, they plot a measurement value over time. For this, the record length of the raw acquisition can be short – a single full signal period would already be enough. Shorter records have the advantage of faster processing and less memory consumption.

Fig. 86 SDS824X_HD_Measure_Trend_AM_Interval

Of course, even a 12-bit DSO is not a metrology-grade instrument, hence we cannot use Trend Plot to observe the stability of a voltage standard; yet there are still plenty of applications where 0.5% accuracy is sufficient.

For the example above, we have used a 1 s time interval Trend Plot, measuring the peak-to-peak amplitude of a 10 MHz sine wave, 100% amplitude modulated by a 10 mHz ramp signal. The minimum time interval for Trend Plot would be 0.5 s.

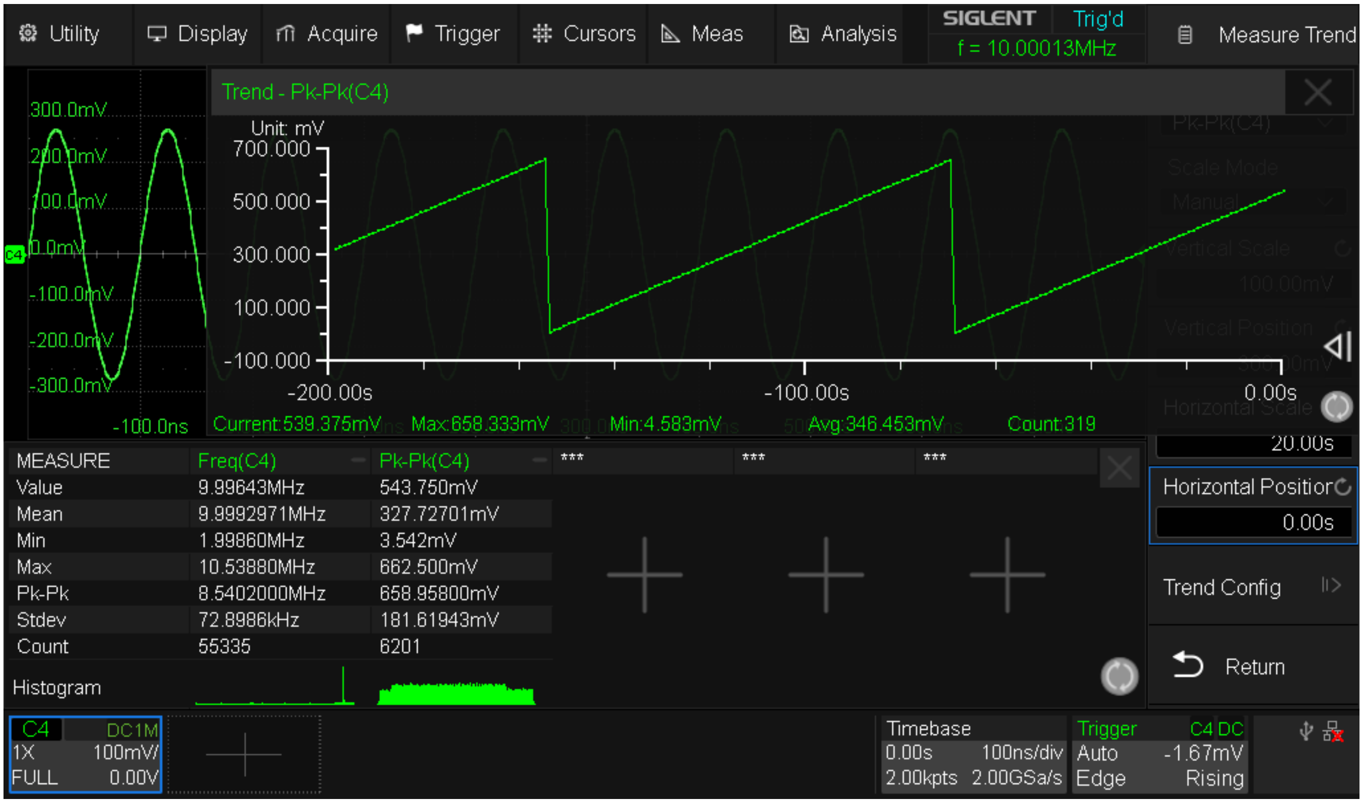

Fig. 87 SDS824X_HD_Measure_Trend_AM_enlarged

It can be seen that the Trend Plot has a separate statistic, as it is significantly slower than the regular measurements, hence works on a decimated subset of the original measurement data. Consequently, since the time interval of the Trend Plot is one second, the measurement rate of the Pk-Pk measurement can be calculated as ~19.4 per second, while at the same time we get 174 frequency measurements per second.

Instead of a fixed time interval, we can alternatively use sequence record mode. Now the trend plot window behaves like a scope in roll mode, i.e. the update rate is faster, but the time axis shows measurement samples instead of time units now.

Fig. 88 SDS824X_HD_Measure_Trend_AM_Sequence

Track Plots

Track plots also show the developing of a measurement value, but not over time but within a single record. As opposed to trend plots, this works best with long record lengths and only with certain measurements – the ones that are computed for the entire record, i.e. all the time related measurements.

Consider a 10 MHz carrier frequency modulated with a 20 kHz sine wave and a frequency deviation of +/-1 MHz. Other than e.g. AM, we cannot really see this in the regular y-t display. This is where the Track Plots come in handy; they let us “demodulate” frequency and phase modulated signals – and such modulations could also come from noise, drift and jitter.

Fig. 89 SDS824X_HD_Measure_Track_FM

Take a closer look at the above screenshot: the record length is 5 Mpts and there are 1000 times more frequency measurement samples than Pk-Pk amplitude measurement samples. Experienced people could tell from the histogram that the modulation signal would very likely be a sine wave, but they would not be able to determine the deviation and modulation frequency.

From the Track Plot we can see that the modulation signal is a sine wave with exactly 50 µs period (=20 kHz) and it alters the carrier frequency between 9 and 11 MHz.

Counting Pulses

This is a demonstration of the pulse count function. As further refinement, gated measurements can be used in order to ignore unwanted portions of the record.

First the basic pulse measurement without any bells and whistles; a 100 MHz pulse packet with 1 ns rise time and 1000000 pulses is fed into Ch.4

Fig. 90 SDS824X_HD_Pulsecount

The scope registers the correct number of one million positive pulses. The negative pulse count naturally delivers the same number minus one. Together with the peak-to-peak deviation of zero over >100 acquisitions it is obvious that the pulse count is spot on.

Let’s add a measurement gate. We define it to start 1.0 ms after the trigger point and to be 5 ms wide:

Fig. 91 SDS824X_HD_Pulsecnt_Gate

We now get a count of ~500k (100 MHz x 0.005 sec.) as expected. The count is a little higher because of the limited accuracy of the time base in the SDS800.

We can engage the zoom view for a closer inspection of the waveform:

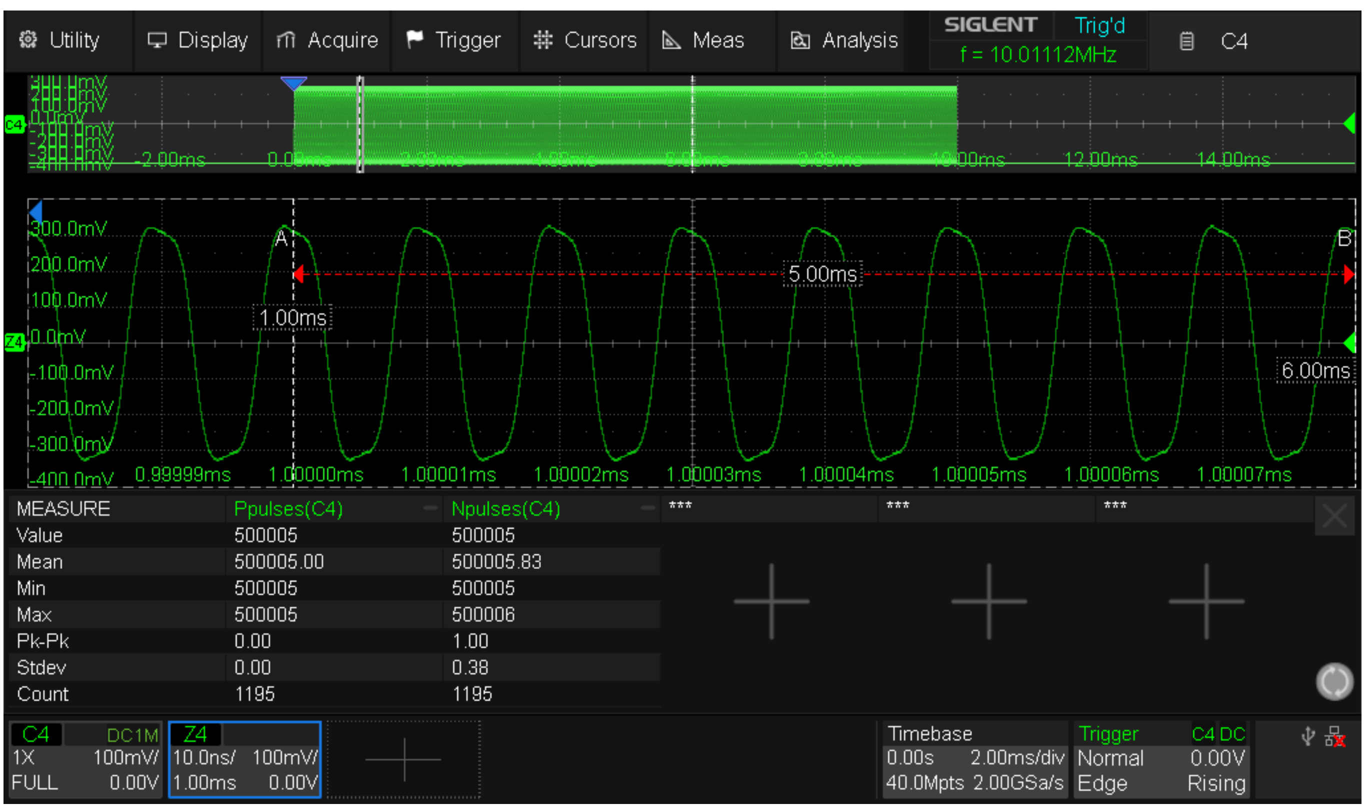

Fig. 92 SDS824X_HD_Pulsecnt_Gate_Zoom

We can still see the gating cursors in the main window, but the detailed view with time data is in the zoom window now. Since the “B” cursor exceeds the size of the zoom window, it is drawn at the right border and the time specification of 6.00 ms is another hint that it is outside the zoom window, which is only 100 ns wide.

The gating cursors are fully independent, consequently we can still add regular cursors (manual/tracking/measurement):

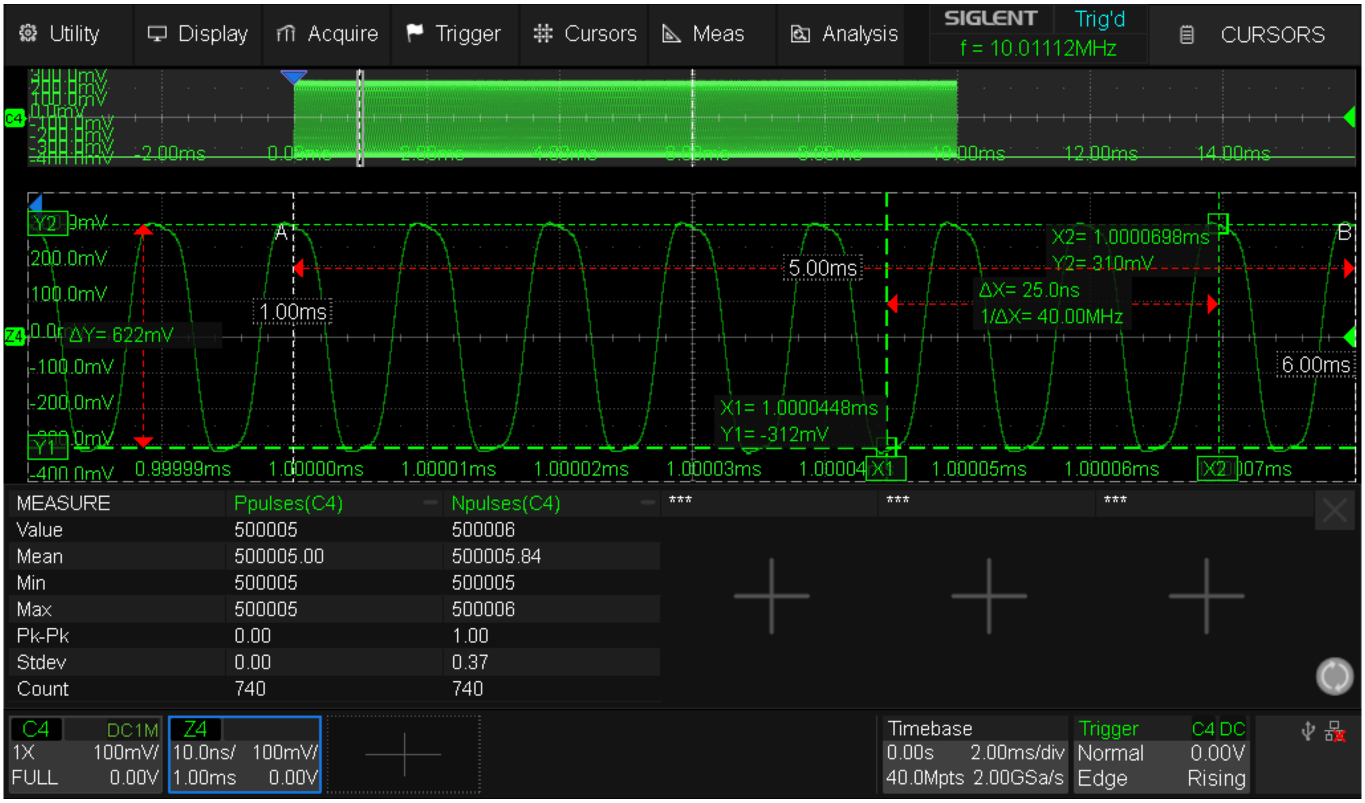

Fig. 93 SDS824X_HD_Pulsecnt_Gate_Zoom_Cursors

With all these information, screen gets a bit busy, but we could also use the traditional info block for the regular cursors and place it at the least disturbing spot:

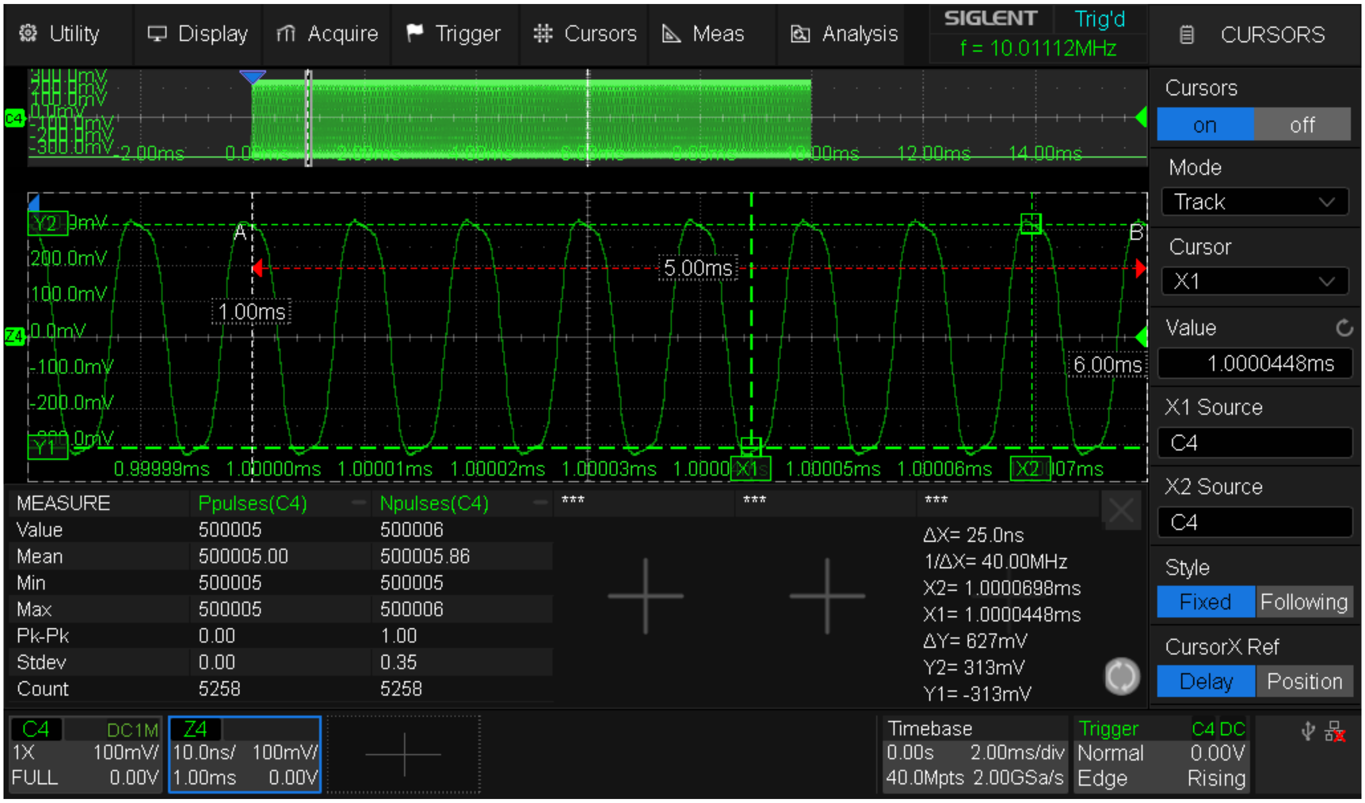

Fig. 94 SDS824X_HD_Pulsecnt_Gate_Zoom_Cursors2

Cursors

Not everyone might be familiar with the various concepts of cursor measurements, so here comes a brief explanation together with some examples.

There are three kinds of cursors: Manual, Tracking and Measure.

Manual Cursors

In *Manual* mode, the x and y cursor pairs can be moved freely and used like calipers to measure distances in both axes, even at the same time if so desired.

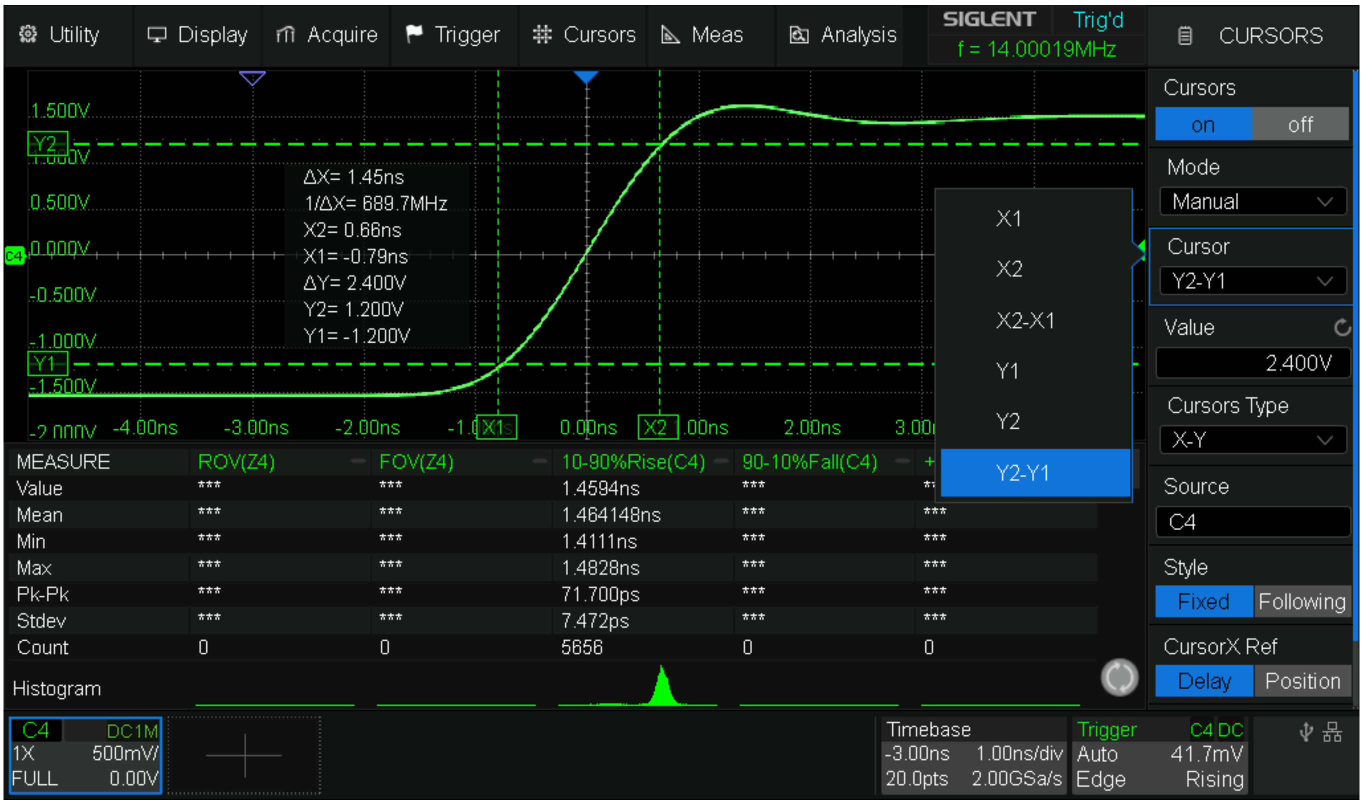

Fig. 95 SDS824X HD_Cursors_Manual

Tracking Cursors

In *Track* mode, the x-cursor pair can be moved freely, whereas the y-cursor pair will always track the selected signal trace. Of course, this makes most kind of cursor measurements much easier and also more precise.

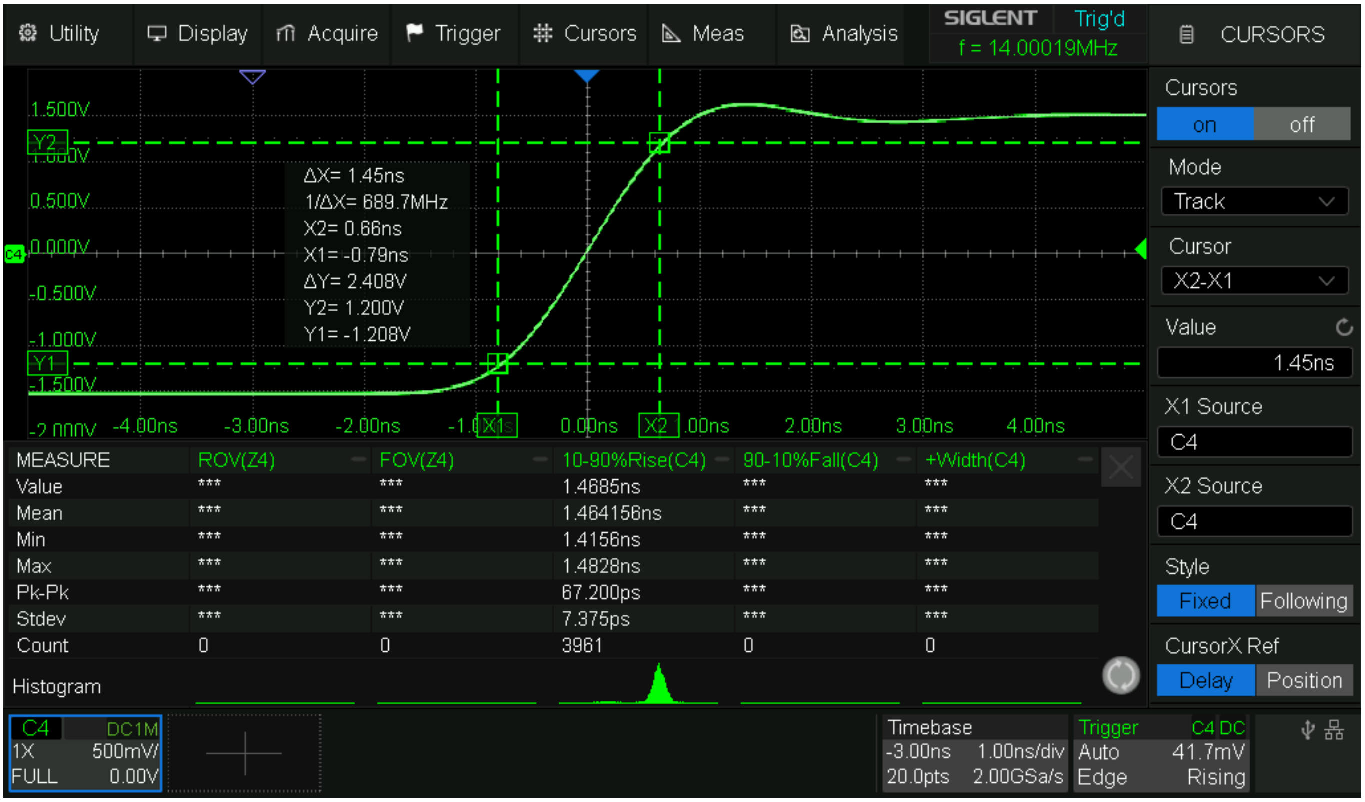

Fig. 96 SDS824X HD_Cursors_Track

Measure Cursors

Finally, in *Measure* mode the cursors are fully automatic and just visualize the points of the signal trace that are used by a certain automatic measurement. Here are some examples:

Rise Time

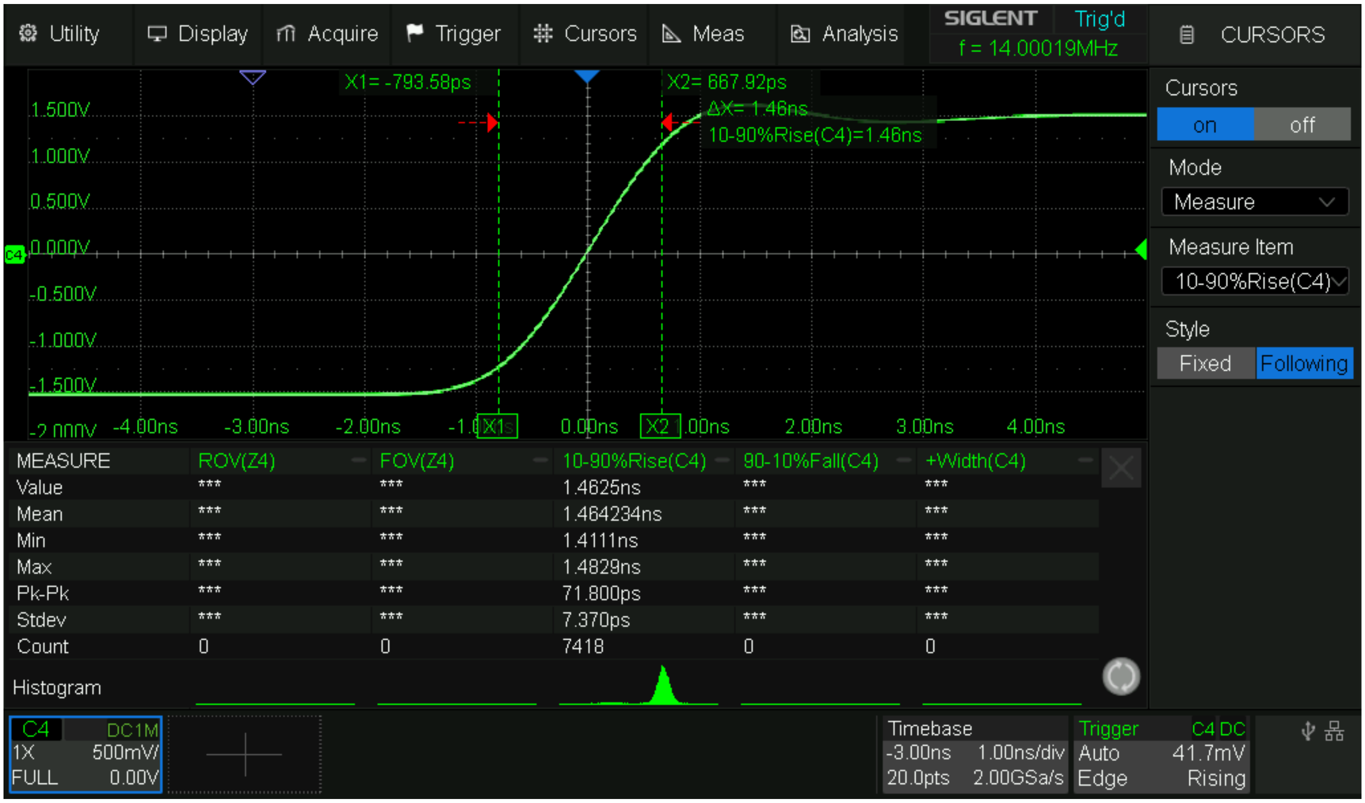

Fig. 97 SDS824X HD_Cursors_Measure_Rise

For the 10-90% rise time measurement, the measure cursors show the corresponding positions for the 10% and 90% threshold on the time axis.

Rising Edge Overshoot

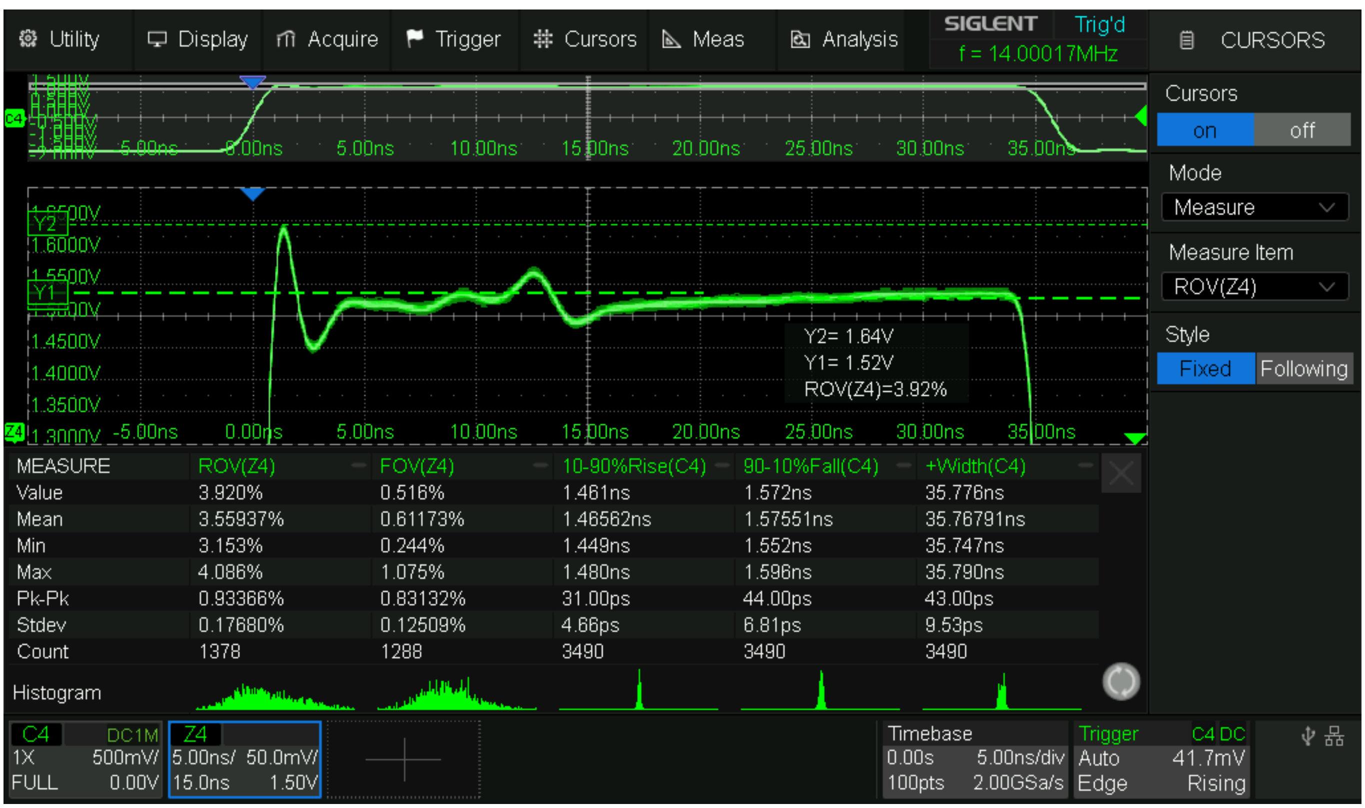

Fig. 98 SDS824X HD_Cursors_Measure_ROV

For the rising edge overshoot, the measure cursors show the amplitude difference between top and positive peak.

Positive Pulse Width

Fig. 99 SDS824X HD_Cursors_Measure_Width

For the pulse width (according to the thresholds configured in the Measurement Config, default is 50%), the measure cursors show the corresponding positions on the time axis.

Math

Identity

Since the SDS800X HD is an extremely affordable package, some corners had to be cut, one of them being the signal samples rendered as clusters of two vertical pixels.

While this hardly poses constraints to our everyday tasks, there might still be situations where we want to get the maximum visual resolution for certain measurements. This is where the math channels come into play.

The Identity function returns the original acquisition data, whereas the Average function is the preferred choice for repetitive signals, because it reduces (also) the 1/f-noise and increases the resolution, hence enables us to produce clean traces even from very noisy signals.

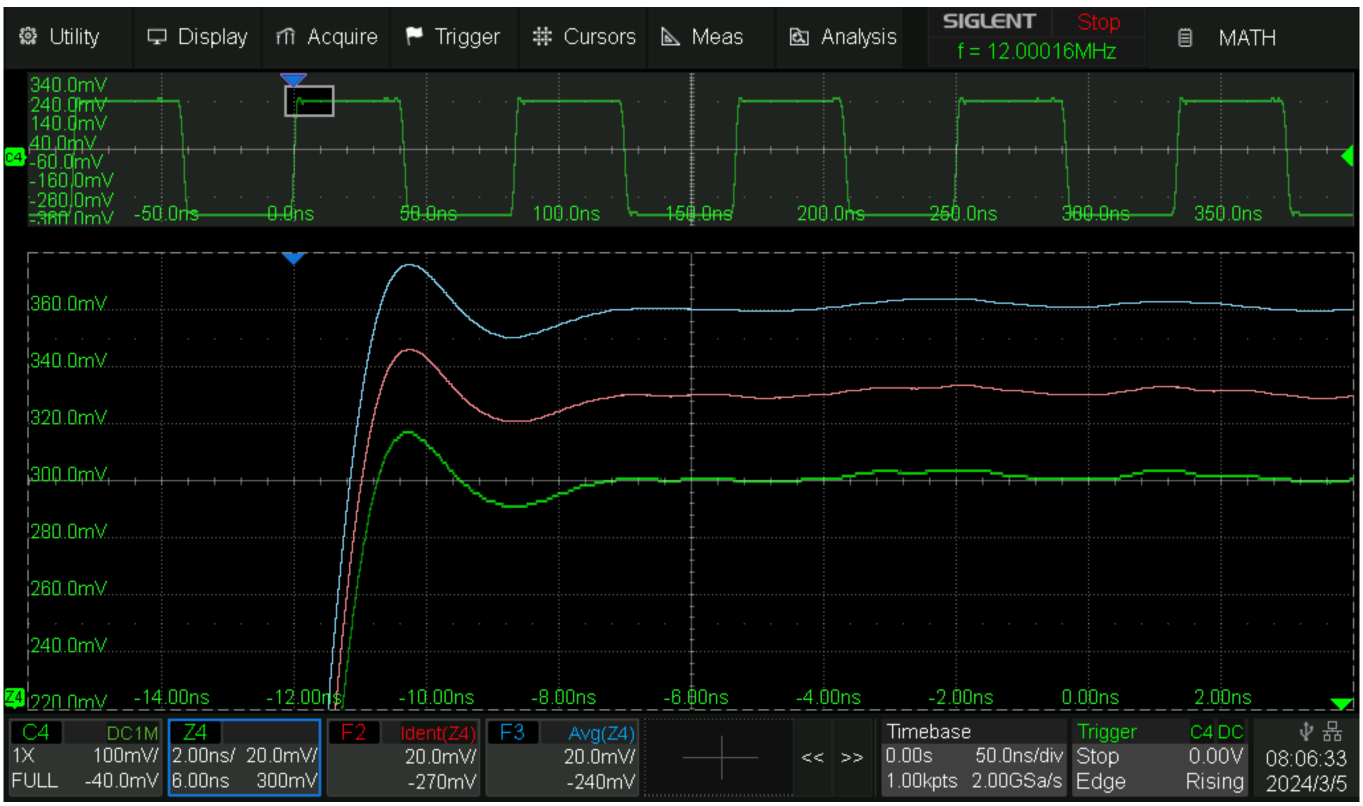

Fig. 100 SDS824X HD_Math_Zoom_Identity_Avg16

The screenshot shows a 12 MHz square wave with 1 ns rise time. Zoom mode has been engaged to take a closer look at the rising edge overshot details.

The channel 4 trace is always 2 pixels high, so it appears thicker than the math traces.

Math trace F2 plots the Identity function, which is basically the same as channel 4, but uses the full screen resolution, hence looks nicer.

Math trace F3 plots the a 16x Average of the signal in channel 4. It reduces the noise and increases the vertical resolution to 16 bits.

If we want to show the math trace(s) exclusively, e.g. for documentation purposes, we can hide the original signal trace in the corresponding channel menu.

ERES

At the end of the day, ERES is just a LP-Filter, and maybe just a simple boxcar filter, see screenshot below with ERES 2.0:

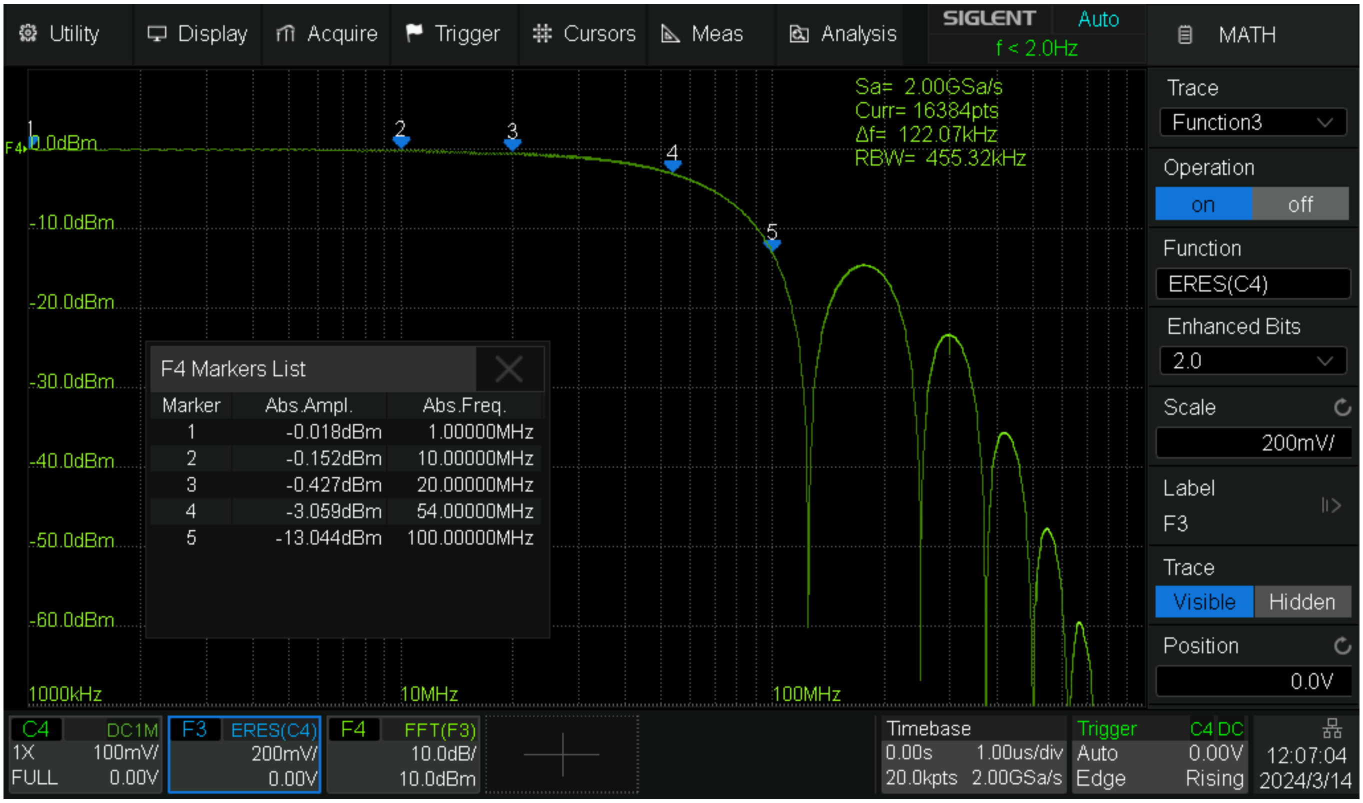

Fig. 101 SDS824X HD_FR_ERES2.0_2GSa

FFT-Setup

Like all Siglent DOSs, starting with the venerable SDS1202X-E, the SDS800X HD has a very useful FFT implementation. To get the most out of it, we should be able to set it up correctly.

The FFT-length in the SDS800X HD can be up to 2 Mpts. This enables low resolution bandwidths and low noise, but it's still “only” a 12-bit DSO, hence results below -73 dBFS shouldn’t be trusted blindingly. In many cases, the usable dynamic range with decent accuracy can be up to 100 dB though.

Here come some hints for proper setup of the FFT on Siglent DSOs in general and the SDS800X HD in particular.

FFT-Bandwidth and RBW

This is quite different to a traditional SA (spectrum analyzer). There is no menu for the resolution bandwidth and also no direct setting for the FFT-bandwidth, even though we have a menu item for the horizontal display parameters, i.e. start/stop frequencies for wideband measurements and center frequency/span for narrowband applications. But this is just for zooming into a longer FFT result; for best speed and lowest RBW (resolution bandwidth) we need to make sure that no high zoom factor is required to get the desired display. The following rules apply:

- The analysis bandwidth (FFT-BW) is always half the FFT sample rate (FFT-SR).
- The frequency step Δf is the sample rate divided by the number of FFT points.
- In the Acquire menu, a constant sample rate can be set in order to also limit the FFT-sample rate.
- The resolution bandwidth (RBW) is the frequency step multiplied by a factor specific for the window function in use.

The maximum number of FFT points can be up to 2 Mpts, but it also depends on the record length, which increases with slower time base settings, which in turn might be limited by the maximum memory as defined in the Acquire menu. Apart from that, the number of FFT points can be further limited by the corresponding setting in the FFT-Config menu.

RBW = Δf * k, where k is the 3 dB bandwidth factor in bins, depending on the window function:

k: Rectangle 0.99, Blackman 1.74, Hanning 1.62, Hamming 1.64, Flattop 3.73. (according to Siglent's implementation)

Blackman and especially Flattop are the most universal and useful window functions in practice. You definitely should stick to these two as long as you cannot prove that some other window would actually work better in your specific application.

Thus: Δf = RBW / 4 (rounded) in case of the flattop window or RBW / 2 for Blackman.

To get the proper settings for any given FFT-BW and RBW pair, proceed as follows:

Determine the FFT sample rate: SR = FFT-BW * 2 [Sa/s];
Determine the number of FFT points: FFT-pts >= SR / Δf [pts];
Determine the time base: TB >= FFT-pts / (10 * SR) [s/div];

As mentioned earlier, you can lower the FFT-sample rate by setting a constant acquisition sample rate; this can be useful when you want really low FFT-sample rates but do not want to use very slow time base settings, which would slow down the acquisition considerably.

In general, we need to keep in mind that the FFT doesn't process the entire screen width. It depends on the time base: at 500 ns/div we get a record length of 10 kpts and an FFT-length of 8192 points and this covers about 82% of the screen width. In many other scenarios, like at 100 or 200 ns/div, it would be just 51.2%.

Whenever you want to analyze a single event like e.g. a short transition, be aware that the FFT might be completely blind for the right half of the screen. As a consequence, do the following:

- place the event between 20-30% of the screen width.
- to get identical results independent of the horizontal position, use the Rectangle window.

Setting up the FFT

Even from the best FFT implementation, we can only expect good results as long as the scope has been set up properly for that specific task. How many so called "reviews" have we seen where FFT has been engaged and some scope settings randomly altered just to get a halfway plausible but actually not very meaningful FFT graph, which was then either praised or criticized?

Of course we can get away with some quick & dirty setup if the requirements are low, but even then we should never ignore the most important parameters like FFT bandwidth, which should always cover the full signal spectrum, otherwise aliasing artefacts could easily spoil our measurement results.

For optimal speed, frequency resolution and dynamic range, we need to put a little more effort into a proper setup, which has quite different requirements compared to the usual Y-t view (aka time domain). Below there is a complete checklist how to properly set up the DSO for analysis in the frequency domain (most of these topics should be obvious, but still listed for completeness):

1. Set acquisition mode to normal. Use ERES only for a good reason and stay away from average. Avoid Peak Detect under all circumstances and without any exception!
2. Use edge trigger in auto mode to make sure signal acquisition doesn't stop even when the signal amplitude drops below the trigger sensitivity. FFT doesn't require a stable trigger, so you can also use AC-line trigger for that.
3. Determine the lower bandwidth limit for the FFT analysis. If it is >10 Hz, use AC-coupling for the input channel to ensure maximum dynamic range even with large DC offsets and/or high input sensitivities. If DC-coupling has to be used, use the vertical position control to compensate for any DC offset, so you can optimize sensitivity and get the highest possible dynamic range.
4. Determine the upper bandwidth limit for the FFT analysis. In order to avoid aliasing artifacts, this should not only cover the desired analysis bandwidth, but include the highest expected input frequency. In general, it's best to start with a higher upper bandwidth limit and reduce it only after it has been confirmed that there is no significant signal content above the desired final limit.
5. Choose the frequency step size according to the explanations given earlier in this article, which would be about one quarter of the required resolution bandwidth when using the Flattop window.
6. Find an appropriate set of horizontal time base setting and the number of FFT points; refer to the explanations given earlier in this article. You should watch the displayed FFT parameters while altering the time base and double check that they match your expectations. Be aware that the desired resolution bandwidth might not be achievable due to the limited choice of sample rates and FFT lengths and/or the maximum specified FFT length of 2 Mpts.
7. Engage FFT mode, select the correct source channel and start with Split Screen mode.
8. Set the vertical gain so that the peak amplitude of the input signal is between ±2 to ±4 divisions.
9. Set the horizontal FFT display parameters according to the bandwidth you want to display and select linear/decade mode for the frequency axis. Decade is advantageous for wideband measurements, whereas linear is best for narrowband applications.
10. Set the vertical FFT display parameters, i.e. the desired level units (dBV or dBm, forget volts!) and make sure the external load impedance matches reality whenever working with power levels, i.e. dBm. Set the reference level and vertical scale so that the FFT amplitude range of interest makes best use of the available screen space.

11. Setup (at least) an automatic peak-to-peak measurement for the input channel. During frequency domain analysis, especially in Exclusive mode, keep an eye on the V_{PP} measurement for the input channel to make sure no overload occurs.
12. Make sure the desired window function is selected.

Hint: stay in Split Screen mode until the amplitude setup is finished and the levels are reasonably stable, then switch to Exclusive mode. By keeping an eye on the peak-to-peak measurement of the input signal, you can still detect an overload condition instantly; the scope indicates that by displaying **>** in front of the measurement value.

Example: Pk-Pk   >851.875mV instead of Pk-Pk   755.000mV.

FFT Window Functions

For the ones who try to understand the consequences of certain settings in the FFT analysis – this is about the window functions.

Why are there so many different windows (only few of them available on the SDS800X HD)? What is the best window to use?

There have been times when processor systems haven't been nearly as powerful as today. Non-RISC architectures with just 1 MHz clock frequency and less than 1 kB RAM were not uncommon during the seventies of last century. Instruments that could compute a FFT at all have been rather exotic, and FFT-lengths like 64 points were quite common. In the light of this, there is no wonder that less than ideal FFT-window functions optimized for certain tasks were popular.

Sometimes there are descriptions about the benefits and drawbacks of the various window functions, yet most folks would rather not care and want a universal setting that works for them almost every time. Just like with a traditional SA (spectrum analyzer) with analog RBW (resolution bandwidth) filters in the final IF (intermediate frequency) path. And fortunately, there is one…

There are several features of a window function, and two of them are amplitude accuracy and resolution bandwidth. If we look at just these two properties, then the rectangle window would have the narrowest resolution bandwidth but the worst amplitude error, whereas it's just the opposite for the Flattop window.

So, whenever we need the best frequency resolution, we just sacrifice a bit of accuracy and use the Rectangle window?

It's not that simple. An FFT divides the entire analysis bandwidth into frequency bins. If, for instance, we have an FFT-length of 32768 points, then we get 16384 such frequency bins and at an effective sample rate of 2 MSa/s, each of them will be 61.04 Hz wide. In this case, 61.04 Hz is the bin width and the bin spacing at the same time. The center of a bin will always be an integer multiple of the bin width.

Now FFT-windows behave differently, depending on the offset of the input signal frequency from the bin center. I did a selectivity test for the various window functions available on an SDS800X HD and used the before mentioned parameters:

FFT-sample rate = 2 MSa/s

FFT-Length = 32768 pts
Bin-width = 61.03515625… Hz

The test will be for amplitude accuracy and the -20 dB, -40 dB and -60 dB selectivity. I define the latter at the frequency distance for a -20, -40 or -60 dBc signal to still produce a visible 3 dB peak in the spectrum (and not drowned out by the leakage of the neighboring 0 dBc reference signal).

The -3 dB bandwidth might be most important for characterizing the passband of any two-port network, but for a filter, where the selectivity is the main concern, the bandwidth at a useful attenuation is even more important.

The metric of a filter shape factor exists, which is usually defined as the ratio of the filter bandwidth at -60 dB and -6 dB. The shape-factor might even be the most important property of any RBW filter at all, because it ultimately defines selectivity. Consider what a proper BP (Band-Pass) filter, as it might be found in any traditional SA, looks like:

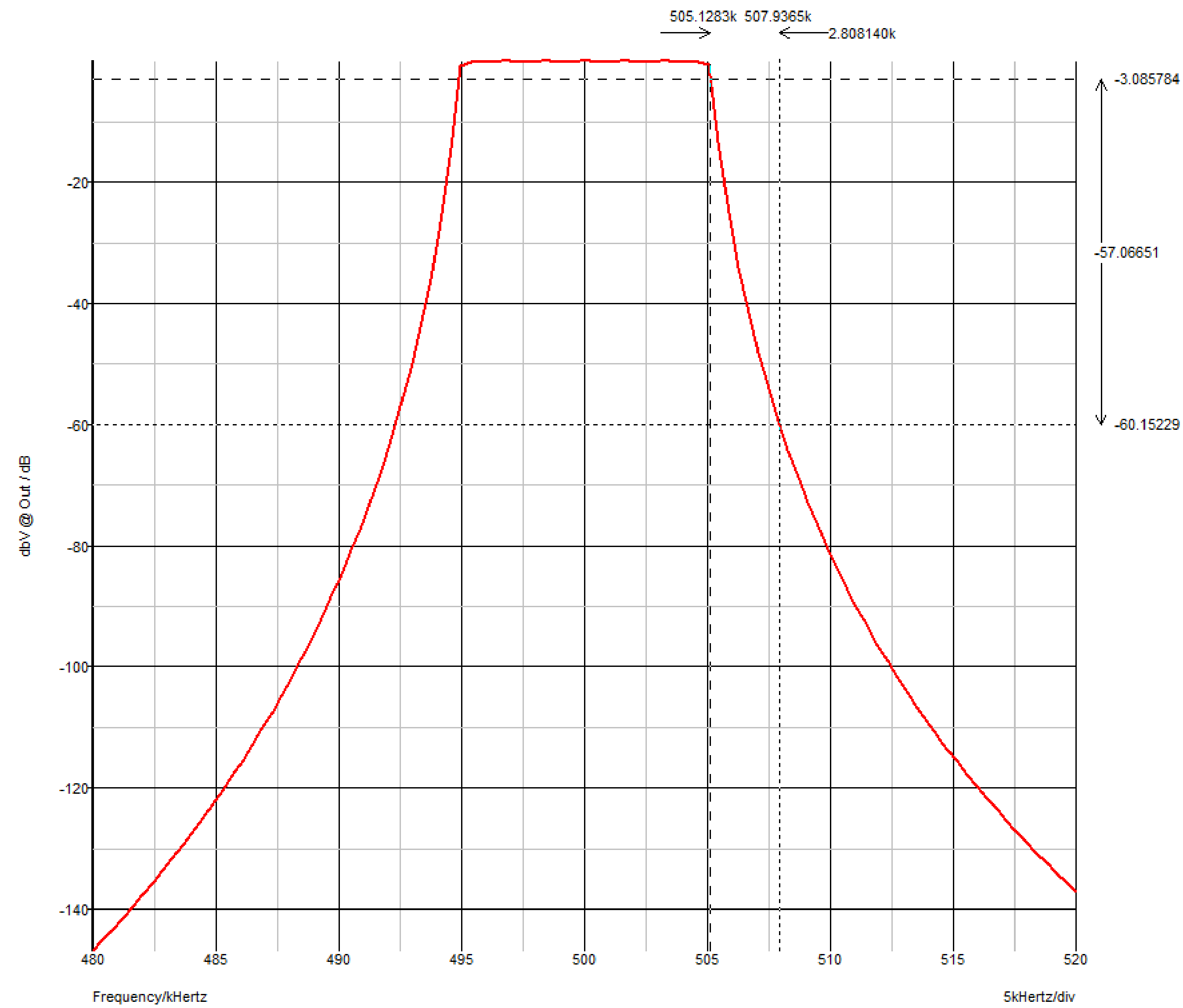

Fig. 102 Ref-BP_Cheby9_0.2dB

This is a 9th order Chebyshev BP filter with 0.2 dB passband ripple. It has a -3 dB bandwidth of about 10 kHz and less than 16 kHz bandwidth at -60 dBc. The shape factor is thus ~1.57. The lower the shape factor, the higher the selectivity.

What does it mean in practice? We can distinguish a strong signal at 0 dBc together with a weak signal at -60 dBc next to it, as long as the distance is at least ~13 KHz, i.e. 1.3 times the RBW. That is excellent and makes for a useful analysis in the frequency domain.

An FFT is quite different to a classic continuously swept spectrum analyzer. If we feed a stable sine wave into a swept analyzer, we'll get the frequency response plot of the RBW-filter.

The FFT doesn't plot a continuous filter response, but simply shows the outputs of the individual frequency bins, and ideally only the one bin covering the input frequency responds with the correct amplitude (and phase, but that's not displayed). It is like a huge filter bank with a bunch (16384 in our next example) of filters, all working in parallel. What we see is different from the RBW-filter response; we rather see the "leakage", i.e. signal outputs from neighboring "filters" (bins), sometimes even far away from the signal frequency - and there are amplitude errors for the main bin.

A 0 dBm test signal of 499.99342 kHz has been used for the first test, which is equivalent to precisely 500 kHz if the time base of the SDS800X HD were accurate. this is precisely 8192 times the bin width, hence the exact center of a bin.

What do we get?

Window	**Ampl.**	**Selectivity [Hz]**		
	[dBm]	**-20 dB**	**-40 dB**	**-60 dB**
Blackman	0.0	236.58	236.58	236.58
Flattop	0.0	366.58	416.58	416.58
Hamming	0.0	168.58	168.58	168.58
Hann	0.0	171.58	171.58	171.58
Rectangle	0.0	98.58	98.58	98.58

Table 4 Window selectivity bin center

Look at the rectangle window, with two signals as an example. The 0 dBm reference signal (carrier) and the 2nd signal at -40 dBc (with an enormous amplitude error of 2.8 dB), which creates just a 3 dB peak at a distance of 98.58 Hz,:

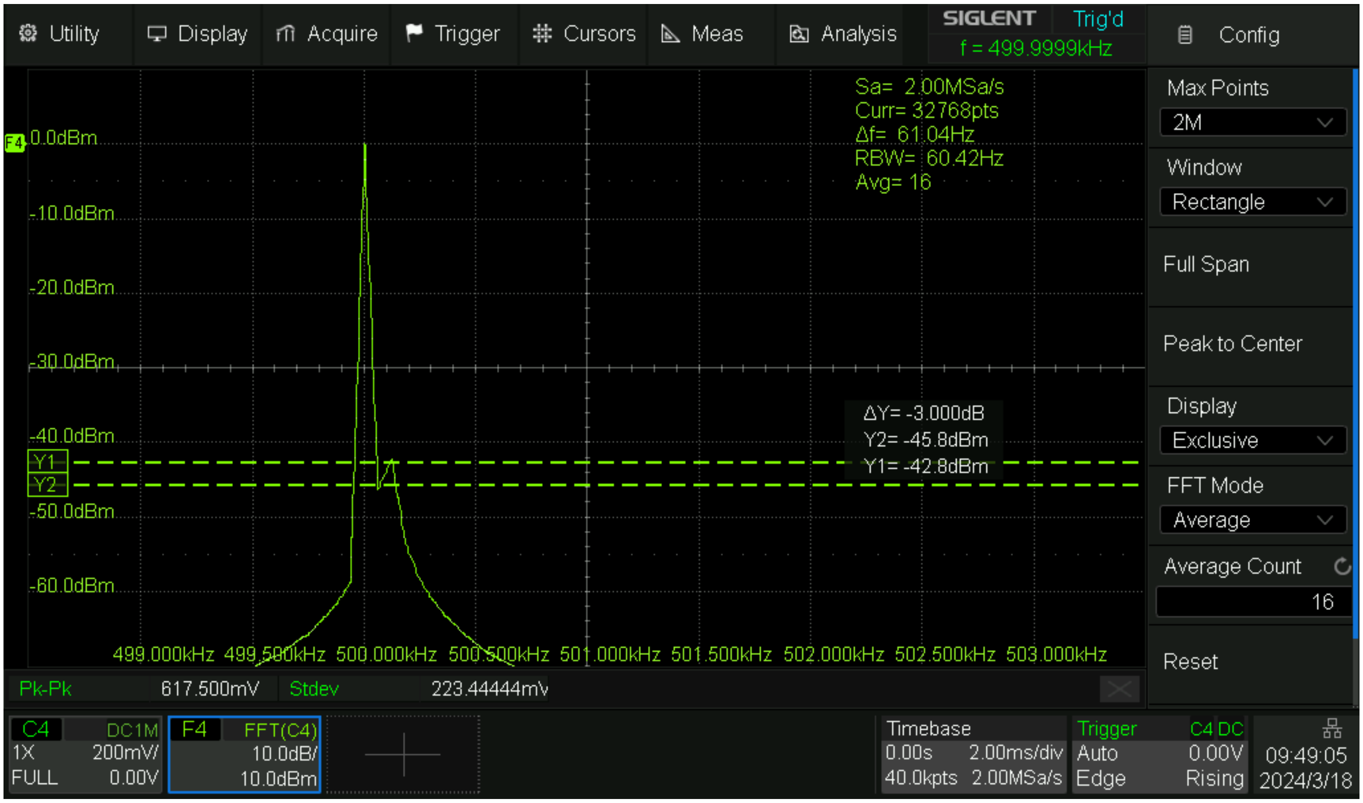

Fig. 103 SDS824X HD_FFT_Rectangle_0.0_S40dB_500092Hz

With a distance of just 1.615 bin widths, selectivity is quite good. Yet real-world signals will usually not be an exact integer multiple of the bin-width, so we need more tests.

In any practical application where the FFT of a general-purpose oscilloscope is to be used, we cannot freely define the signal frequencies, hence they will be more or less off center. Even if we could select a frequency, most related signals like intermodulation (mixer) products and spurs can still have any frequency offset with regard to the bin spacing. Consequently, we need to take the worst case into consideration, that is a frequency offset of half the bin-width.

A 0 dBm test signal of 500.02394 kHz has been used for the following test, this is precisely 8192,5 times the bin width, hence the exact bin-border for my individual sample of the SDS824X HD.

Window	**Ampl. [dBm]**	**Selectivity [Hz] -20 dB**	**-40 dB**	**-60 dB**
Blackman	-1.0	206.06	266.06	426.06
Flattop	0.0	386.06	386.06	451.06
Hamming	-1.8	196.06	196.06	-
Hann	-1.3	203.06	326.06	586.06
Rectangle	-3.8	381.06	2780.06	-

Table 5 Window selectivity bin border

This looks very different, doesn't it? All of a sudden, the rectangle window has 3.8 dB amplitude error and its selectivity isn't all that good anymore. In fact, it is unbelievable >45 bin-widths for the -40 dBc selectivity! By contrast, the Flattop window hasn't changed at all: the amplitude error is effectively zero as it was before and also the selectivity has only marginally changed. That means more than -60 dBc attenuation at 300 Hz (less than 5 bins) distance from the center.

Hamming does not have a 60 dB attenuation within a reasonable bandwidth – in fact it is so wide that I could not be bothered to measure it. The same applies to the Rectangle window, where even the -40 dBc selectivity test didn't reveal anything useful:

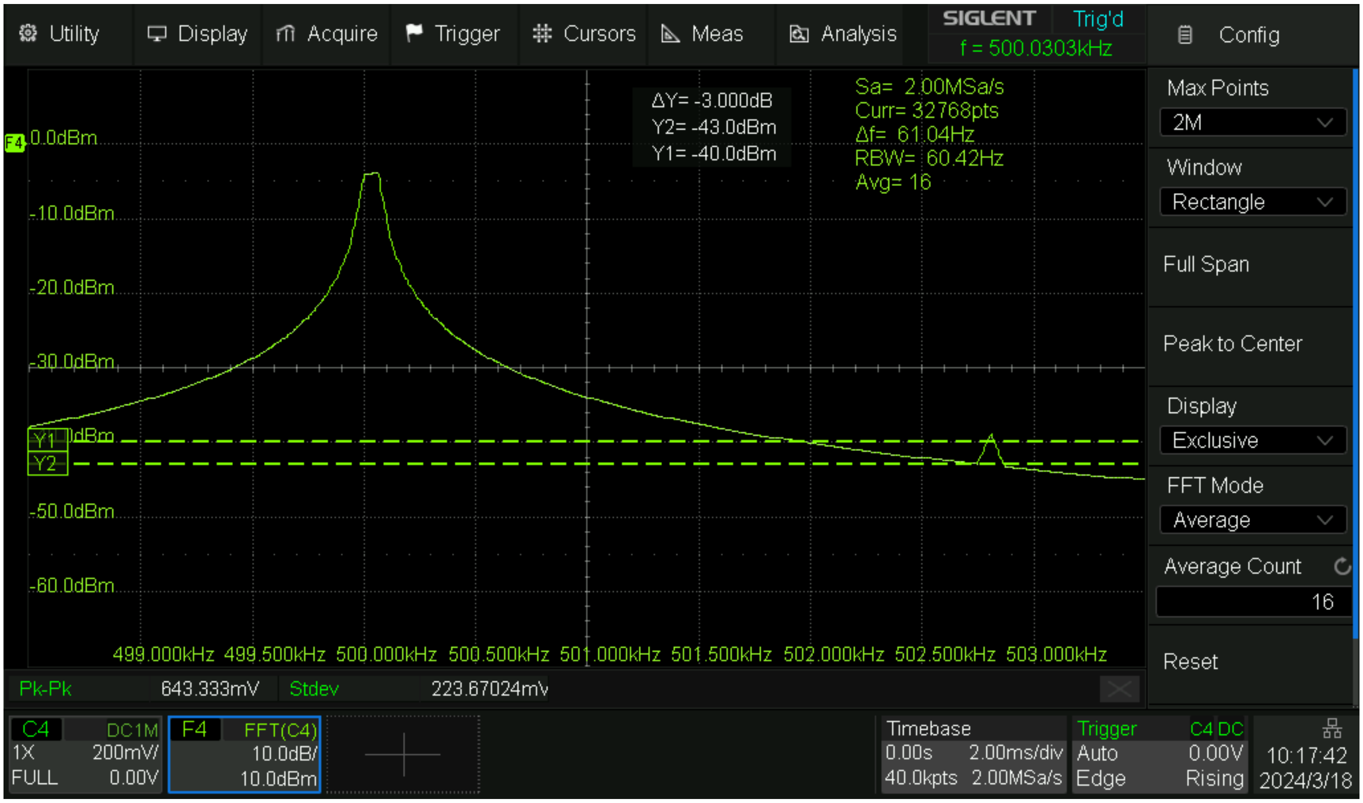

Fig. 104 SDS824X HD_FFT_Rectangle_0.5_S40dB_502800Hz

What do we want for a proper RBW filter for spectrum analysis, in order to get serious and accurate measurements?

1. High dynamic range. We have no use for a RBW filter that has no proper stopband attenuation, hence picks up all the garbage from the neighborhood.
2. Fast transitions into the stopband, which is equivalent to a low shape factor.
3. A reasonably flat passband without massive amplitude errors, as soon as the signal frequency gets off center a bit.
4. And most importantly, all parameters shall be constant and independent of the exact input frequency.

The only window that appears to be perfect in almost all regards is the Flattop window. It has a very high dynamic range, low shape factor, a totally flat passband, and even more important, its properties remain constant and do not depend on the signal frequency. Only downside: it has the widest bandwidth of all candidates. Yet in modern equipment, where at least 1 Mpts FFT have become standard, we need not desperately look for a windows function that sacrifices a lot of good properties just for a little narrower RBW.

Blackman is the only alternative that I can recommend from the selection in the Siglent SDS800X HD. It has less than half the RBW of the Flattop window and the shape factor is still reasonable. It works down to -60 dBc even with the worst-case frequency offset of half a bin width, and the passband flatness, hence also the amplitude error, is not too high. This is generally true for all window functions of the Blackman-family, especially Blackman-Harris. Siglent only implements the original Blackman window though.

There are special applications, like analyzing isolated single events, where the Rectangle window can have advantages, so it should still be considered in these situations.

You can take a look at all the remaining window functions in the attachment, albeit only for worst case frequency offset, indicated by “0.5” in the file name.

FFT Dynamic range

Since the introduction of the SDS1004X-E in early 2018, FFT has always been a strong point of Siglent DSOs. The SDS800X HD is no exception and numerous examples have been published already in this review, as the FFT is an incredibly universal tool to demonstrate fundamental features like frequency response, noise distribution and signal spectra in general, as well as measure distortion products, spurious signals and weak signals, deeply buried under the noise floor.

One of the concerns with the FFT in DSOs is the dynamic range. For 8-bit acquisition systems, this is only about 49 dB according to the textbook, as it is some 73 dB for 12 bits. And indeed, we need to be careful when acting outside these “guaranteed” dynamic ranges. Yet the wonders of process gain in an FFT and other resolution enhancement techniques can extend the usable dynamic range quite a bit, and this shall be demonstrated for the SDS800X HD in some best-case scenario.

What is the “best case” scenario? It is a frequency at or above 1 MHz in order to escape the 1/f noise, but at the same time the frequency should be low enough so we can get rid of all the high frequency noise by using the 20 MHz bandwidth limiter plus an additional steep 20 MHz lowpass filter. Consequently, we practice math on math (the SDS800X HD could also have combined it in one formula instead) and calculate the FFT on the filter output instead directly on the channel data.

Two signals from an AWG at 9.9 and 10.1 MHz are fed into a wideband signal combiner, where the second signal goes through a fixed 20 dB attenuator together with a 1 GHz step attenuator before it hits the combiner, whose output is connected to the SDS824X HD channel 4 input via another 10 dB inline attenuator and a 50 ohm through terminator. Of course, the tolerances of these various components sum up, so I have calibrated the whole setup for a 0 dB setting of the step attenuator first by means of the AWG output levels, but left them untouched for all subsequent measurements. As a consequence, only the tolerance of the step attenuator will affect the results. The step attenuator is a Wavetek 5080.1 with a specified tolerance of +/-1 dB up to 400 MHz. After using it many decades, I can tell from experience that it thankfully is clearly more accurate than that.

In order to get a low RBW (Resolution Bandwidth), hence also a low noise floor, we don’t want an excessively high sample rate; 100 MSa/s and a Nyquist frequency of 50 MHz is plenty to deal with a 10 MHz signal and also a 20 MHz FIR filter.

Here's the calibration result:

Fig. 105 SDS824X_HD_FFT_DR_500kpts_10MHz_40dB

The error is <0.05 dB. Going from there, here's the measurement for 80 dB level difference:

Fig. 106 SDS824X_HD_FFT_DR_500kpts_10MHz_80dB

The error is < 0.25 dB, which could well be attributed to the step attenuator.

Now let's try 100 dB with the same setting:

Fig. 107 SDS824X_HD_FFT_DR_500kpts_10MHz_100dB

The measurement error is still <0.7 dB, yet the 2nd signal is almost down in the noise.

Up to now, we've only computed a 512 kpts FFT, so let's try 1 Mpts now, thus cutting the RBW in half.

Fig. 108 SDS824X_HD_FFT_DR_1Mpts_10MHz_100dB

Now the 2nd signal is clearly above the noise floor, yet the total error is even slightly worse now at ~0.85 dB.

Nevertheless, taking the 100 dB dynamic range and the tolerances of the step attenuator into account, this sis till a remarkable result.

FFT Speed

The FFT speed is limited to about 18 computations per second, which can be maintained up to 64 kpts. We still get more than 10 frames per second for 128 kpts and from there it scales proportionally, i.e. 5 frames for 256 kpts and so on up to 2 Mpts, where the update rate is only about 0.75 FFT computations per second. At this setting, the highest absolute speed of ~1.6 Mpts/s is reached.

Distortion measurements

General purpose oscilloscopes cannot have ultra-low distortion frontends, especially nowadays, where even entry level instruments start at 70 or 100 MHz bandwidth. And even a low-end device like the SDS800 goes up to 200 MHz for the top model within the line, and I'd bet the integrated PGA (Programmable Gain Amplifier) in these devices has more than 0.5 GHz bandwidth.

To cut a long story short: the usual techniques to keep distortions down in audio devices, particularly global feedback, cannot be applied to wideband amplifiers. Taking this into account, it's still amazing what can be achieved with modern integrated circuits, yet it's the main reason why 12-bit DSOs fail to come even close to 12-bit ENOB.

Let’s start with the harmonic spectrum of a low distortion 10 MHz sine wave:

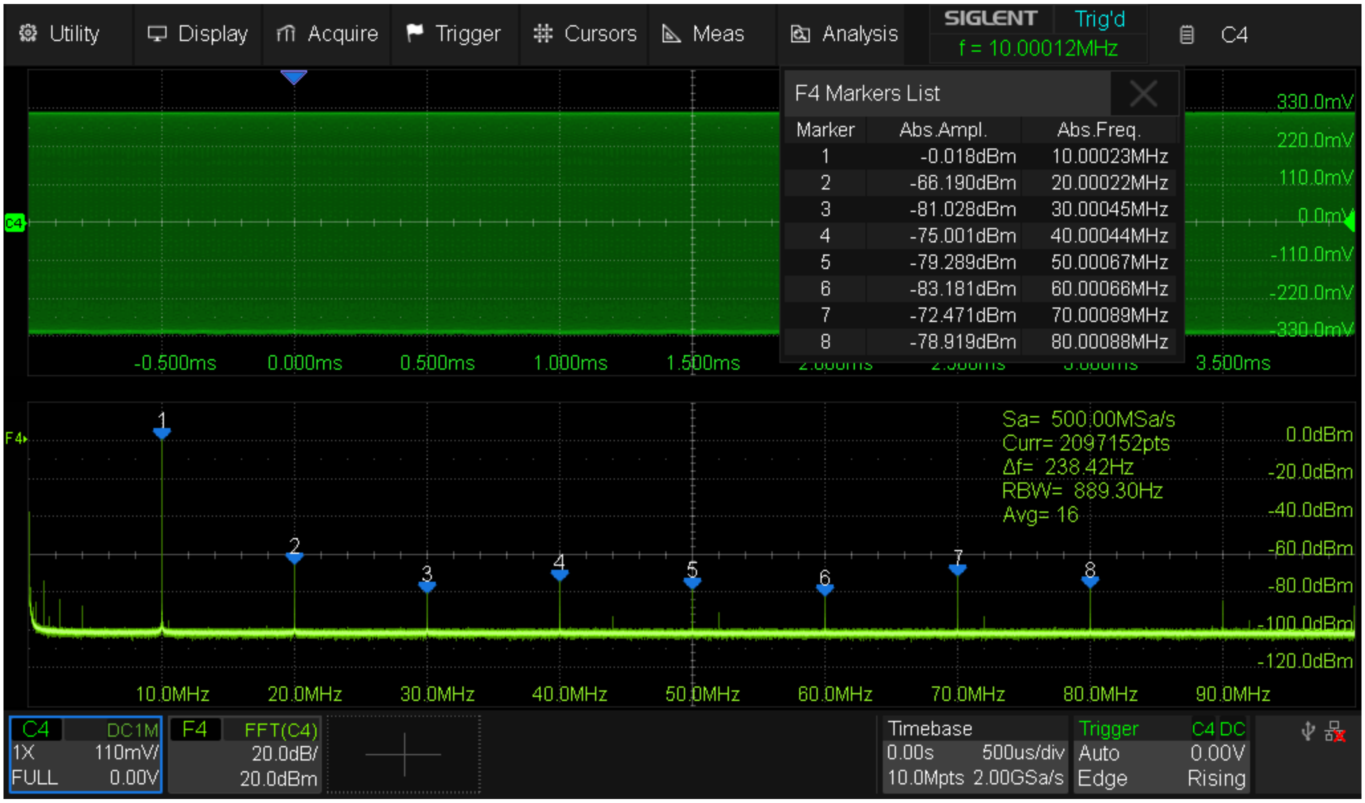

Fig. 109 SDS824X_HD_FFT_THD_10MHz

Strongest harmonic is the 2nd at -66 dBc. Since all other harmonics have considerably lower amplitude, we can safely state that THD is about 0.05%, which isn’t bad at all.

Yet such results cannot be guaranteed; the individual gain stages within the PGA can have differences in linearity, so we need to know our instrument and take note of the weak as well as sweet spots within the vertical gain range. My particular sample of the SDS824X HD has a weak spot at exactly 100 mV/div, whereas all the settings >100 mV/div up to (at least) 110 mV/div yield results like the one shown above.

Whenever we do distortion measurements of the DSO frontend, we need to be confident that the distortion products actually come from the DSO and not the signal source. It can be tricky to verify this, hence a different approach might be more precise: the dual tone intermodulation test.

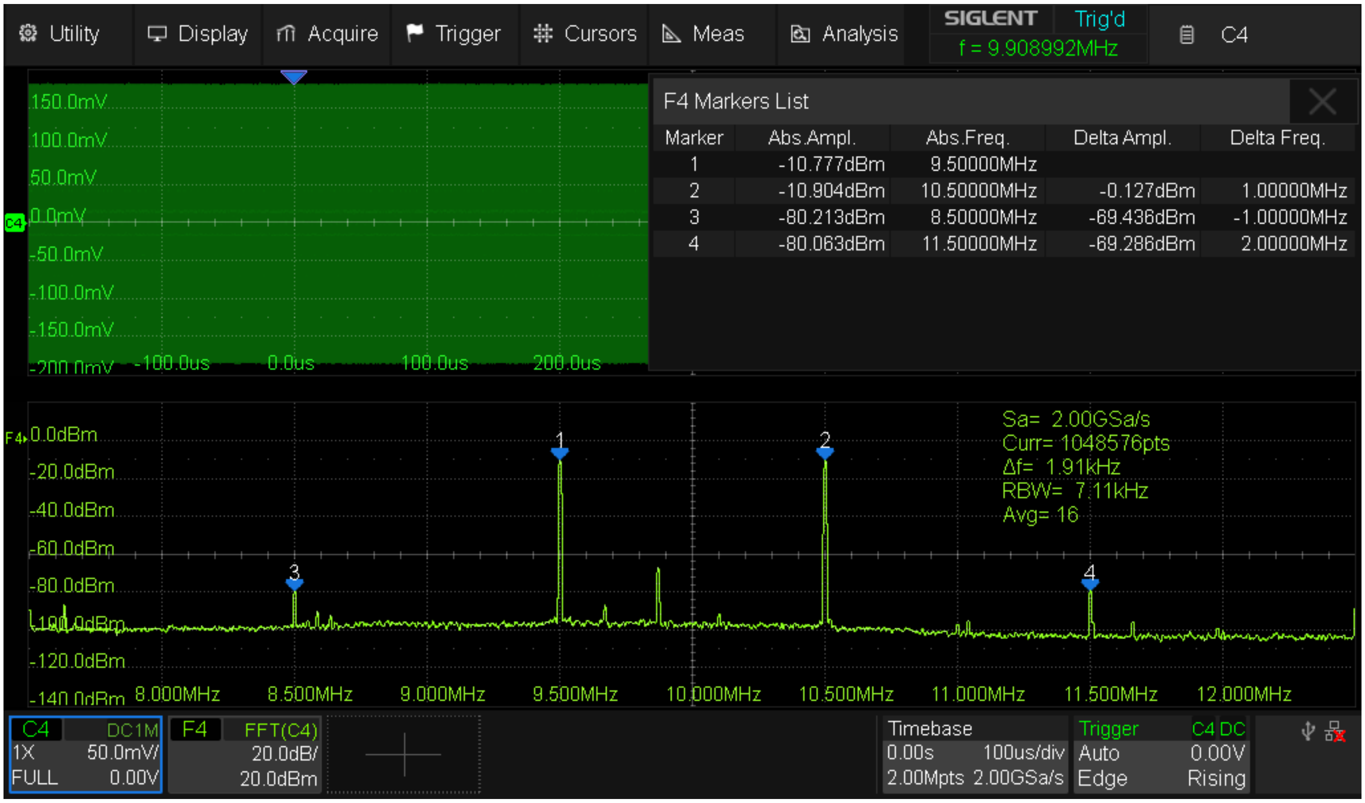

Fig. 110 SDS824X_HD_FFT_IMD_10MHz_95FS

Two independent +6 dBm signals at 9.5 and 10.5 MHz are fed into a resistive wideband power combiner. To ensure proper isolation between the two signal sources and avoid intermodulation distortion at their output stages, a 10 dB inline attenuator has been added to each generator output. Together with the 6 dB attenuation of the splitter, we'd expect two -10 dBm input signals.

Quite obviously, the external termination of the DSO input, which cannot compensate for the ~17 pF input capacitance, is responsible for the higher-than-expected losses, hence inaccurate input level. Since we use relative measurements anyway (delta amplitude in the markers list), this doesn't matter though.

A vertical gain of 50 mV/div doesn't appear to be a weak spot in this instrument, so we get respectable -69 dBc for the third order intermodulation products.

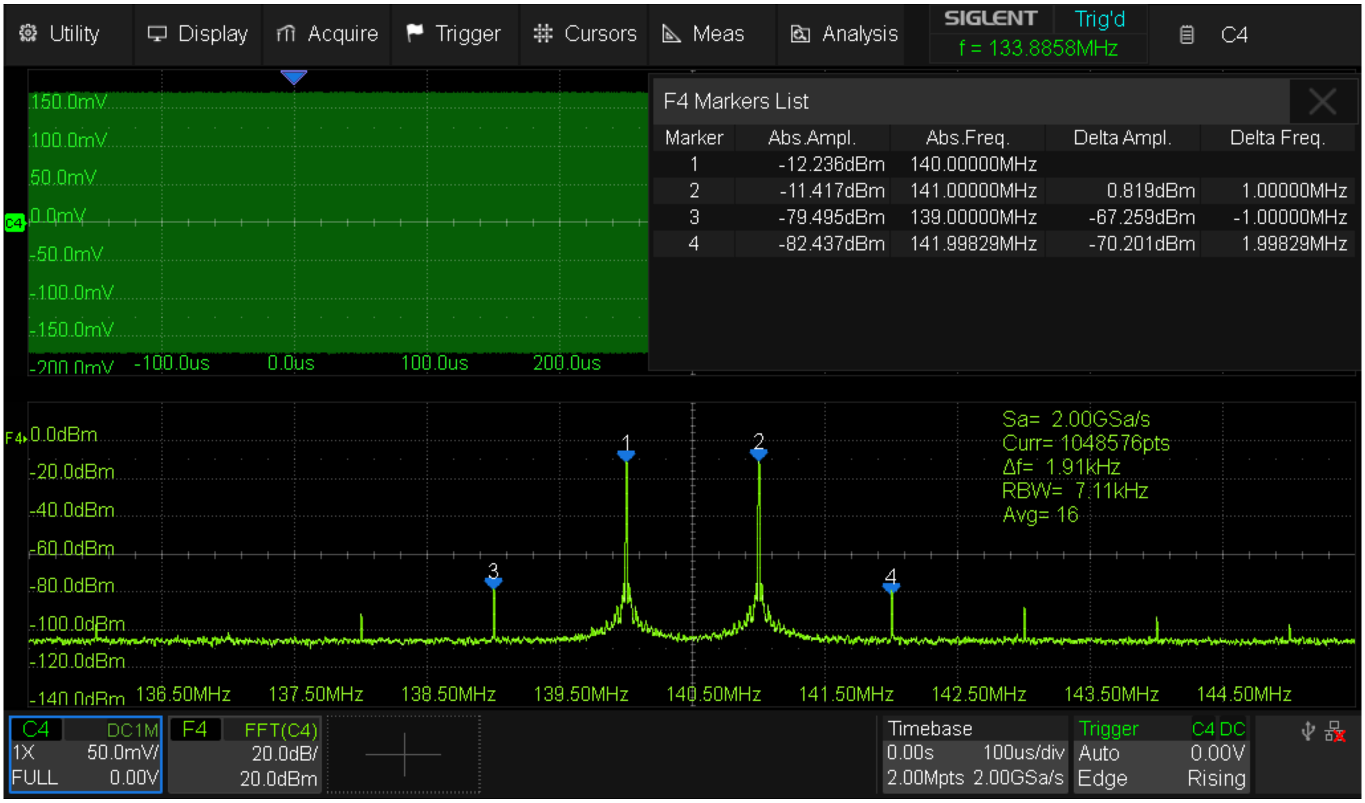

Fig. 111 SDS824X_HD_FFT_IMD_140MHz_95FS

Two independent signals at 140 and 141 MHz are fed into the power splitter. At these higher frequencies, the problems with the external termination get even more obvious and instead of the nominal level of -10 dBm we get up to 2.2 dB less. Once again, we don't care because we're only interested in relative levels.

The vertical gain of 50 mV/div from the last test is used again and we get respectable -67 dBc for the third order intermodulation products. Other than an OpAmp with global feedback, distortion performance does not necessarily get much worse at higher frequencies.

Analysis

Counter

The SDS800X HD does not provide a DMM, but it has at least the Counter application. It can be used as a frequency counter or totalizer. I don't see much use in the frequency counting function, chiefly because the automatic measurements can do exactly the same – and even on more than one channel at a time – and then we have the always visible 7-digit trigger frequency counter on top of that (I wouldn't ever want a scope without that feature!).

But the Counter is still not useless, because it always works on undecimated data just like the trigger frequency counter, but with complete statistics if so desired.

And then, the counter can also be configured as gated totalizer:

Fig. 112 SDS824X_HD_Totalizer_Gated

Channel 3 is fed with a 1 ms wide pulse as a gating signal, channel 4 is connected to a 10 MHz sine source. Consequently, a single shot acquisition results in 10000 hits = 10 khits. This is deadly accurate because both signals come from the same AWG, hence both signals are derived from the same OCXO (whose accuracy is irrelevant in this application, yet is specified <100 ppb)

Other than the pulse counts in the measurements, the counting process can be controlled by an external signal.

Bode Plot at a glance

Instead of showing an inexpressive first order RC-lowpass filter demonstrating less than 40 dB dynamic in the audio range, I'd rather check the most important characteristics of a Bode Plot: frequency- & dynamic range and accuracy.

For this, I've refrained from using inline terminators at the scope inputs but fed them from 50 Ω sources directly via ~25 cm long coaxial cables. The source resistance of 50 Ω, together with the cable and scope input capacitances, forms a first order lowpass filter at ~10 MHz. This can also serve as a warning how even very short cables can introduce significant amplitude errors at relatively low frequencies, as long as a transmission line is not properly terminated.

We can see this characteristic when using the "Vout" mode of the Bode Plot, where we get the absolute amplitude of the DUT output (where the DSO itself represents the DUT).

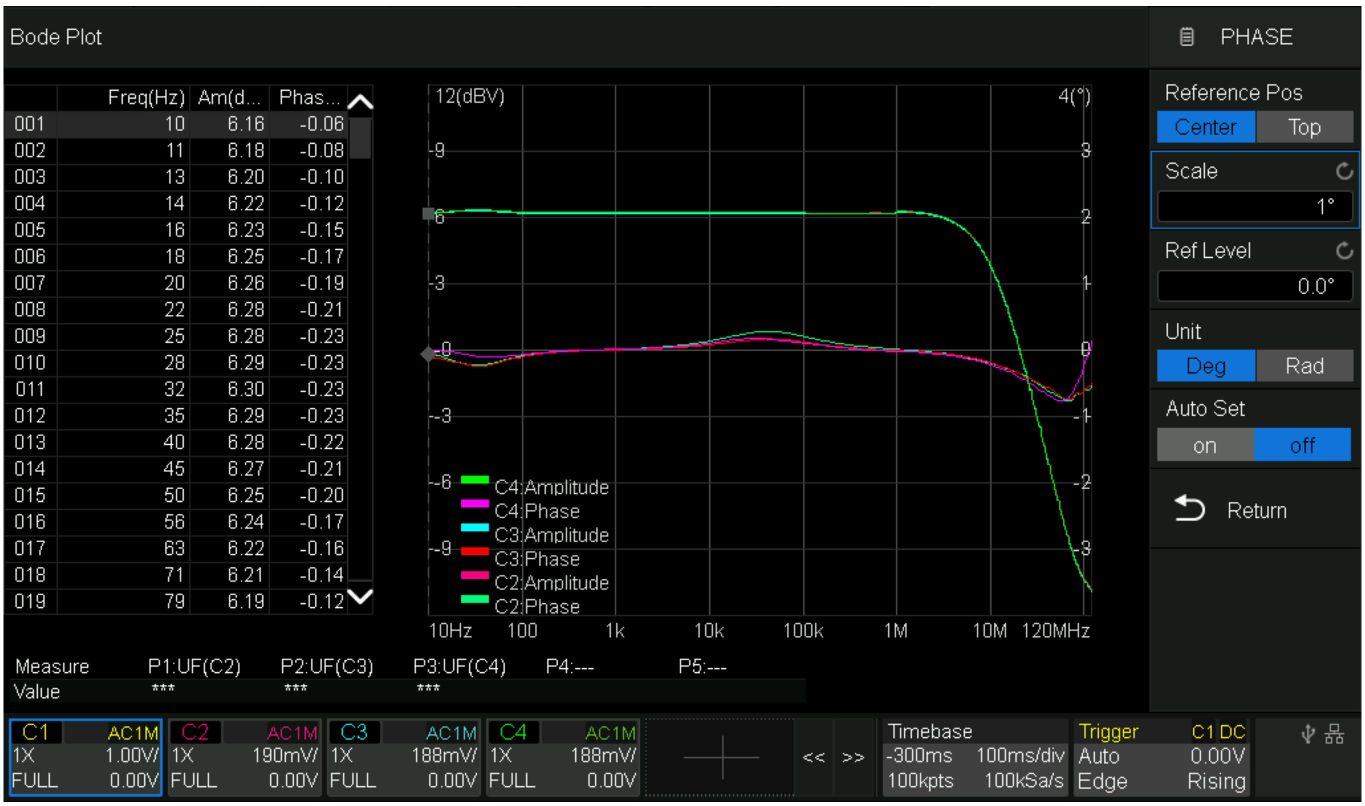

Fig. 113 SDS824X HD_Bode_1M_Vout

The amplitude drops quite significantly above 10 MHz. It is not the 20 dB/decade like a classic first order lowpass – and this is for a number of reasons that I won't discuss in this article. Bottom line is, that even with very short cables, accuracy of the absolute signal level is gone already at moderate frequencies of a couple MHz.

The phase plot does not resemble this, as it stays within +/-1° up to 120 MHz quite easily. It almost looks like this would not be a minimal phase system, yet it's just the nature of a multi-channel oscilloscope, where the input signals are always phase aligned.

When using the relative (Vout/Vin) mode (as we usually do), things look completely different:

Fig. 114 SDS824X HD_Bode_1M_S21

Bode Plot now shows the difference between reference channel 1 and the other channels. It is indicative of the quality of the SDS800X HD that the differences between channels are really negligible: less than 0.3 dB amplitude error as well as less than 1° phase error up to 120 MHz, and almost no differences between channels 2-4, speaks for itself.

Let’s check the accuracy and dynamic range now. Two signals are used to visualize a 60 dB amplitude difference. This time, 50 Ω inline termination has been used.

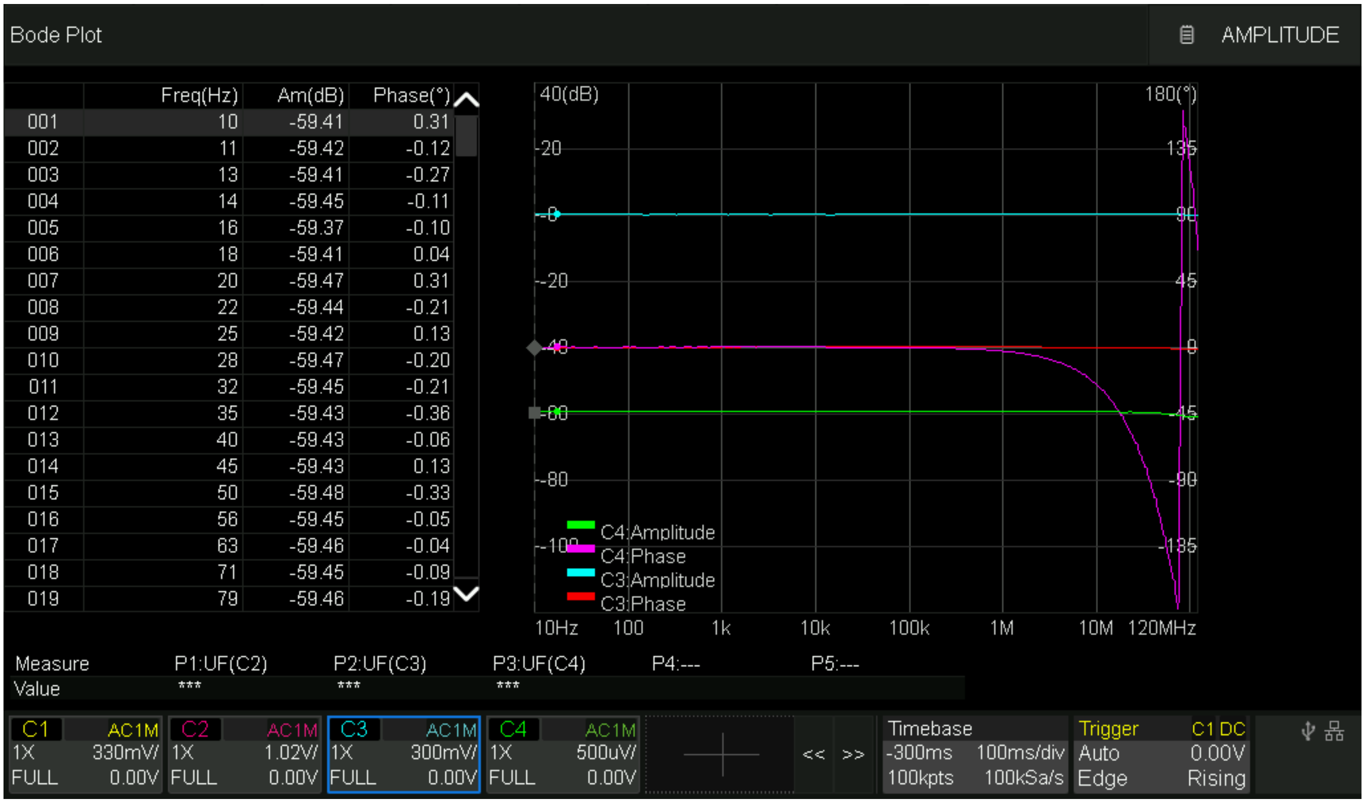

Fig. 115 SDS824X HD_Bode_50_S21_60dB

There is a significant phase difference, and this comes from the additional 3-stage step attenuator + Inline attenuator + some 50 cm additional coaxial cable for channel 4.

As a final experiment, here is a 100 dB amplitude difference (phase has been adjusted by means of the channel skew parameter):

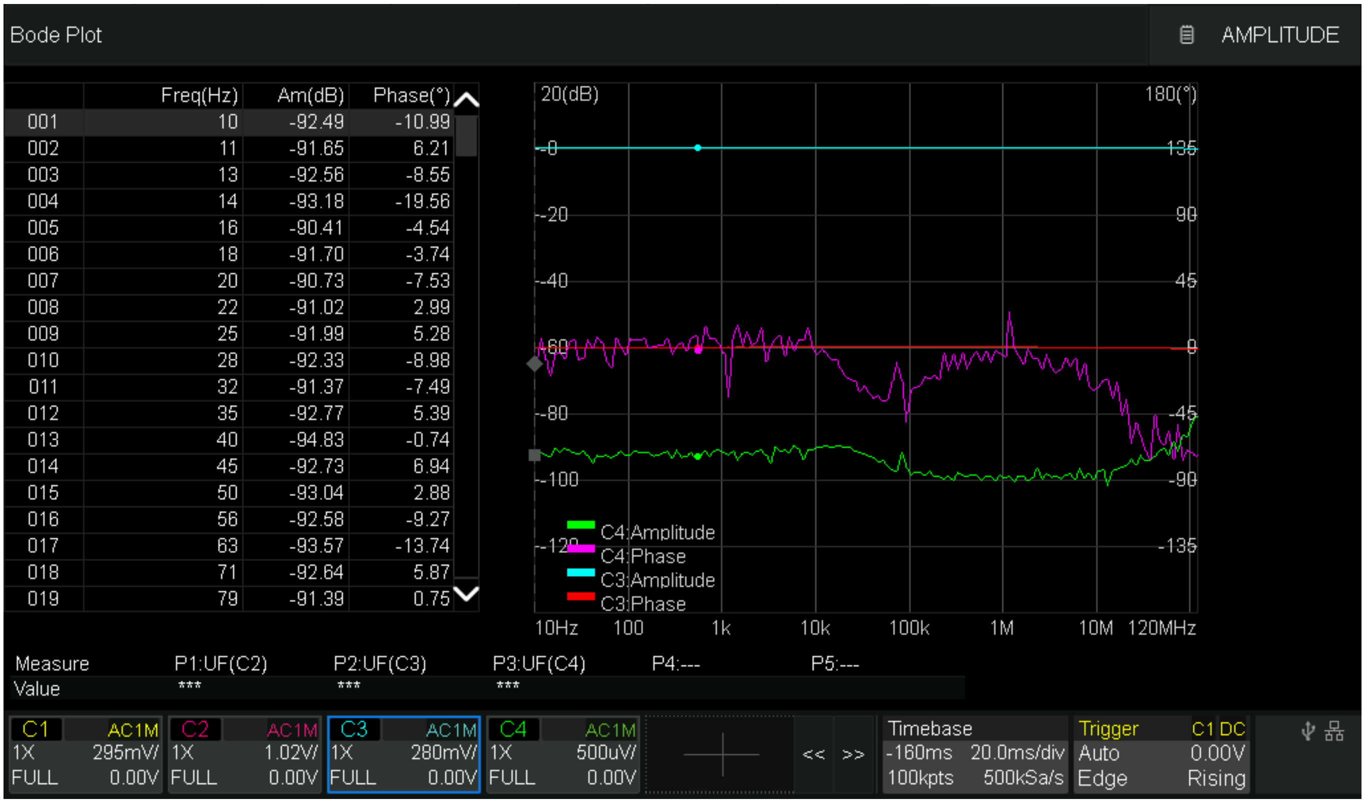

Fig. 116 SDS824X HD_Bode_50_S21_100dB

Noise is getting a major problem, yet amplitude measurements can still yield useable results in the range 100 kHz to ~20 MHz.

The reference level is low (~570 mV$_{RMS}$), hence channel 4 input sees only 5.7 µV$_{RMS}$!

I've not nearly exploited the dynamic range of the SDS800X HD, which could handle up to 28 V$_{RMS}$ (but then with beefy external >16 W terminators) if the need should be.

Bode Plot Example

Here is my standard test for Bode Plot: a simple 455kHz IF filter, consisting of a Kyocera KBF-455R-20A ceramic 6 element filter with two resonant 2nd order L-matching networks for the 50/1500 Ω impedance transformation at both the input and output.

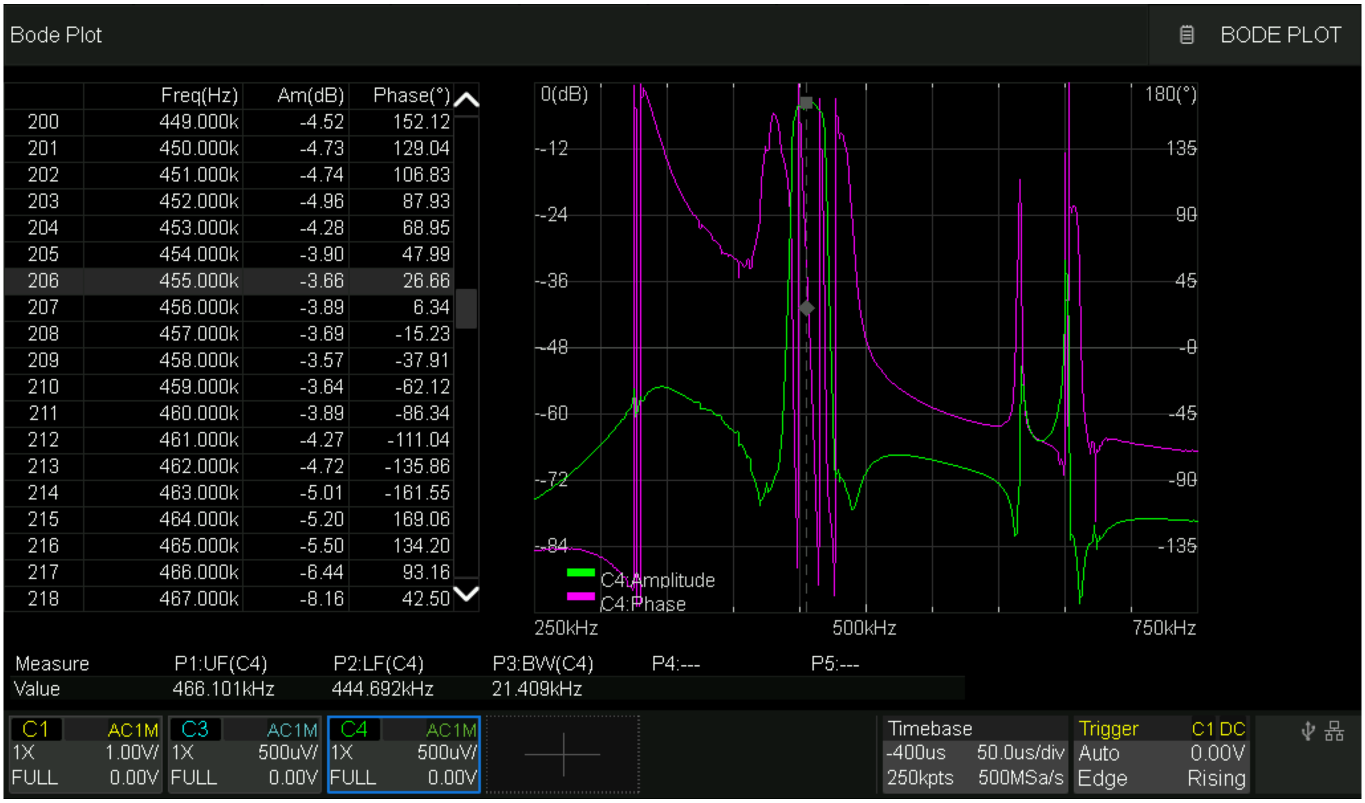

Fig. 117 SDS824X HD_Bode_50_S21_IF455kHz

A complex structure like this has a somewhat chaotic phase response, especially in the passband and the transitions into the stopband. The amplitude shows nice steep transitions into the stopband, yet there are some unwanted resonances as is typical for this type of filters. The important fact is that we need high frequency resolution to capture all the fine details. Furthermore, this test demonstrates at least 90 dB dynamic range.

The data table has been adjusted to the nominal center frequency of the filter, which should be 455 kHz. A vertical cursor marks the selected frequency in the plot and from the table we can see that insertion loss of this filter is some 3.66 dB. The frequency with the lowest attenuation of 3.57 dB is 458 kHz though. This “data cursor” is independent of the manual cursors that are available in the Display menu.

I’ve set up some specific measurements: UF (Upper Frequency), LF (Lower Frequency) and BW (Band Width) to characterize the most important properties of the filter. We can see at a glance that the bandwidth is precisely 21.4 kHz.

For those bothered by the complex phase plot, there is always the option to disable any trace we like:

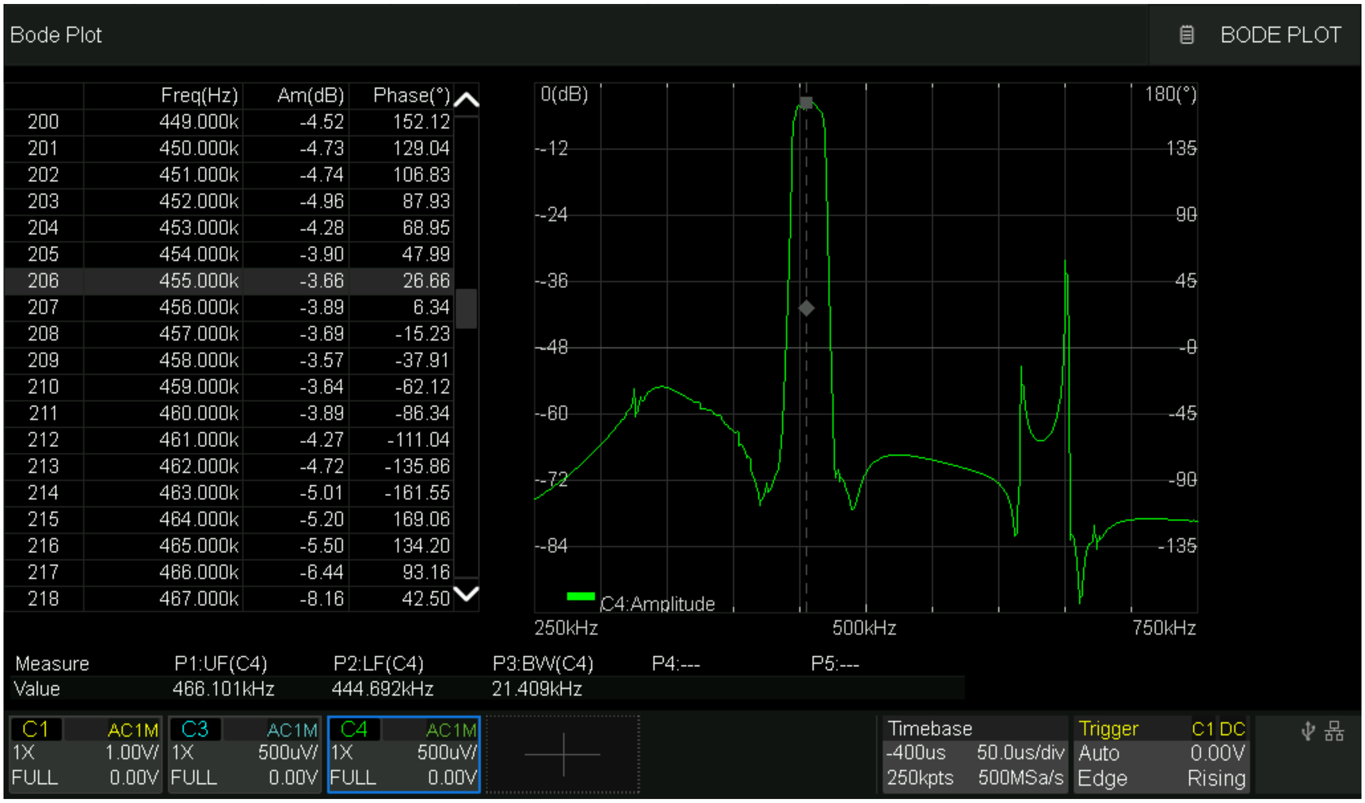

Fig. 118 SDS824X HD_Bode_50_S21_IF455kHz_NP

SPI Speed Test

We could ask the question: how fast an SPI data stream can we decode with a 200 MHz DSO? Huch much oversampling do we need?

For some DSOs it is said that they need a fair bit of oversampling for proper decoding. Yet the answer is, a proper implementation doesn’t require much in this regard, hence it should be perfectly adequate to have a bandwidth three times the SPI clock frequency. The sample rate on the other hand should be irrelevant, as long as the Nyquist criterion for the required bandwidth is fulfilled. That makes for more than 6 times the SPI clock frequency.

The SDS824X HD true bandwidth is limited to 200 MHz in full channel mode and its sample rate is 500 MSa/s. According to the hypothesis stated above, this bandwidth would allow a max. SPI clock of 66 MHz and the sample rate is sufficient for that. The only way to know how good this works is to try it out…

I felt adventurous, hence didn’t bother with 66 MHz, but tried 100 MHz right away:

Fig. 119 SDS824X_HD_SPI_100Mbps

The above screenshot shows a bidirectional (full duplex) 100 Mbps SPI data stream with a message length of 11 bytes. No special trigger has been used, just falling edge trigger on the /CS line.

The decoding works without issues, but how can we prove that the results are correct? Here's a simple test: I've replaced the MOSI signal with a phase synchronized copy of the 100 MHz clock signal, phase shifted 90° such that it is always sampled at its low state and then shifted by another 180° so that it is always sampled at its high state. This way we're putting the maximum stress on the acquisition chain, whereas it would be an easy task if we had just static signals for MISO/MOSI.

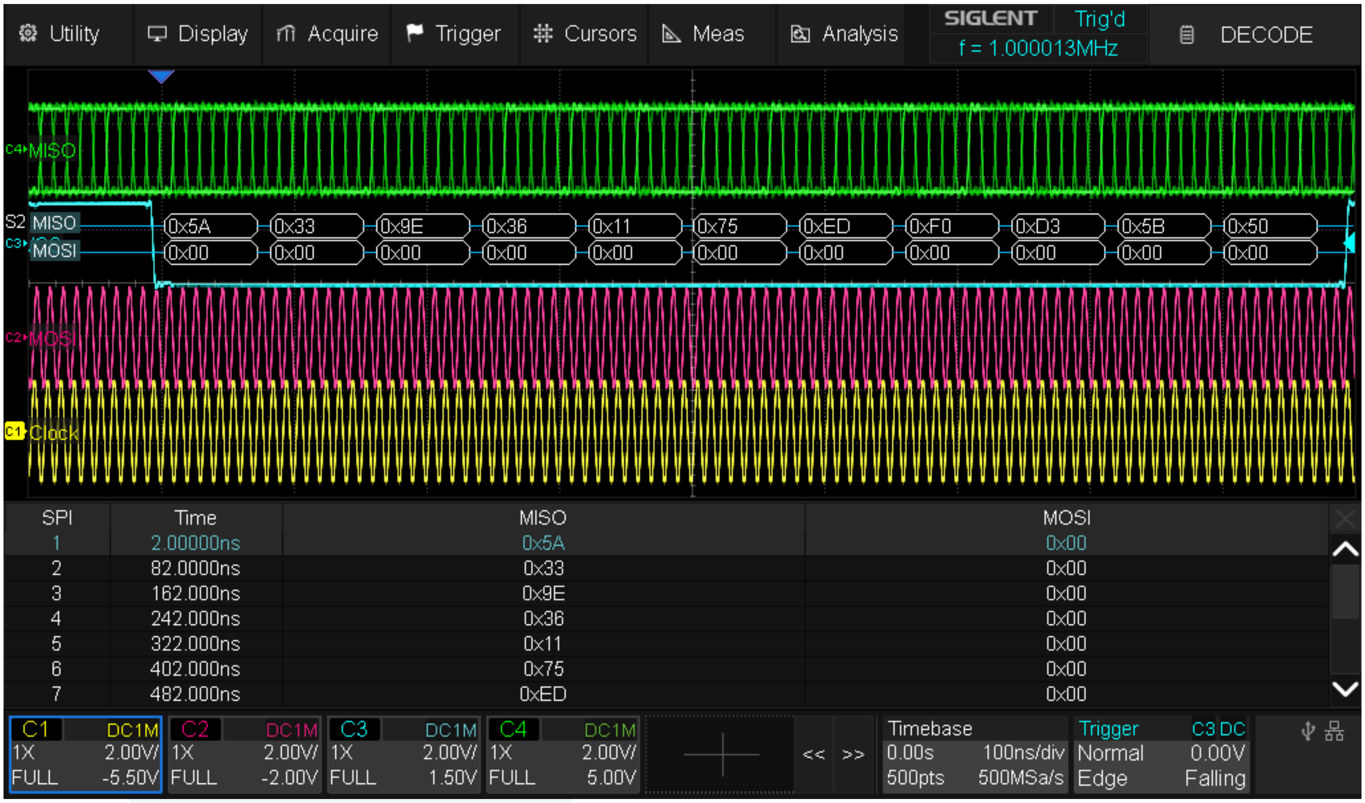

Fig. 120 SDS824X_HD_SPI_100Mbps_0

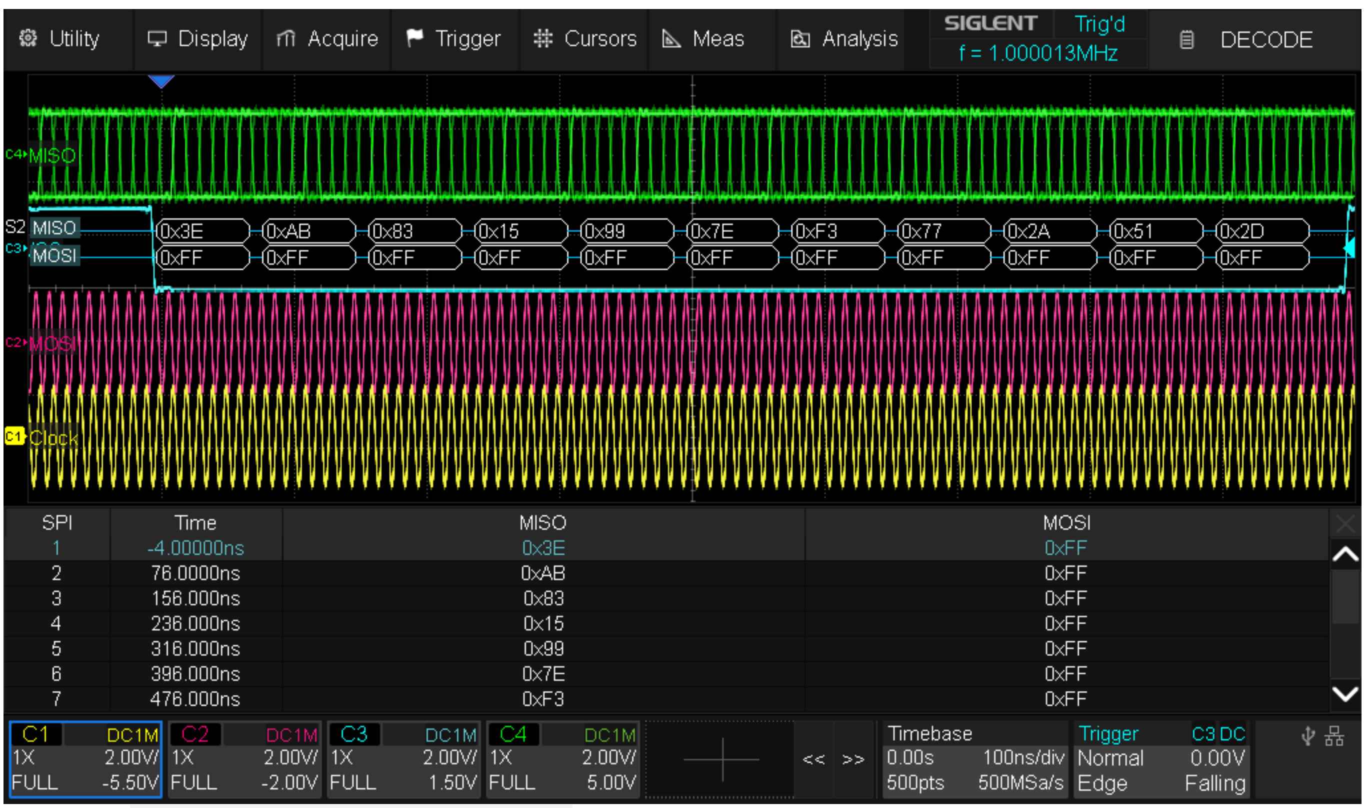

Fig. 121 SDS824X_HD_SPI_100Mbps_1

The expected results of all 0x00 for the first test and all 0xFF for the second one has been achieved, proving that there is a good chance to decode a 100 Mbps SPI data stream correctly with only twice the bandwidth and five times the sample rate.

If we trigger on the rising edge of the clock, we can produce an eye diagram which clearly shows that there is plenty of margin for correct decoding. Even the MISO signal, while slightly delayed, causes no problem in this scenario.

Fig. 122 SDS824X_HD_SPI_100Mbps_Eye

The fast pulses look very soft on the 200 MHz scope and the clock signal is a pure sine now, since we can only capture the 2nd harmonic, but not the 3rd.

Compare this to the same clock and MOSI signals displayed at ten times the bandwidth of the SDS6204:

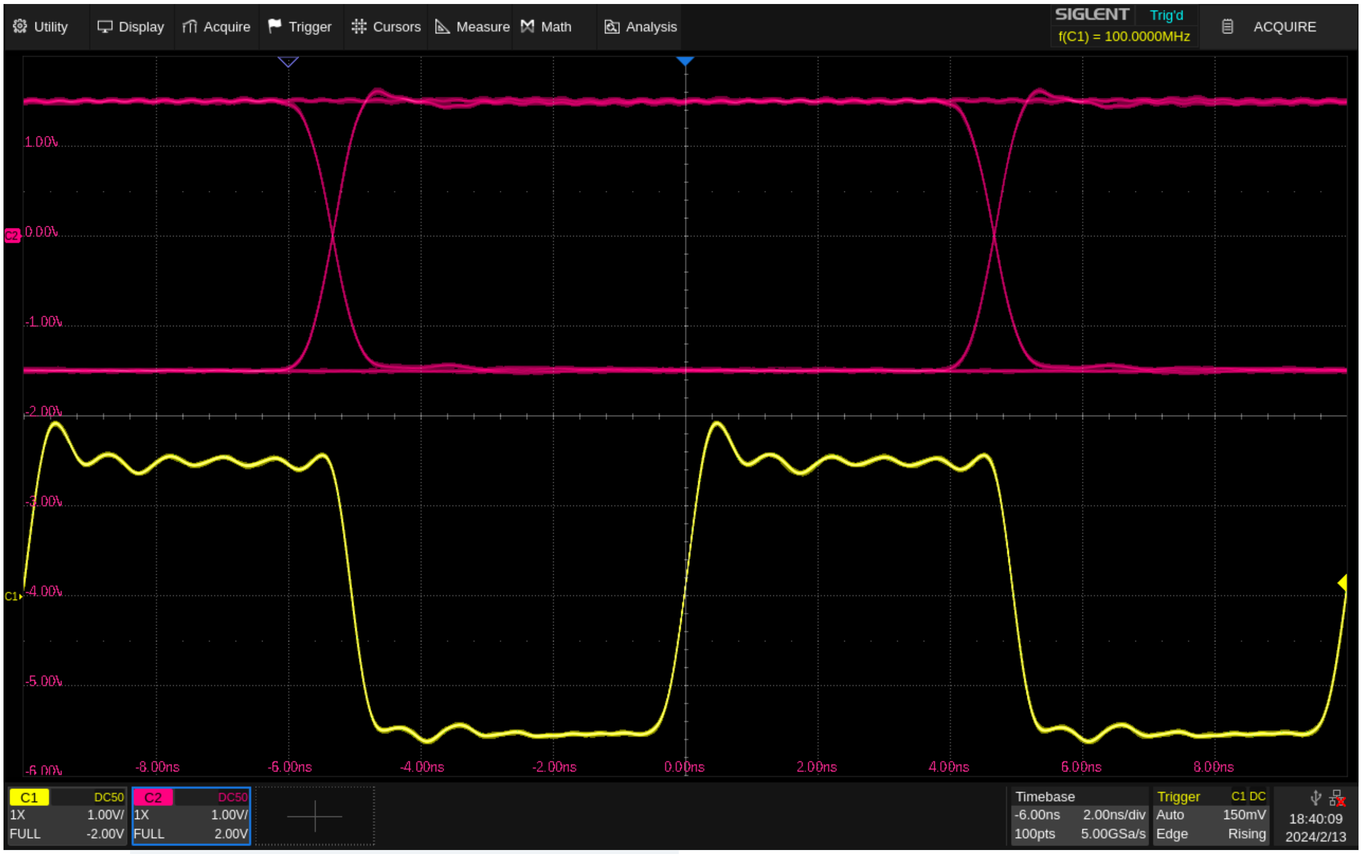

Fig. 123 SDS6204_Pro_H12_SPI_100Mbps_Eye

So, could we go up to even 200 Mbps maybe?

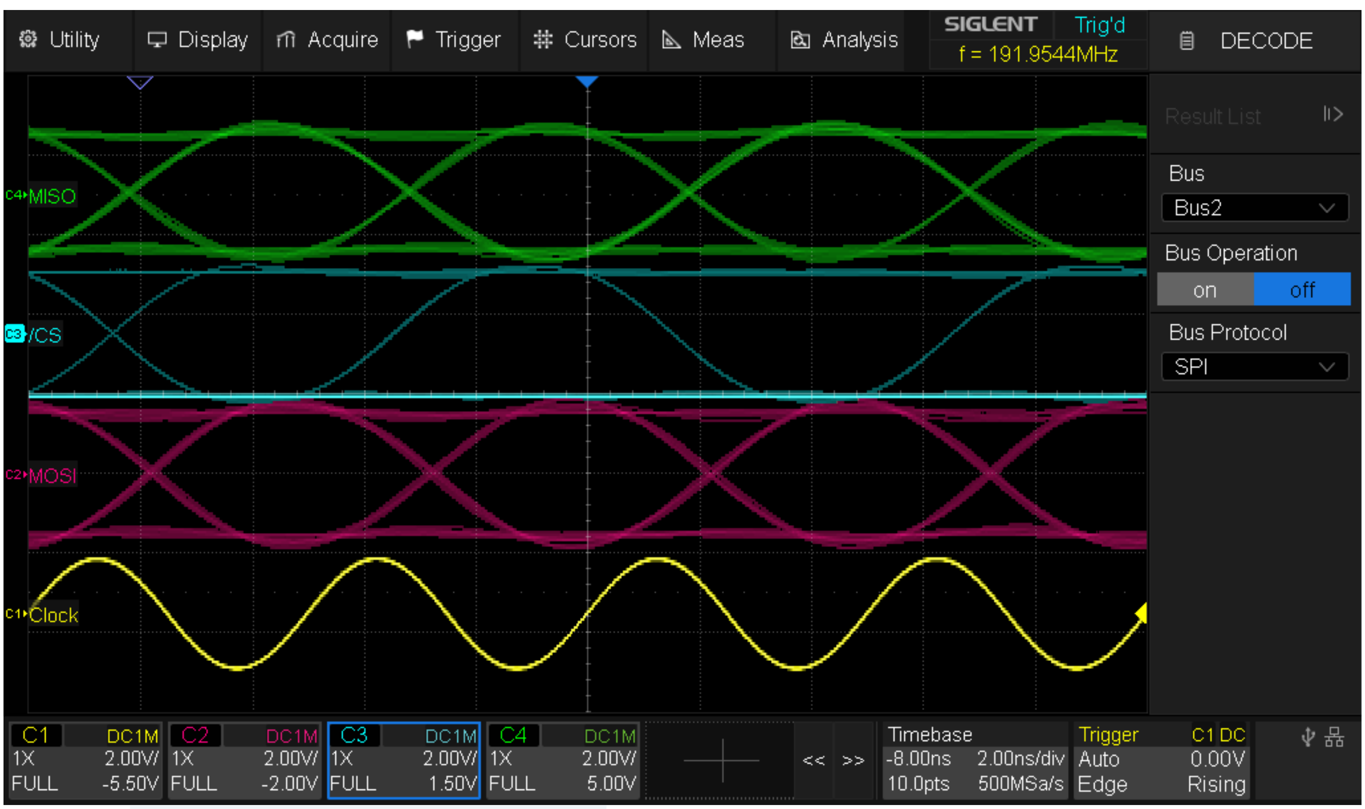

Fig. 124 SDS824X_HD_SPI_200Mbps_Eye

This is quite revealing. There might be a small chance to get it working with a fair share of luck, yet this certainly isn't a robust solution. The transitions are way too slow to give any reasonable error margin for the decoder.

The 200 Mbps SPI challenge

While we cannot decode a SPI data stream in 4-channel mode, it appears to be no problem if we stick to just two channels (I've switched to 32-bit words by now):

Fig. 125 SDS824X_HD_SPI_200Mbps_2Ch

Here's the obligate 0 and 1 test:

Fig. 126 SDS824X_HD_SPI_200Mbps_2Ch_0

Fig. 127 SDS824X_HD_SPI_200Mbps_2Ch_1

… and the eye diagram:

Fig. 128 SDS824X_HD_SPI_200Mbps_2Ch_Eye

The eye-diagram looks way better now, as the bandwidth is 245 MHz in this configuration and there is no digital filter. The sample rate is twice as high, and my guess is that it's the sample rate making all the difference.

Mask Test

Consistent with the previous SPI speed test, I want to demonstrate the usefulness of the full speed mask test, as it’s a perfect tool for automatic eye diagram monitoring.

Let’s get back to the 100 Mbps eye diagram of the previous test and set up a mask for it. For this, there is an integrated mask editor; we cannot just use the automatic mask creation here, because we don’t want to define a particular waveform with certain tolerances, but rather some forbidden area, so that the “eye” stays wide open.

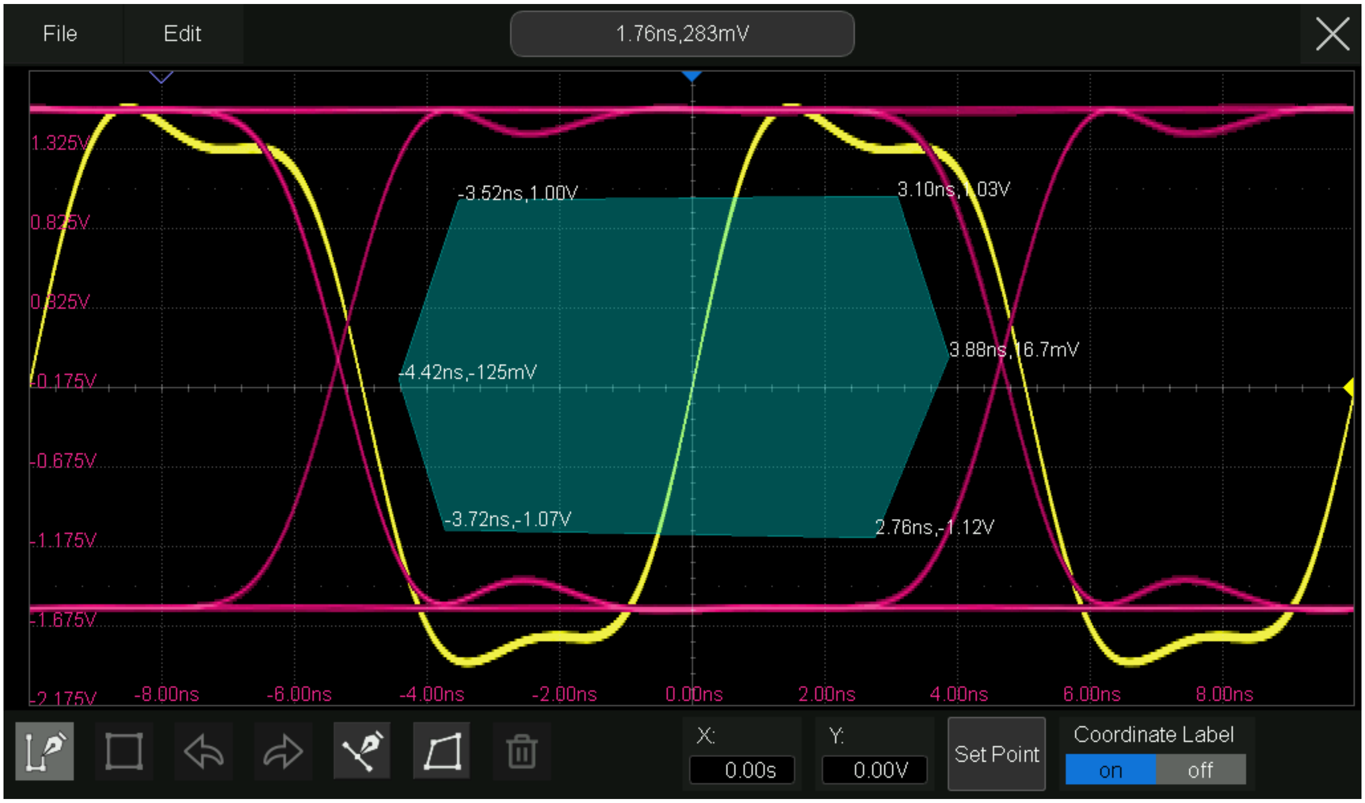

Fig. 129 SDS824X HD_Mask_Editor

I didn't spend much time creating a perfect mask; this one has just been cobbled together quite quickly. Furthermore, the eye most likely doesn't need to be that wide open in practical applications, yet this is just for demonstration's sake.

Now we let the mask test run. With the clean signals produced in a near ideal lab environment, we could have run that test for days without any failure, so I've added 224 mV_{RMS} noise with 500 MHz bandwidth to the data signal, hoping that I'll get a mask violation eventually:

Fig. 130 SDS824X HD_Mask_Test_run

The mask test is implemented in hardware, so we are getting a high number of passes within a short time. The screenshot has been taken a few seconds after the test started and up to this point, no mask violation has occurred. Yet it only takes a little while longer and a total of 411262 test runs until the first violation occurs.

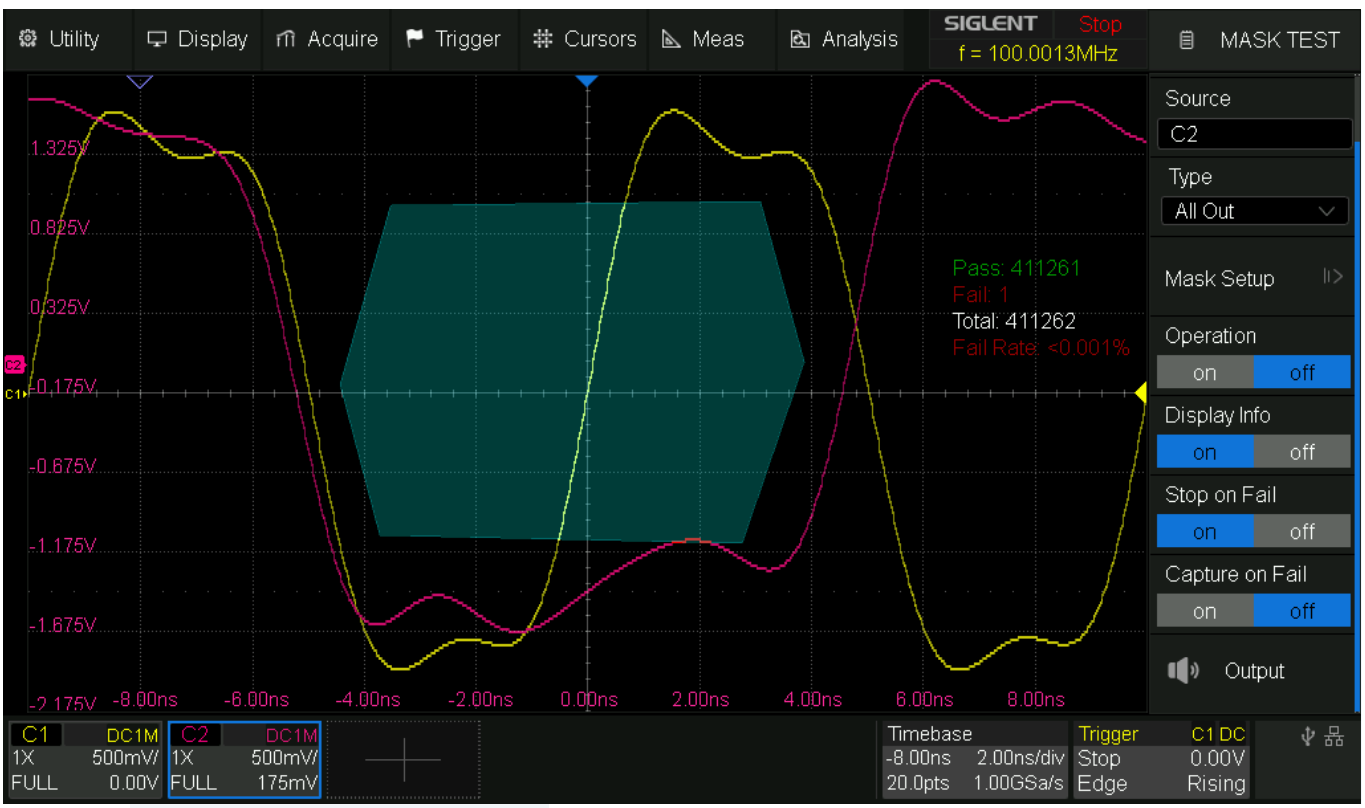

Fig. 131 SDS824X HD_Mask_Test_fail

As can be seen, the mask test was set up in a way that it would beep and stop on the first violation. Alternatively, we can also get an automatic screenshot every time the violation occurs.

This is another small difference to the SDS2000X HD: there we can also have the option "Failure to History", which means that all mask violations (and only these) are stored in the history, so we can have the mask test run overnight and then analyze all the mask violations in peace later.

The info block in the display tells us that the failure rate in this test scenario was less than 0.001%. Of course, it can be seen quite clearly that this violation wouldn't have prevented the SPI decoder to deliver correct results. It's all a matter of setting up the mask appropriately…

Probes

PP510

The frequency response plots in the “Bandwidth” section have been made with a properly terminated coax connection. A proper review should also test the associated probes – unfortunately, I don’t have one, as my test unit didn’t include any accessories. I suspect that the standard probes delivered with the SDS824X HD will be the well-known PP215. Even though I do have some very old PP215 (which probably aren’t quite the same as the current ones), I don’t have access to them right at the moment, hence make do with the only slightly younger 100 MHz PP510, just to give you an idea.

First the frequency response up to 500 MHz:

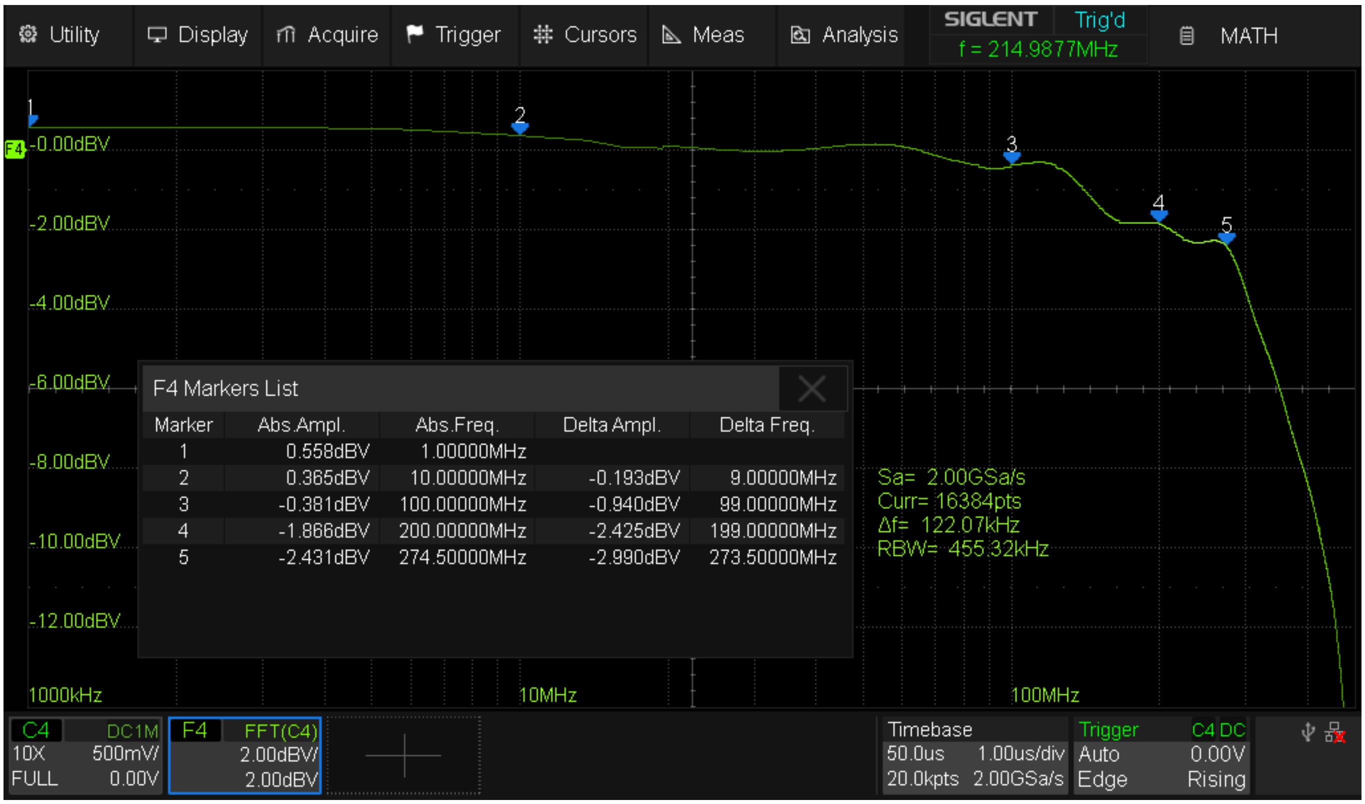

Fig. 132 SDS824X_HD_Probe_PP510_FR

It can be seen, that even Siglent’s cheapest 100 MHz probe extends the system bandwidth to ~274 MHz (244 MHz with direct coax connection). So much for the practical relevance of probe ratings and textbook formulas, which are way too simplistic as to actually model the real world.

Of course, the probe has been properly LF-compensated prior to the measurements:

Fig. 133 SDS824X_HD_Probe_PP510_PR_1kHz

The transition times are about 2 ns, which is slightly slower than with the direct coax connection. This is another occasion, where we can see the (ir)relevance of textbook formulas when it comes to real-world performance. The bandwidth was wider, yet the rise time is slower – how can that be?

It quite obviously is the frequency response, which is a far cry from the first order low pass, that is assumed in textbooks. The sudden drop of ~1.5 dB at about 120 MHz is most likely responsible for the slower rise time.

The ultimate test for proper HF-compensation is done with a fast (1 ns) risetime 1 MHz square wave.

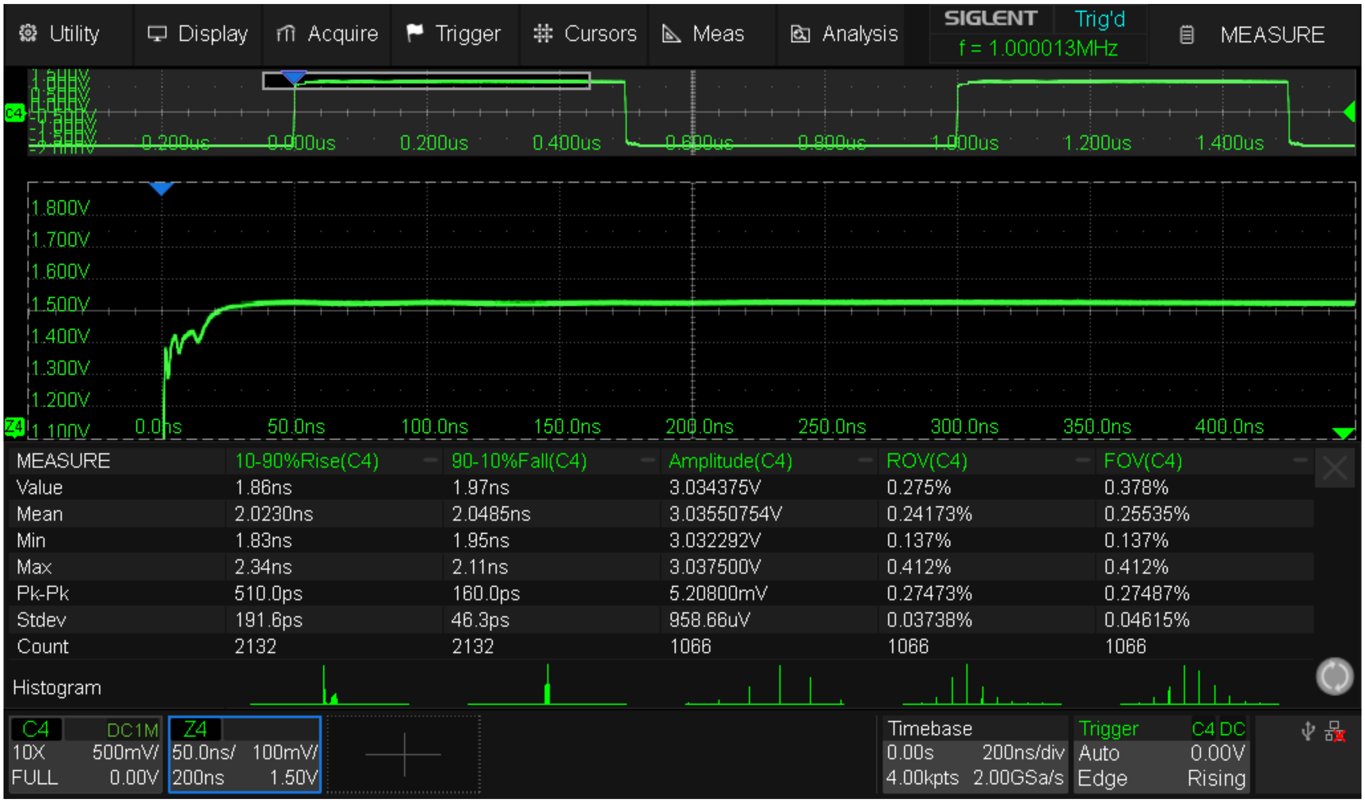

Fig. 134 SDS824X_HD_Probe_PP510_PR_1MHz_Zoom

It can be seen, that the PP510 isn't an ideal match for this scope because of an overdamped edge. In other words, the HF-compensation, which is not user adjustable, is not perfect for this probe-scope combination.

The screenshot above is also another demonstration how such details can be observed on a 12-bit DSO with proper zoom implementation, without the need to take a chance by overdriving the inputs.

Poor Men's Differential Probing

With analog scopes, we were able to combine two regular (single ended, ground referenced) channels into one differential channel. This was done by adding both channels with the 2nd channel inverted, whose gain had to be fine-tuned in order to get the maximum common mode rejection. Of course, this solution was far from ideal and sensitivity as well as common mode rejection were rather limited, especially at higher frequencies, which made it hard to get meaningful results when common mode voltages were high compared to the differential signal.

We can't do the same on a modern DSO, for a number of reasons:

- The fine adjust of the vertical gain has only ~2% resolution, so it is not suitable to balance the channels for a CMMR >34 dB.
- The difference has to be computed by a math channel, which always takes the vertical gain setting into account and scales to the true value, thus ignoring any gain adjustments.
- Finally, with only 8 bits the math result doesn't have enough resolution to properly analyze the differential signal. It is the same problem as with having vertical zoom on a 8-bit DSO.

The screenshot below demonstrates the result of two identical 10 MHz signals fed into channels 3 and 4 at 100 mV/div and a difference math channel is set to a 100 times higher gain at 1 mV/div:

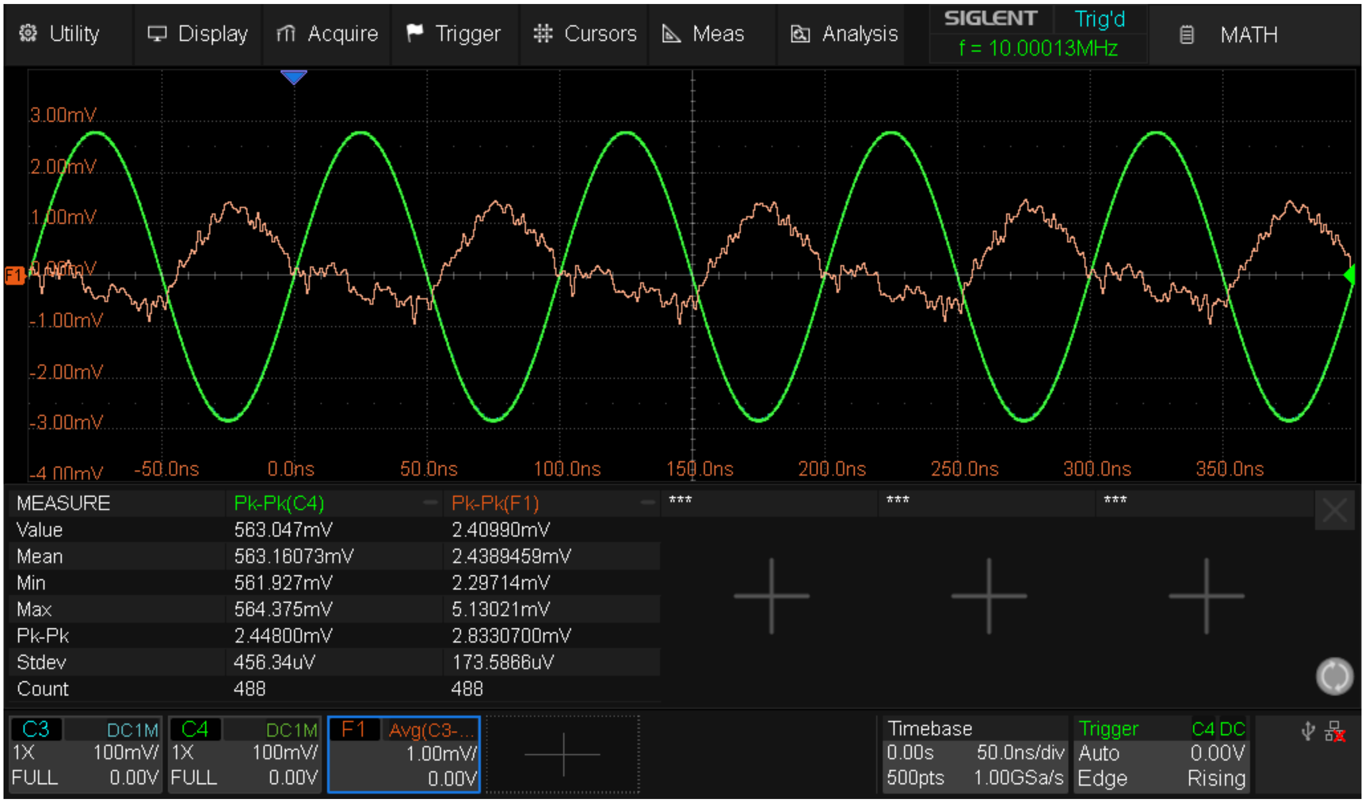

Fig. 135 SDS824X_HD_PMDiff_10MHz

Common mode rejection can be estimated from the amplitude measurements and would be 606.7/2.44 = 248.6 ~ 47.9dB, which is not bad at all – but we can do even better…

Of course, the balance is not perfect out of the box. Input channels and probes will both have gain tolerances, which compromise the common mode rejection. With a fairly precise instrument like the SDS800X HD we can just measure that:

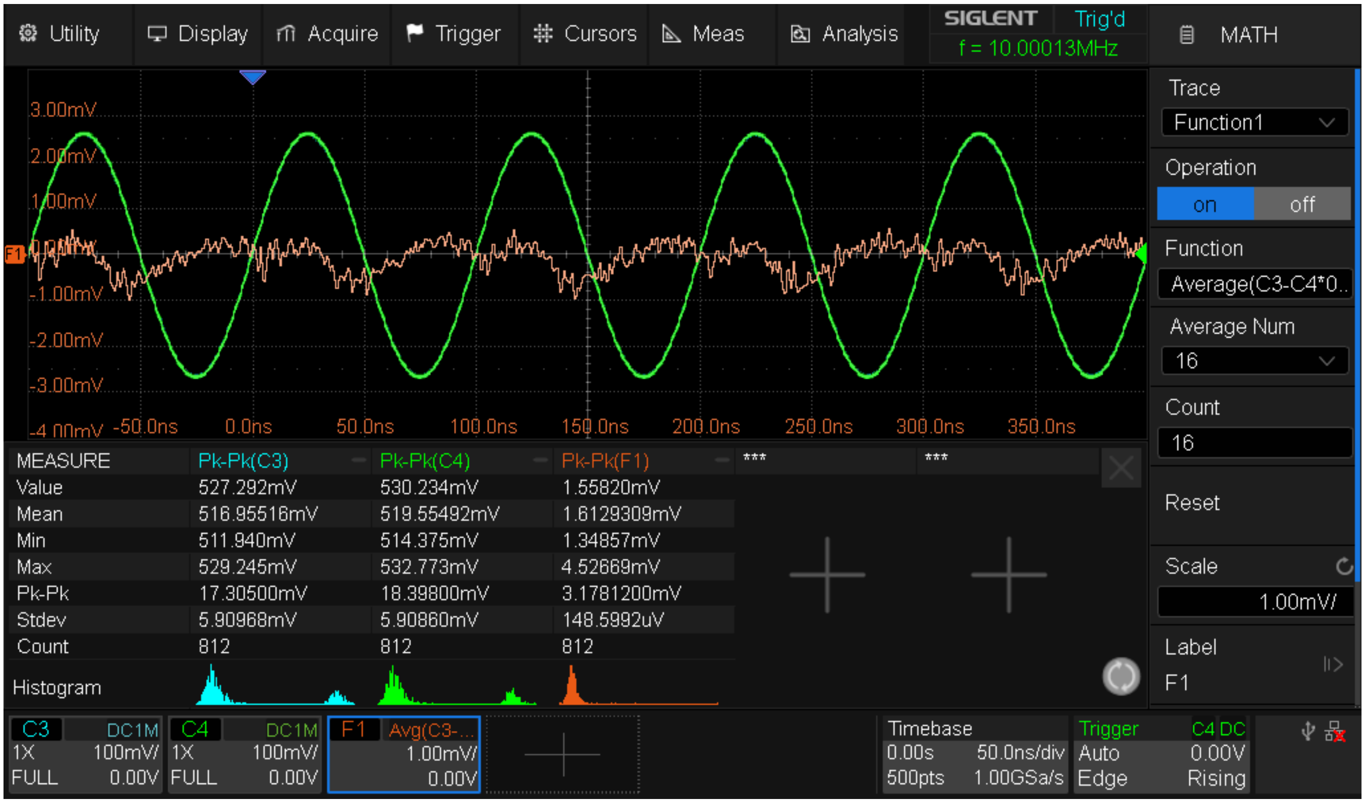

Fig. 136 SDS824X_HD_PMDiff_10MHz_corr1

As expected, there are slight differences. In this particular case, we get 516.955 mV_{PP} for Ch.3 and 519.555 mV_{PP} for Ch.4, so we can calculate the correction factor as 516.955/519.555 = 0.995. Consequently, we just replace C4 in the formula by the expression C4*0.995.

The common mode rejection can be estimated from the amplitude measurements and would be 519.555/1.613 = 322.1 ~ 50.16 dB, just 2.25 dB better than before. But the correction would certainly make a much more significant difference when the initial imbalance is more pronounced.

For best accuracy, (especially external) 50 ohms termination cannot be used at the scope input, as their tolerances could be up to 2%. Without termination, a direct coax connection of 1 meter length can work reasonably well up to a couple MHz, but at higher frequencies, amplitude accuracy and common mode rejection degrade considerably. Even at only 10 MHz, the 600 mV_{PP} signal was measured ~14 % low. The next screenshot demonstrates the same test at 100 MHz:

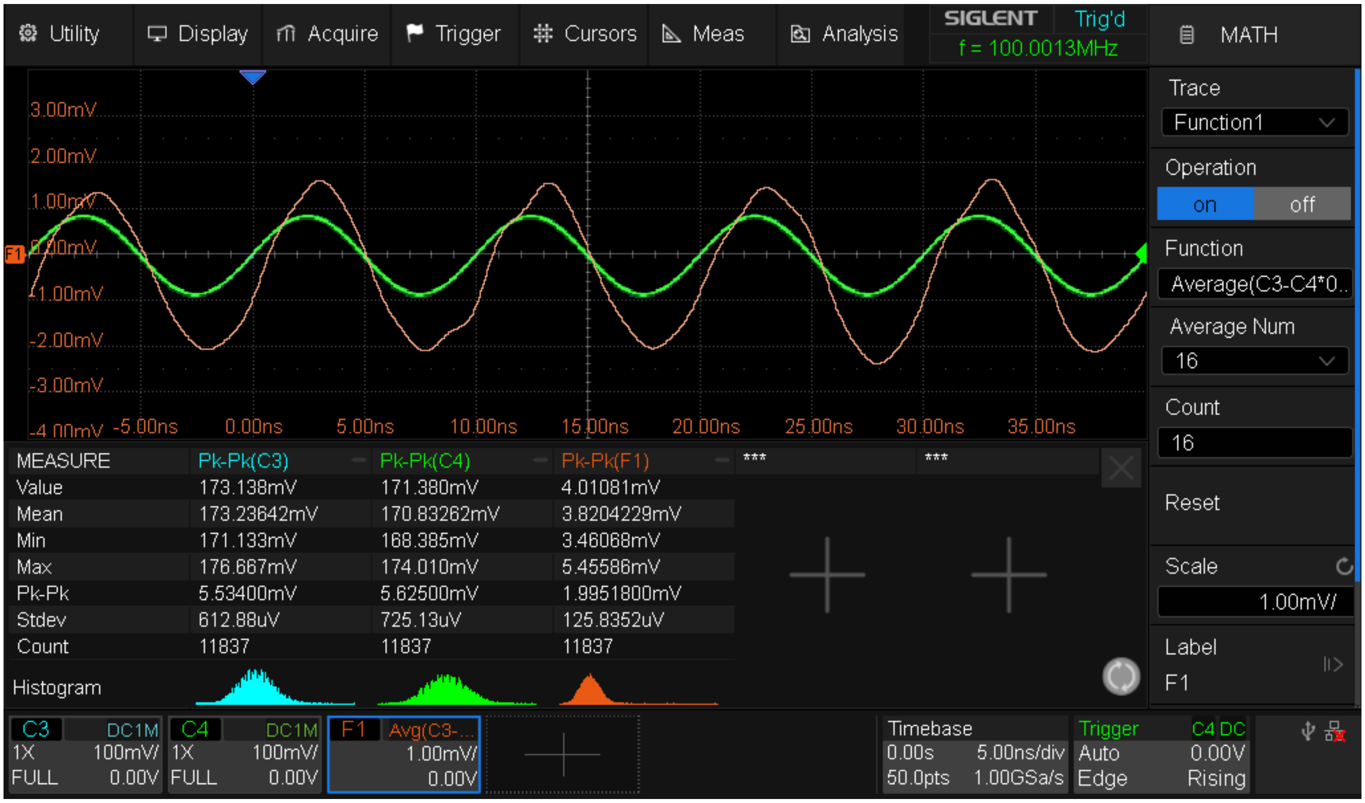

Fig. 137 SDS824X_HD_PMDiff_100MHz_corr1

Peak amplitude is totally off now and common mode rejection is degraded to 173.2/3.82 = 45.34 = 33.13dB, which could still be acceptable for some tasks, yet is clearly degraded compared with the 10 MHz test. Even more importantly, the amplitude ratio has significantly changed now. This means, that the correction factor is not valid over the entire DSO bandwidth.

Just for fun, we could try to alter the correction factor; now we get 173.236 mV$_{PP}$ for Ch.3 and 170.833 mV$_{PP}$ for Ch.4, so we can calculate the correction factor as 173.236/170.833 = 1.014.

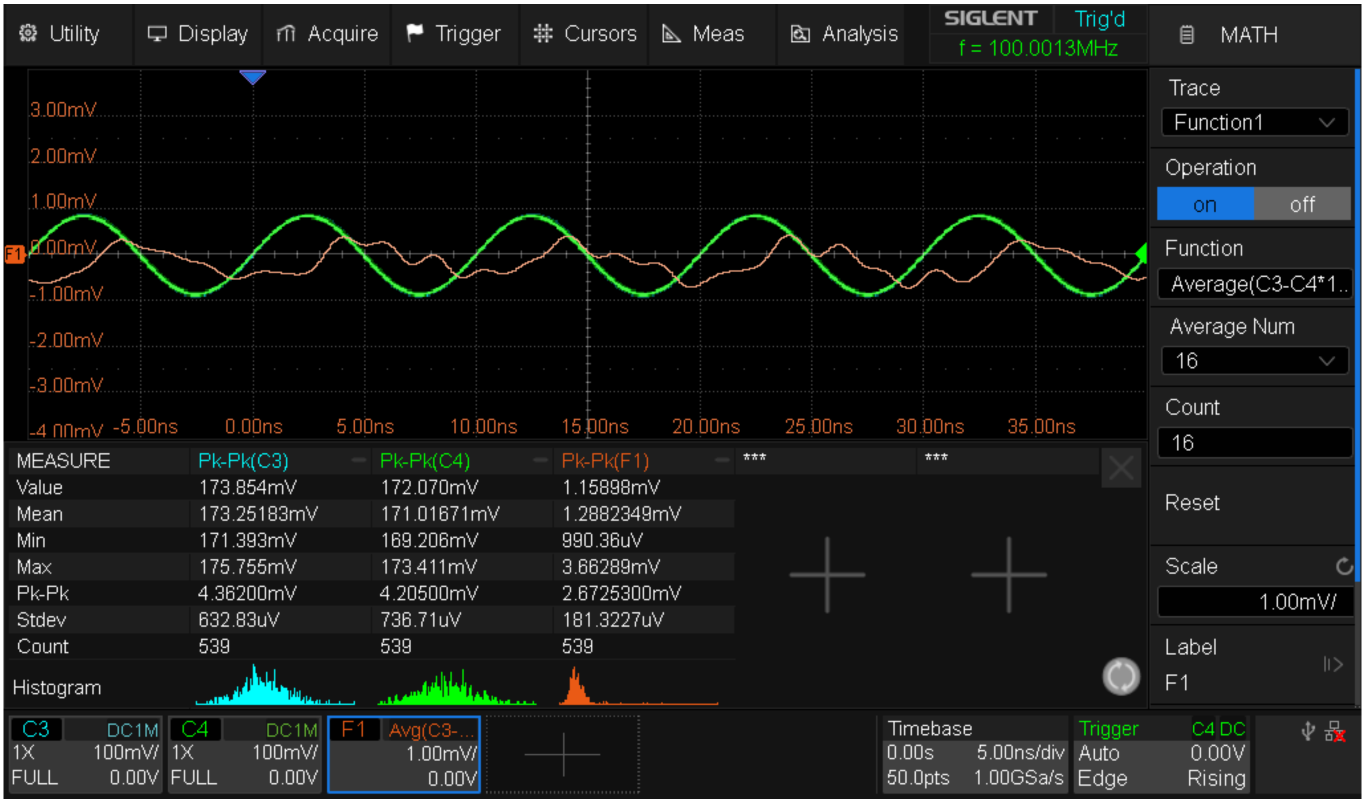

Fig. 138 SDS824X_HD_PMDiff_100MHz_corr2

Common mode rejection would now be respectable 173.25/1.288 =134.5 ≈ 42.5 dB.

In most practical scenarios, we'll use probes; this is problematic because of their complex impedance and transmission characteristics, so that the tolerances cannot be eliminated by applying a simple correction factor. I'll demonstrate the use of probes for a low frequency like 1 MHz. I only have a set of SP5050A probes here, yet I'm pretty confident my test results are still representative for any suitable 10x probe:

Fig. 139 SDS824X_HD_PMDiff_SP5050A_1MHz_corr

We get 3.0097 V_{PP} for Ch.3 and 3.0633 V_{PP} for Ch.4, so we can calculate the correction factor as 3.0097/3.0633 = 0.9825.

With this, common mode rejection is 3.0633/0.02541 = 120.55 ≈ 41.6 dB. This degrades quickly at higher frequencies.

As a conclusion, thanks to 12-bit resolution, 16-bit data processing and high accuracy of 0.5%, poor men's differential probing can be an option at low frequencies with this scope, whereas it didn't work at all with the older 8-bit SDS1000X-E series.

System Performance with SP5050A Probe

Many folks worry about the adequacy of the supplied probes.

There are few situations where it would be appropriate to use passive 10x high impedance probes at a test node within a circuit carrying >100 MHz signals. With a tip capacitance of 10 pF, the impedance at 100 MHz is just 160 Ω and forms a low-pass filter together with all not extremely low source resistances. This might still be okay for low impedance nodes like the outputs of line drivers, but certainly not anywhere else.

Apart from the fact, that at higher frequencies alternative probing solutions are required, the previous test of the good old 100 MHz PP510 has shown that they do not limit the system bandwidth, but even extend it to ~274 MHz and the rise time is quite adequate at 2 ns, see chapter "PP510".

If you wonder what you could gain with a much better (and much more expensive) probe – here's a test with Siglent's top model, the 500 MHz SP5050A.

Just to give you an idea, I've once tested the quite similar SP3050A, which is also rated 500 MHz, as always using the industry standard test with 25 Ω source impedance:

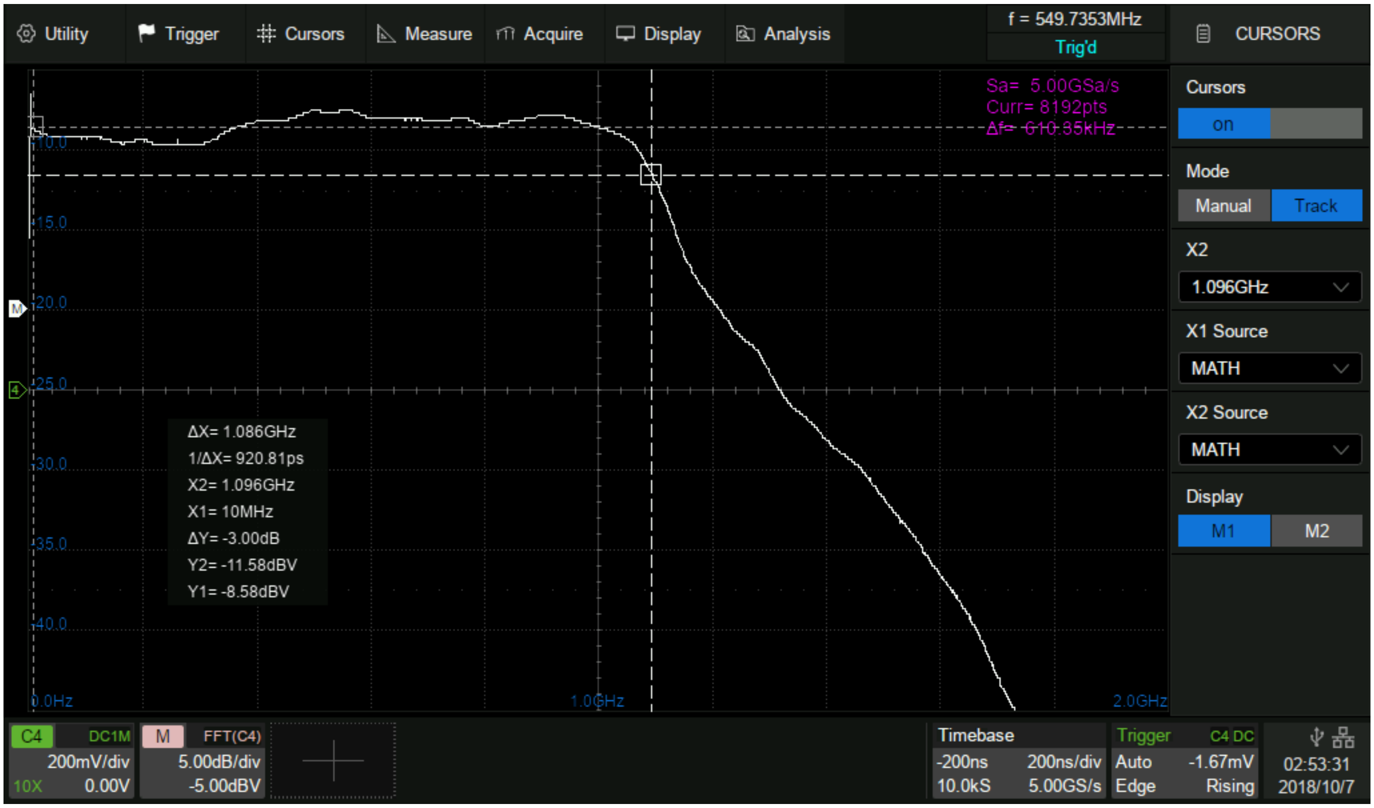

Fig. 140 SDS5104X_SP3050A_FFT_FR_2GHz_20mV_01

On a 1 GHz SDS5104X, the system bandwidth was 1.096 GHz with this "500 MHz" probe. Of course, this is useless in practice as it would only work on a terminated 50 Ω port with the supplied coax-adapter – and in that case we don't need the probe at all and would just use a direct coax connection instead.

Now here's the system frequency response with SP5050A up to 500 MHz. It can be seen, that this 500 MHz probe extends the system bandwidth to ~291 MHz (274 MHz with PP510 and 244 MHz with direct coax connection).

Fig. 141 SDS824X_HD_Probe_PP5050A_FR

Of course, the probe has been properly LF-compensated prior to the measurements:

Fig. 142 SDS824X_HD_Probe_PP5050A_PR_1kHz

The transition times are now in the 1.7-1.8 ns ballpark, hence very similar to the direct coax connection.

The ultimate test for proper HF-compensation is done with a fast (1 ns) risetime 1 MHz square wave.

Fig. 143 SDS824X_HD_Probe_PP5050A_PR_1MHz_Zoom

It can be seen, that the SP5050A is a pretty good match for this scope because the initial overshoot is about the same level as the top of the pulse.

Verdict: yes, the SP5050A outperforms a PP510 in just about every regard. It’s a very nice probe overall and it demonstrates what can be realistically achieved with the SDS824X HD. There still isn’t a huge difference after all.

Custom Probe Factors

This is a demonstration how to use custom probe factors for current measurement.

Consider we want to measure current using channel 4 and this should be set up for a rather weird conversion factor of 0.1234567 amperes per volt. First thing to do would be changing the Channel units from Volts to Amperes:

Fig. 144 SDS824X HD_Ch_Current_1V_A

Now we have set channel 4 to measure current at a conversion factor of 1 V/A (one volt per ampere), as it is displayed in the channel tab now. But in our example, we need a different conversion factor. Consequently, we enter the Probe menu:

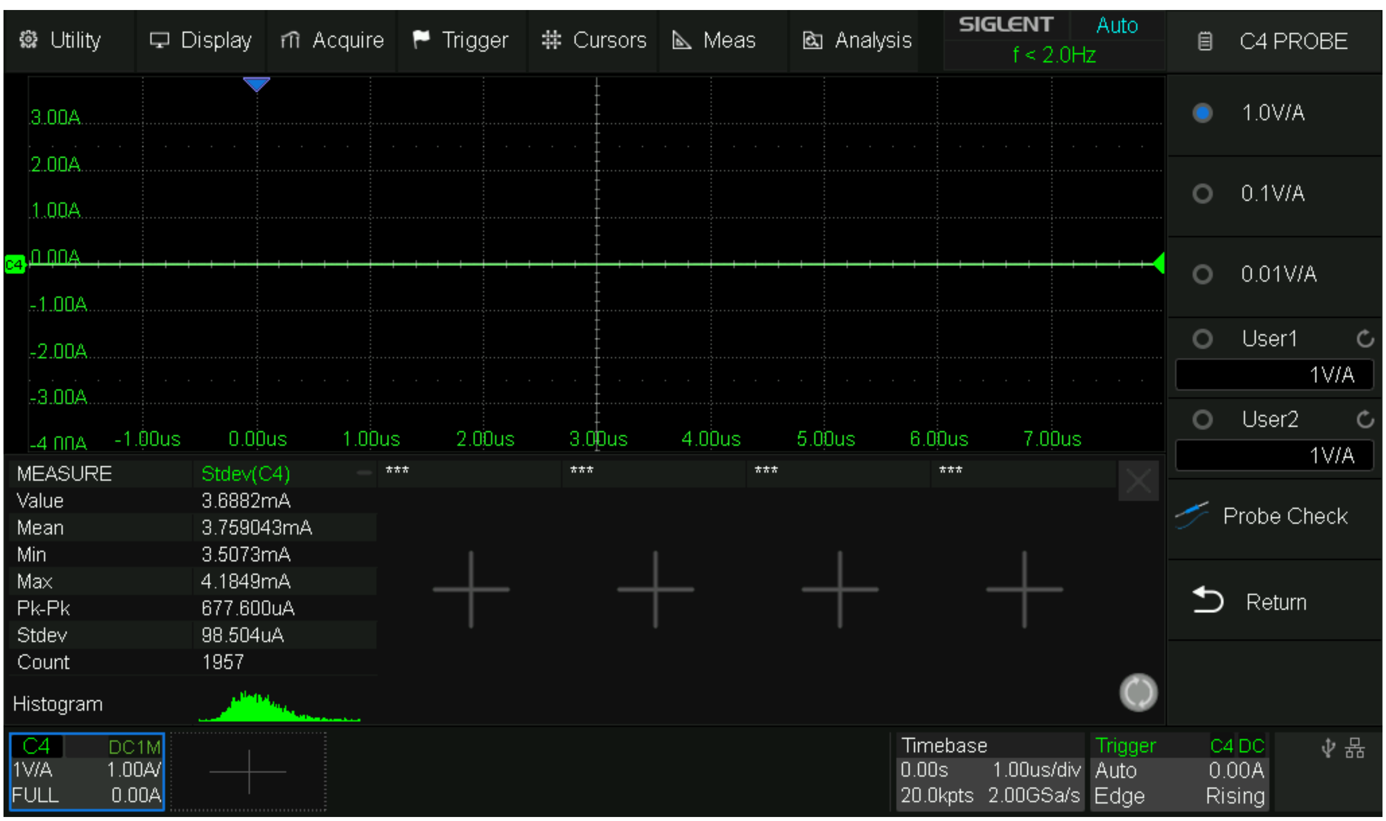

Fig. 145 SDS824X HD_Ch_Current_Probes

There is currently 1.0 V/A selected, and we can also get some more predefined probe factors, but not the one we want to use. Thankfully, there are also two permanent user settings. We can preset them to our most used custom probes and can use them just like the predefined ones from now on. Tapping on the user setting, we get the input keypad for the custom V/A setting:

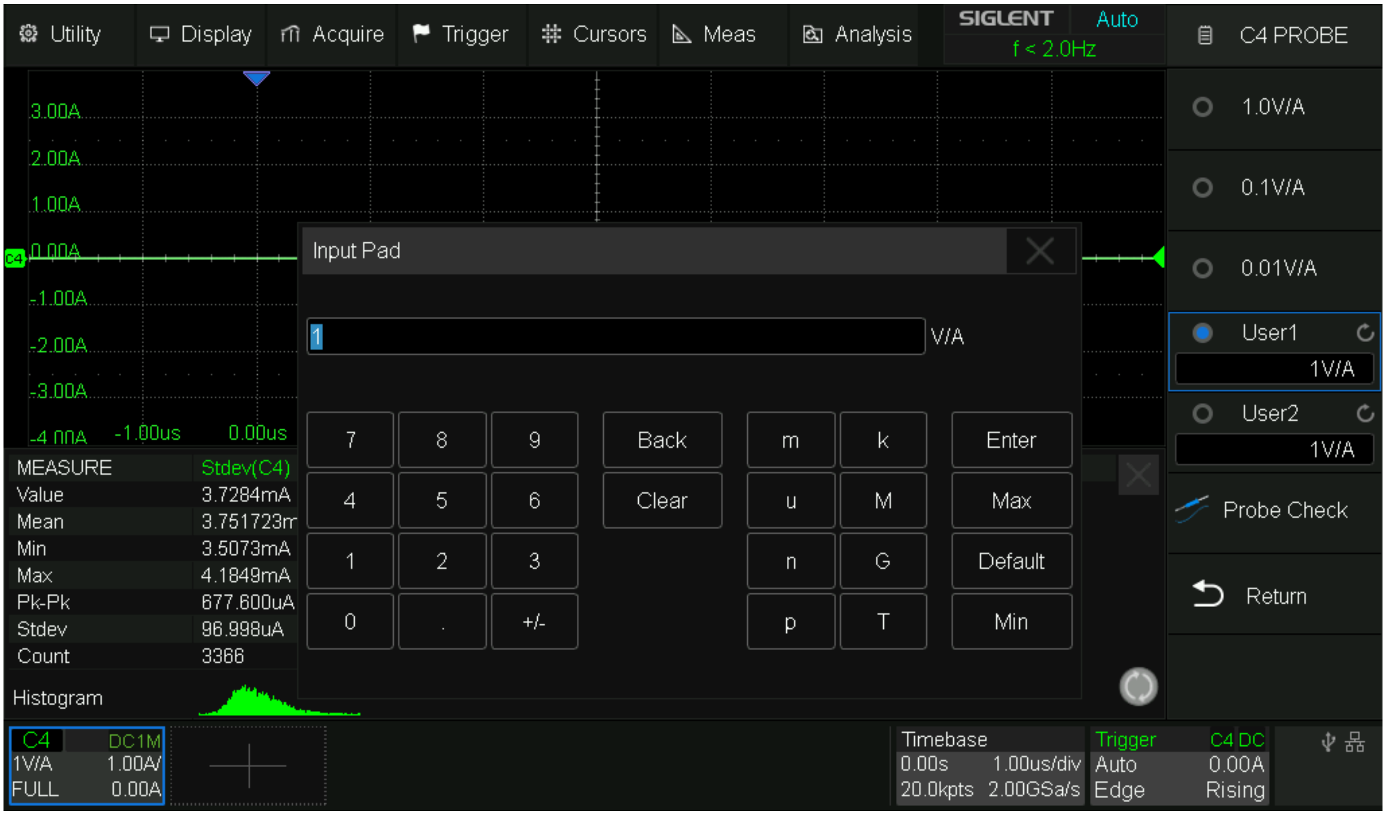

Fig. 146 SDS824X HD_Ch_Current_Probe_Input

Here we can enter any desired conversion factor from 1 µA/V up to 1 MA/V with at least 6 digits resolution. For example, here is 0.1234567 V/A:

Fig. 147 SDS824X HD_Ch_Current_Probe_0.1234567

Here's a measurement example: channel 1 is set to 1x voltage probe at 500 mV/div, whereas channel 4 is set to a custom probe at 4.05 A/div.

Fig. 148 SDS824X HD_Ch_0.1234567_Power

During normal use, channel 4 would be set to 500 mV/div with a 1x voltage probe. The custom probe factor of 0.1234567 V/A is equivalent to 8.1000059 A/V. This multiplied by 500 mV/div results in 4.050003 A/div, just as it is displayed in the corresponding channel tab

Just for fun, I’ve set up a math operation, simply multiplying the two channels to get the power. Trace F1 shows a scaling of 8.1 W/div and a vertical offset of 0.0 W.

The measurements show the standard deviation (=AC-RMS) for channels 1 in volts and for channel 4 in amperes. Formula trace F1 shows the AC-RMS power in watts.