EE 465 Filter Test Circuit/Project

Created by Rory Gowen

March 12, 2021

Introduction

This document describes the setup of a Verilog Project that can be used to measure the magnitude/frequency response of a filter. The project has been geared towards measuring a Square-root Raised Cosine filter response to a random signal.

Setting Up a Project to Test the Response of a Filter

Create a Verilog Project for the DE2-115 (EP4C115F29C7) and include the following verilog files:

- filter_design.v
- filter_design.sdc
- EE465_filter_test.qxp
- stp1.stp

Set the `filter_design.v` file to be the top level module. Connect the filter to be tested to the `EE465_filter_test` module in the `filter_design` module.

EE465 Filter Test Module – (EE465_filter_test.qxp)

The `EE465_filter_test` module provides the following functions for testing a filter:

- Generates the following clock outputs:
  - system_clk: clock_50 divided by 2
  - sample_clk_ena: pulse of duration (1/system_clk), frequency of clock_50 divided by 8
  - symbol_clk_ena: pulse of duration (1/system_clk), frequency of clock_50 divided by 32
- Generates a 2-bit pseudo-random sequence based upon a 22-bit maximum length LFSR (`lfsr_value`)
- Generates a 1s17 signal containing a 4-ASK signal that has been upsampled by 4. (`input_to_filter_1s17`)
- Scales the amplitude of `input_to_filter_1s17` by a factor of $2^{-\texttt{filter\_input\_scale}}$.
- Translates the filter output up to a center frequency of $\frac{3277}{16383} system\_clk$ (approximately 5 MHz when clock_50 is 50 MHz)

- Upsamples the output of the filter being tested by a factor of 4 and anti-aliases the upsampled output of the filter.
- Provides a 14-bit value that can be connected directly to the DAC on the DE2-115 daughter card.

The ports list of the module is given in table 1 with a general description of the function of each connection in the port list.

It should be noted that not all filters to be tested will require all of the port connections. For timing considerations the clocks generated by the module should be used throughout the project containing the module.

A basic block diagram of the module is given in figure 1.

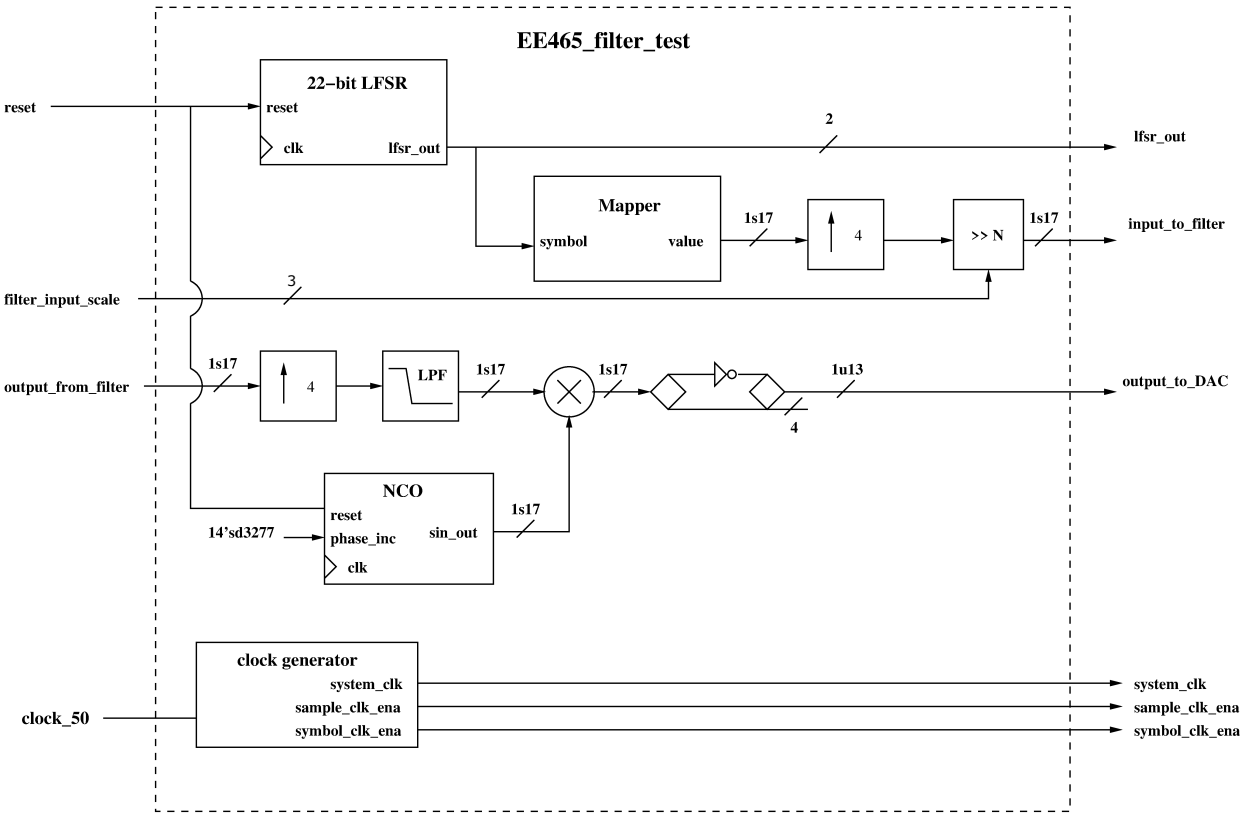
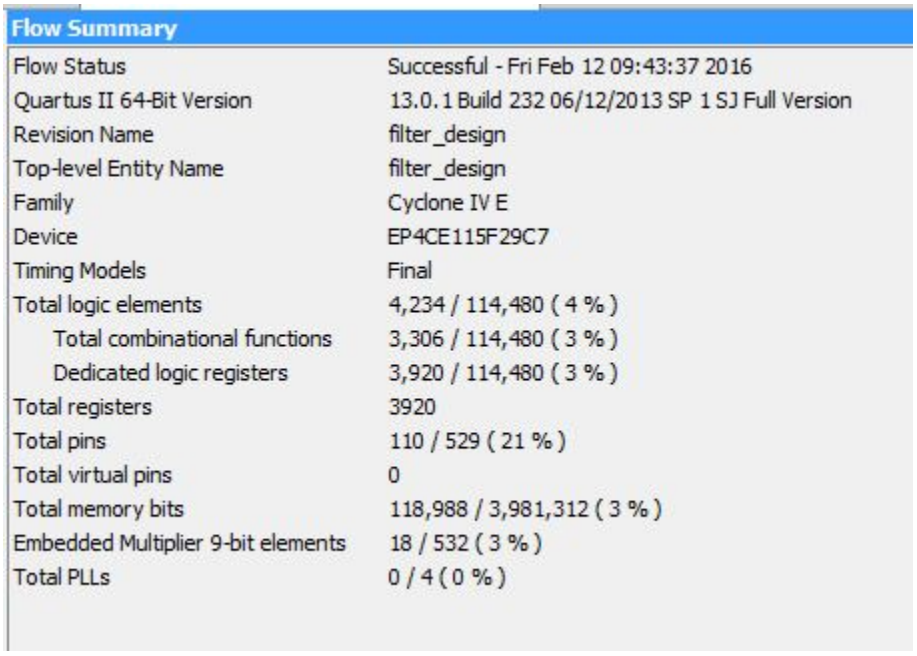


Figure 1: EE465_filter_test Block Diagram

The resource usage of the module (without a filter being tested) is shown in figure 2. The module uses 9 18x18-bit multipliers to implement the anti-aliasing and frequency translation of the filter, therefore when testing a filter with this module 9 multipliers may be subtracted from the resource usage to determine how many multipliers the filter being tested uses.

Table 1: Port list and Description for EE465_filter_test Module

Port Connection Name	**Direction**	**bit format**	**Description**
clock_50	input	clock	Provides a master clock to the module clock is expected to be 50 MHz.
reset	input	1-bit	Active high reset. Resets the module and starts the LFSR.
output_from_filter_1s17	input	1s17	The output of the filter should be connected to the module here.
filter_input_scale	input	3-bit	Scales the input signal to the filter (`input_to_filter_1s17`) by $2^{-\texttt{filter\_input\_scale}}$. This signal can be used to help debug overflow issues.
input_to_filter_1s17	output	1s17	A 4 times upsampled signed value that can be connected to the filter input. The upsampled signal is a full-scale and zero-stuffed.
lfsr_value	output	2-bit	Provides 2-bits from the LFSR that can be used for testing certain implementations of filters. This value is sampled at a frequency of system_clk Hz.
output_to_DAC	output	1u13	Provides a signal that can be directly connected to the DAC.
system_clk	output	clock	Provides a clock at a frequency of $\frac{clock\_50}{2}$
sample_clk_ena	output	clock	Provides a clock pulse of duration $\frac{1}{system\_clk}$, frequency of $\frac{clock\_50}{8}$
symbol_clk_ena	output	clock	Provides a clock pulse of duration $\frac{1}{system\_clk}$, frequency of $\frac{clock\_50}{32}$



Flow Summary	
Flow Status	Successful - Fri Feb 12 09:43:37 2016
Quartus II 64-Bit Version	13.0.1 Build 232 06/12/2013 SP 1 SJ Full Version
Revision Name	filter_design
Top-level Entity Name	filter_design
Family	Cyclone IV E
Device	EP4CE115F29C7
Timing Models	Final
Total logic elements	4,234 / 114,480 (4 %)
Total combinational functions	3,306 / 114,480 (3 %)
Dedicated logic registers	3,920 / 114,480 (3 %)
Total registers	3920
Total pins	110 / 529 (21 %)
Total virtual pins	0
Total memory bits	118,988 / 3,981,312 (3 %)
Embedded Multiplier 9-bit elements	18 / 532 (3 %)
Total PLLs	0 / 4 (0 %)

Figure 2: EE465_filter_test Resource Usage

Adding You Filter to the Project

The filter to be tested should be instantiated in place of the **ee465_gold_standard_srrc** module in `filter_design.v`. Your particular design may not require the same signals as the **ee465_gold_standard_srrc** module. You will have to change connections as appropriate for your filter.

1 Testing Your Filter

In order to test the filter to determine out of band emissions a Signal Analyzer will be used.

In order to get the best measurement results from the signal analyzer the filter being tested should have the maximum output possible without overflow errors occurring. This allows for the largest dynamic measurement range of the signal analyzer when measuring your filter characteristics. *You may want to adjust your filter coefficients to achieve the maximum output gain possible*

Using Signal Tap can help determine if the filter being tested has any overflow errors. An example of a filter **without** overflows is shown in figure 3. A filter with overflow errors should have abrupt changes in the `output_from_filter_1s17` signal and will not be smooth.

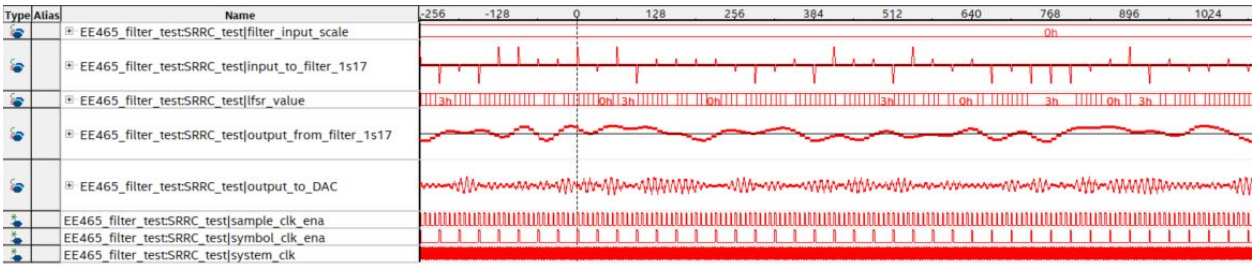


Figure 3: Example of filter without overflow error

Once the filter to be tested appears to be working correctly with a maximum gain the signal analyzer can be setup to make out of band emission measurements as will be discussed below.

Making Out of Band Emission Tests with a Signal Analyzer

Ensure that the FPGA Daughter board DAC B is connected to the Signal Analyzer.

Follow the proceedure given in the **EE465-2016 VSA Tutorial** guide in section titled "Measuring Power in a Frequency Range" to setup an ACP measuremnt on the signal analyzer. Use the following parameters when setting the carrier and offsets:

- Carrier Reference Frequency: 5 MHz
- Offset A (OB1):
  - Offset Freq: 985 kHz
  - Integrated BW: 220 kHz
  - Offset Side: both
- Offset B (OB2):
  - Offset Freq: 1.86 MHz
  - Integrated BW: 1.53 MHz
  - Offset Side: both
- Offset C (OB3):
  - Offset Freq: 3.5 MHz
  - Integrated BW: 1.75 MHz
  - Offset Side: Pos

The spectrum analyzer should show results similar to those seen in figure 4.

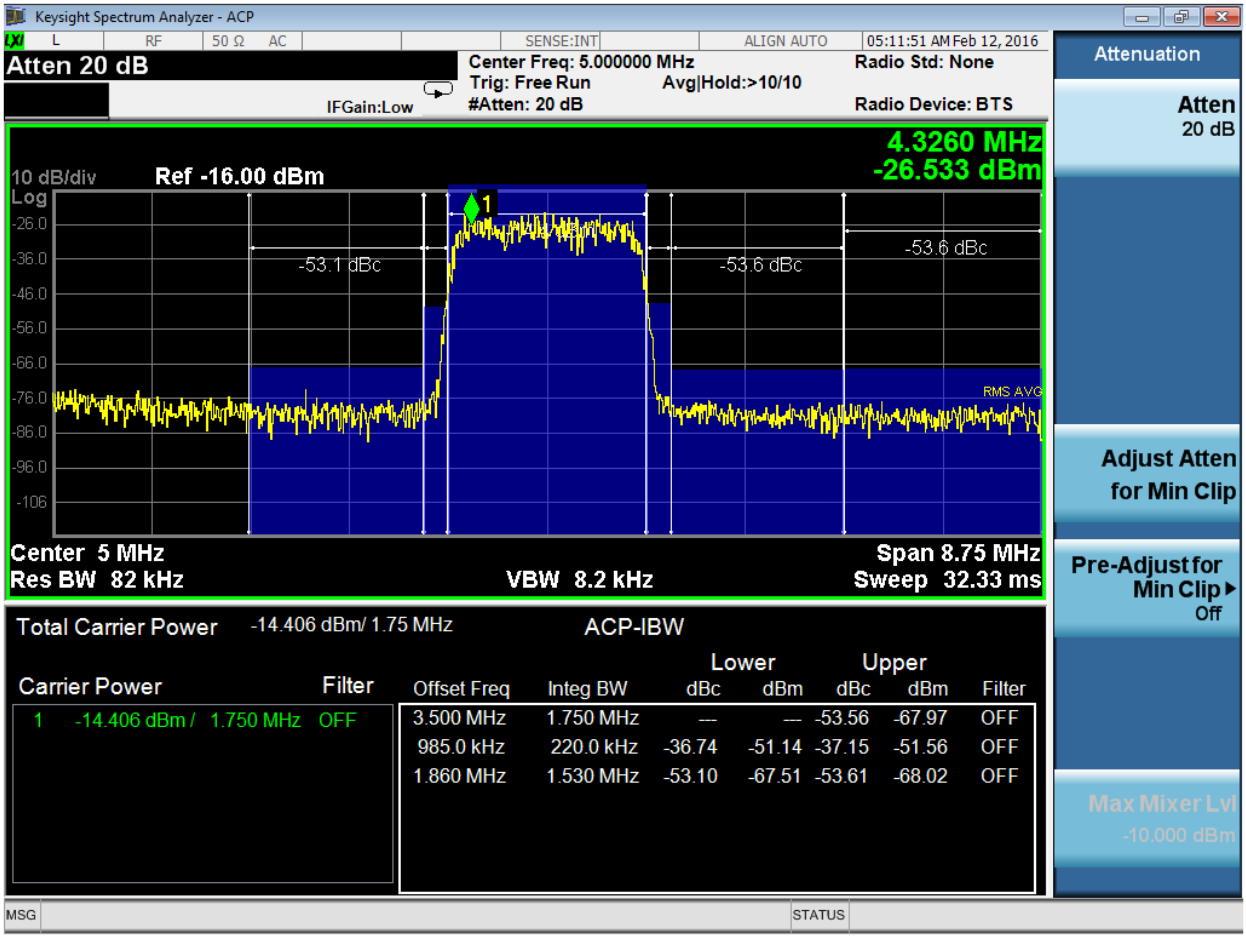


Figure 4: Initial ACP measurement setup

In order to achieve a better measurement the dynamic range of the signal analyzer must be improved.

Adjust the resolution bandwidth of the signal analyzer to be 10 kHz and notice the improvement. *Ensure that the ADC is not overloaded. If the Signal Analyzer indicates an ADC over-range then adjust the input attenuation accordingly*

The goal here is to reduce the resolution bandwidth and input attenuation as much as possible without overloading the signal analyzer input.

Set the resolution bandwidth to be 2 kHz and wait for an entire sweep to occur (about one minute). The results seen should be improved.

The performance of the signal analyzer measurement can be further improved by reducing the input attenuation. It should be possible to reduce the input attenuation to 10 dB with a resolution bandwidth of 2 kHz. (see figure 5)

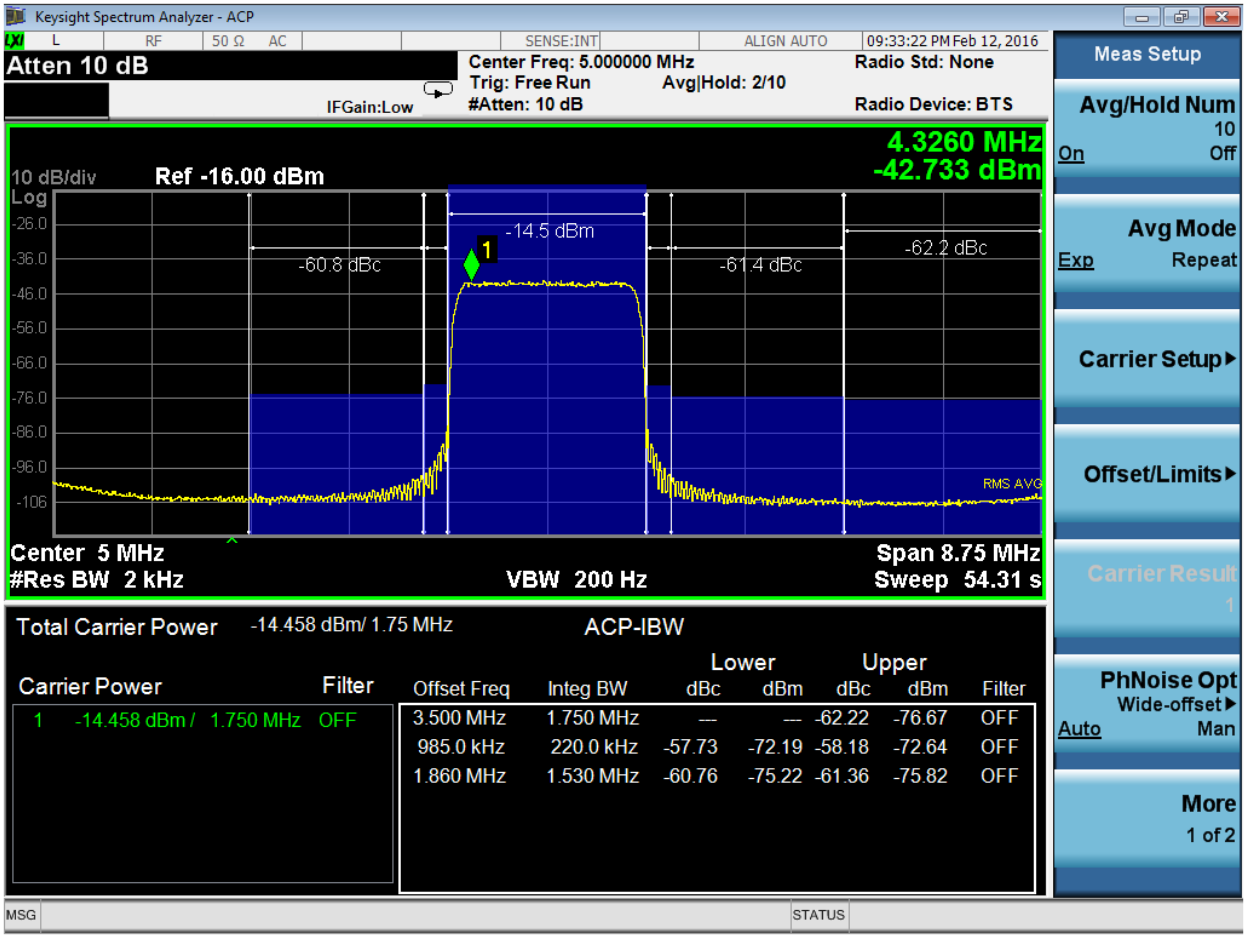


Figure 5: ACP measurement with reduced RBW

At this point you should be able to achieve measurements that will meet the specifications for the filter design requested in EE 465. You may find however that the filter is just barely meeting the required specifications.

It is possible to further improve the measurement on the signal analyzer by compensating for the noise inherent to the signal analyzer. To do this press the `MEAS SETUP` button and select `MORE` and turn on the `Noise Correction` feature. Wait for another complete sweep to occur. The results should now be greatly improved as can be seen in figure 6.

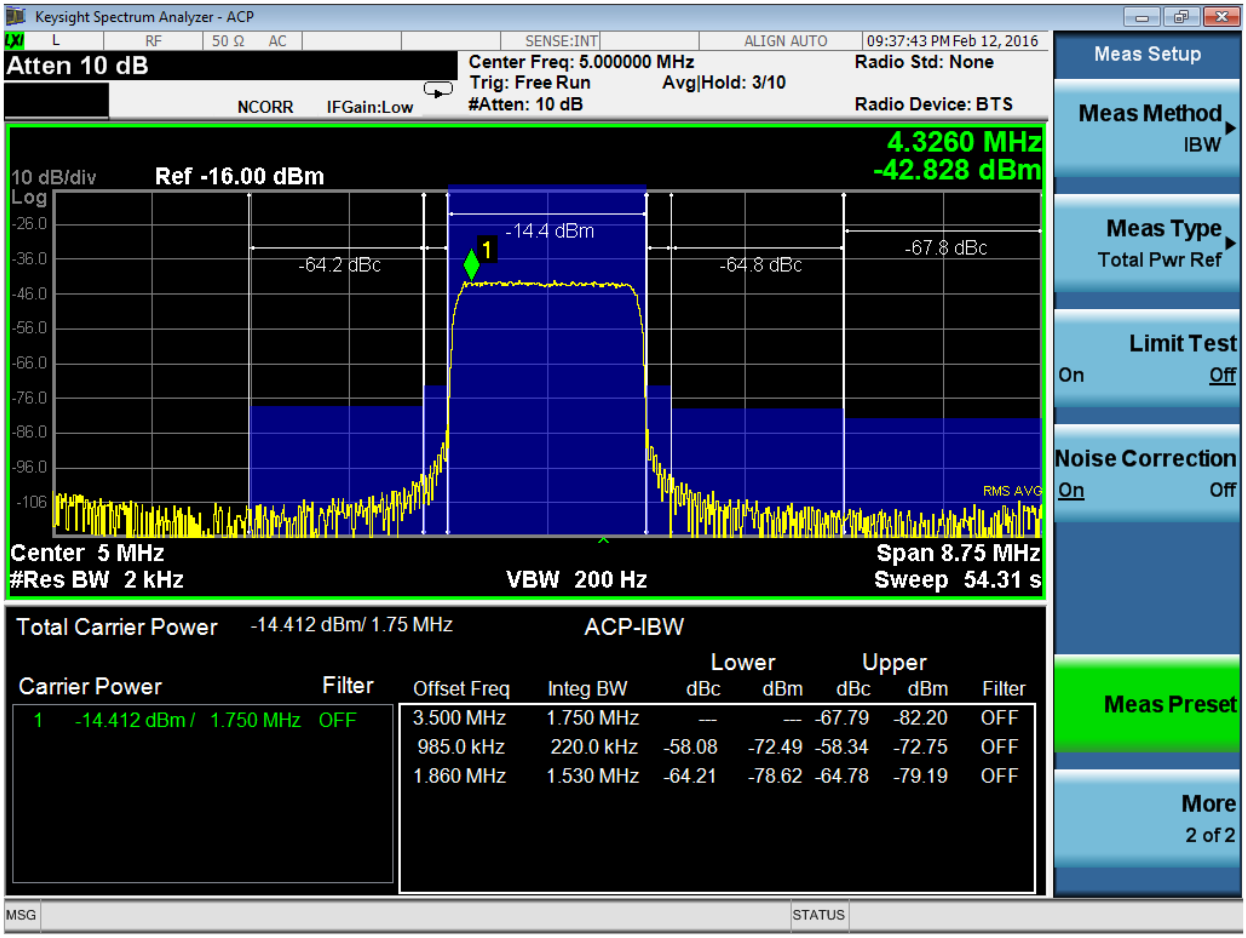


Figure 6: ACP measurement with noise correction

At this point it should be possible to measure filters with specifications better than the ones requested in EE 465. This is assuming, however, that the filter is designed for maximum gain for the given inputs from the pseudo-random generator in the EE465_filter_test module.