Signal chain LTspice simulation guide

Precision single channel voltage, current and biosignal measurement

Introduction

This guide offers a walkthrough on simulating a precision low power signal chain optimized for noise, using the accompanying LTspice schematic (.asc file). AC, transient and noise simulation examples are shown. A basic level of familiarity with LTspice is assumed, those new to the tool might want to go over a primer such as [1] first and make sure to check the attached readme.txt file. For support, you can post your questions in the LTspice forum in ADI EngineerZone [2]. You will need to create an account and log in to post questions.

Signal chain presentation

A block diagram of the signal chain can be seen in Figure 1.

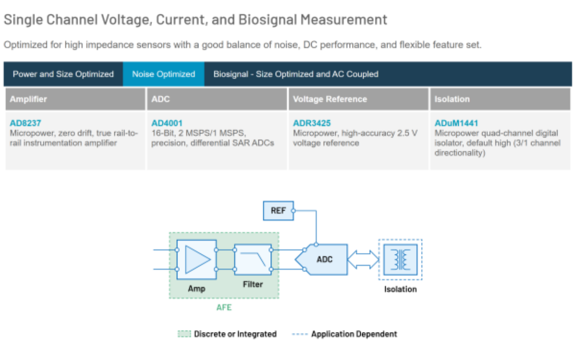

Figure 1.- Signal chain block diagram

Additional information in relation to this signal chain is available in Analog Device's website [3].

At the heart of this signal chain we have the AD4001, a 16-bit, differential, SAR ADC. We will be making use of its high-z mode, which simplifies driving the part and leads to a simpler, smaller and less power intensive signal chain. The ADC itself displays very low power consumption itself, which along with its low noise spectral density makes the part ideal for biosignal measurement applications.

An ADR3425 will be feeding a reference voltage of 2.5 V to the ADC. The key specs for this component are a temperature drift of 8 ppm/ºC (for B-class), low quiescent and shutdown current, and a low output voltage noise of 8 μV_{rms}.

The analog front end (AFE) of the signal chain is formed by an AD8237, a programmable gain instrumentation amplifier (PGIA), whose gain can be controlled via two external resistors.

As for the filter, in this case a simple RC network can be enough for many applications. For a more detailed look at the topic of filtering, see this reference [4].

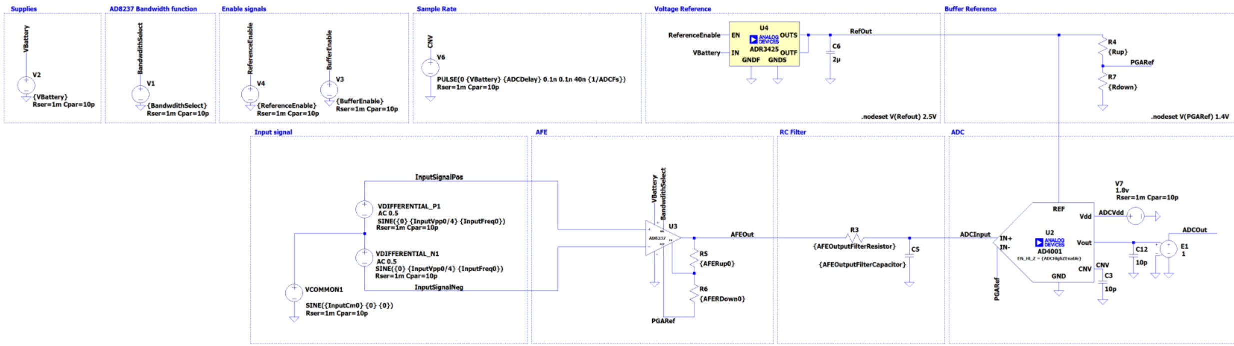

Figure 2.- Signal chain LTspice schematic

Simulation file usage

The accompanying LTspice simulation file is divided into boxed sections for readability. Some of these sections are:

Simulation commands

LTspice simulation commands for three kinds of simulation are located in this block, namely, transient, AC and noise. LTspice requires that only one simulation command exists in the schematic, so the most convenient use of this block is to comment out all commands except for the one we intend to run. Turning a simulation command into a comment can be quickly done by right clicking on the text box containing it, pressing escape to dismiss the simulation dialog, and then, in the newly appearing one, choosing “Comment” on the “How to netlist this text” control. The reverse operation is quicker, as right clicking on a comment will show the dialog where we can select “SPICE directive” as the way we want to netlist the text.

Parameters

Most variable elements in this signal chain can be controlled by modifying the parameters defined in this section. LTspice parameters are created through the .PARAM command. This allows quick modification of the signal chain, keeping track of its current status and it is convenient for running stepped simulations (if unfamiliar with .STEP command, see [5]).

Other signals

The generation of the “CNV” (convert) signal that triggers the ADC can be found here. This is a pulse train whose rising edges trigger the conversion process. See this reference [6] in relation to waveform generation in LTspice.

The “Enable signals” block generates the signals that would make it possible to power down the reference and buffer, if that was required by the system

Finally, the “AD8237 Bandwidth function” block generates the signal that toggles between high bandwidth mode and low bandwidth mode for this opamp.

Transient simulation

Transient simulation allows us to observe the signals in our circuit in the time domain; it is done through the .TRAN command.

Transient simulation example: verifying circuit gain of the signal chain

In this example, we will use a transient simulation to check the gain set for the amplifier stage.

We start out by commenting out all simulation commands except for the .TRAN one, after which our simulation command box should look like this:

Simulation Parameters

```
.tran 0 10m 1n 1u
.ac dec 100 1 1MEG
.noise V(Vout_noise) VDIFFERENTIAL_P dec 100 1 1MEG
```

Figure 3.- TRAN example: simulation commands

The input signal will be setup as follows:

- Differential Amplitude = 100 mV_{pp}
- Common mode = 1.25 V
- Frequency = 500 Hz sinusoidal

```
Input signal
.param InputVpp0 100mV --> Input signal amplitude (peak-to-peak) for voltage source
.param InputFreq0 500  --> Frequency for the sinusoidal differential input signal for voltage source
.param InputCm0 1.25V  --> Input signal common mode DC voltage for voltage source
```

Figure 4.- TRAN example: input signal configuration

The PGIA gain is determined by R5 and R6, and it is set to 10 in this example through the *AFEGain0* parameter. This should result in a 1 V_{pp} output. This peak to peak voltage will be superimposed on a mean value of 1.25 V, which is the level applied to the REF pin of the AD8237. That 1.25 V level is the mid value of the ADC reference voltage (Vref/2), it is created through a voltage divider (R4, R7).

With respect to the bandwidth pin of the AD8237, the datasheet [7] indicates:

> The AD8237 includes an RFI filter to remove high frequency out-of-band signals without affecting input impedance and CMRR over frequency. Additionally, there is a bandwidth mode pin to adjust the compensation. For gains greater than or equal to 10, the bandwidth mode pin (BW) can be tied to $+V_S$ to change the compensation and increase the gain bandwidth product of the amplifier to 1 MHz. Otherwise, connect BW to $–V_S$ for a 200 kHz gain bandwidth product.

As we are measuring input signals with frequencies around 1 kHz, the 200 kHz GBP will suffice so we can tie the BW pin to GND.

The default cut-off frequency for the filter is:

- AFE output filter = 7.5 kHz

And it is set with the following parameter values:

```
AFE Output filter (RC)
.param AFEOutputFilterCutoffFreq 7.5k   --> Desired cut-off frecuency (3dB) for AFE output filter
.param AFEOutputFilterResistor 1k   --> Resistor value for AFE output filter
.param AFEOutputFilterCapacitor {1/(2*PI*AFEOutputFilterResistor*AFEOutputFilterCutoffFreq)} --> AFE Ouput filter Capacitor value calculation
```

Figure 5.- TRAN example: other signal chain parameters

We are now ready to press the "Run" button to start the simulation.

Probing the input and output of the signal chain we will get the desired plot. As we have a differential input, we will click on the positive terminal and drag towards the negative one to plot their voltage difference (red signal input and blue output).

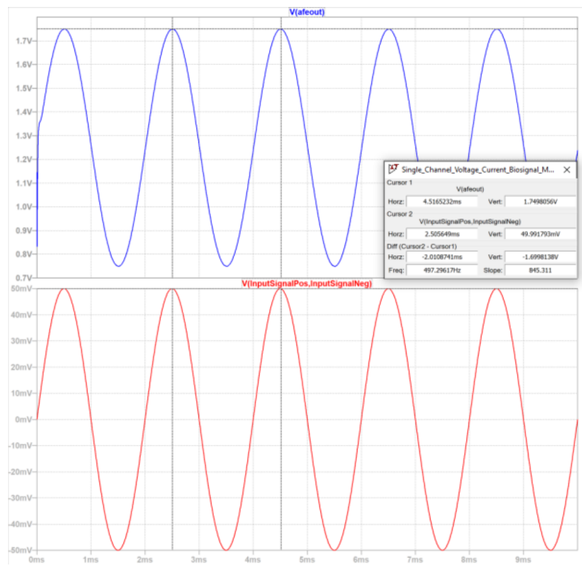

Figure 6.- TRAN: input and output signals on IN0

Observing Figure 6, we can confirm that the 100 mV_{pp} input signal has been amplified by a factor of 10 resulting in a 1 V_{pp} output riding on top of the 1.25 V DC level (See Figure 7).

Waveform: V(afeout)

Interval Start:	0s
Interval End:	9.9999ms
Average:	1.2502V
RMS:	1.2994V

Figure 7.- Average value of the output signal

AC simulation

For the AC analysis, the simulation calculates the circuit response over the frequency domain.

AC simulation example: gain stage and signal filter bandwidth

In this example, we use an AC simulation to check the resulting bandwidth of the amplifier stage and the signal filter stage.

Commence by enabling the AC simulation command only and configuring the signal chain as in the previous example:

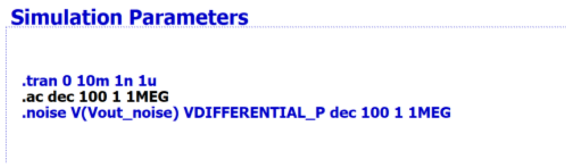

Figure 8.- AC simulation enabled

After running the simulation, we can probe the ADC input by clicking on the *adcinput* net.

The plot we get by default is the following:

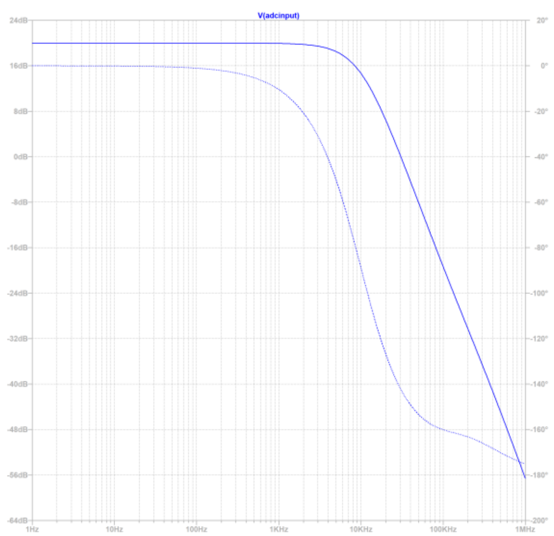

Figure 9.- AC example: ADC input and output default plot

Let's produce a more usable plot for our purpose by:

1) Right-clicking on the right y-axis, then clicking on the "Don't plot phase" button to remove the phase traces
2) Right-clicking on the left y-axis to adjust the range top (25dB), bottom (0dB) and tick values (1dB)
3) Clicking twice on the V(vout) trace name to place two cursors and drag them until we get a measure of the 3 dB bandwidth of the signal chain. The result is this plot:

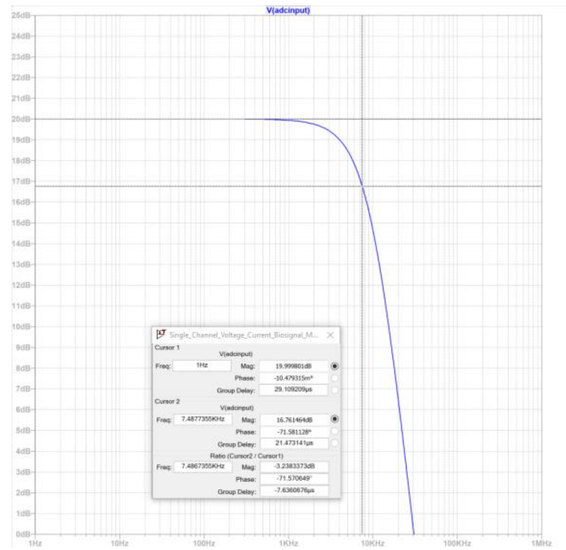

Figure 10.- AC example: ADC input and output formatted plot

Through the V(ADCinput) trace we observe the transfer function of programmable gain instrumentation amplifier and the RC filter, we confirm that in this case we set the 3 dB bandwidth around 7.5 kHz.

Furthermore, we can verify our gain 10 as well (20dB) which is the amplitude shown by the cursor sitting on the passband.

Noise simulation

A noise simulation performed through the .NOISE command allows to extract the noise spectral density at a specified (output) node in the circuit. It is concerned with random noise (thermal, flicker, shot) generated by the components that comprise the circuit, meaning it has nothing to do with other kinds of unwanted signals such as coupled interference, out of band signal components, crosstalk, power supply harmonics, etc. The .NOISE analysis is a particular case of small-signal AC analysis, and it is independent from .TRAN and .AC ones. That is, even though .NOISE analysis exposes noise voltages present in the circuit, those voltages cannot be observed in the time domain through a .TRAN simulation or have any effect on an .AC simulation.

For a first-order theoretical approach to signal chain noise analysis, see [8].

Noise simulation example: total signal chain noise

In this example, we will use the .NOISE simulation to check overall noise performance of the complete signal chain.

We start out by commenting out all simulation commands except for the .NOISE one:

Simulation Parameters

```
.tran 0 10m 100n 1u uic
.ac dec 100 1 1MEG
.noise V(ADCOut) VDIFFERENTIAL_P1 dec 100 1 1MEG
```

Figure 11.- NOISE simulation enabled

The .NOISE simulation command requires to specify what is the circuit node where we want to measure the noise (in this case the output of the ADC) as well as the input signal source.

The result we get in the waveform viewer after running the simulation, then left clicking on the output node is the following:

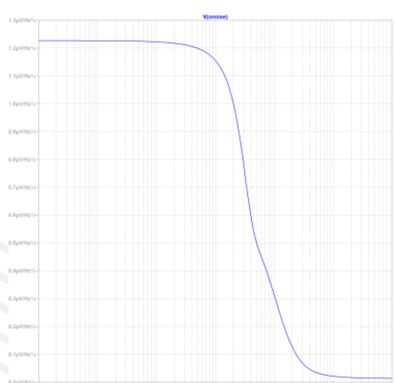

Figure 12.- .NOISE example: noise spectral density at signal chain output

This is the noise power spectral density at the output of our signal chain. The RMS value of the noise is calculated by integrating the power spectral density over the bandwidth of interest. We can have the waveform viewer perform this integration for us by pressing CTLR+left click on the trace label:

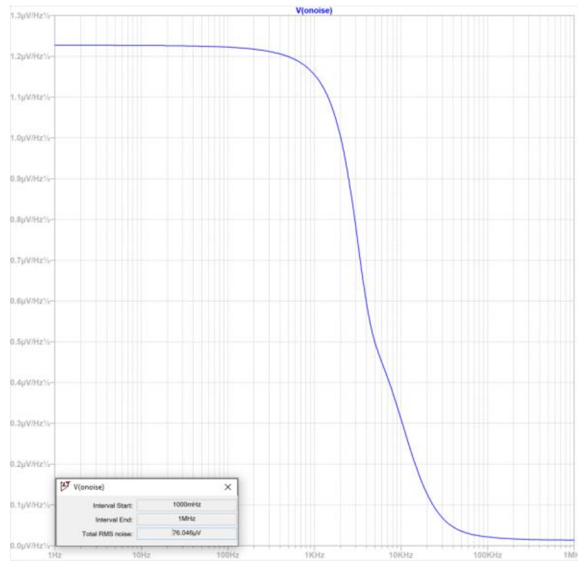

Figure 13.- .NOISE example: displaying integrated RMS noise

As we can see, according to the simulation the digital signal at the output of this signal chain will exhibit 76.046 µVrms of noise.

Note that the result is the integrated RMS noise of the **displayed** waveform. This means the result will change if we modify the frequency interval over which the analysis is performed (in the simulation command parameters), and it will also vary if we zoom-in on the x-axis.

Other considerations

Component models

LTspice models for all electronic components in this signal chain are included in the LTspice built-in library, so no external files are needed. Make sure your LTspice installation is up to date by clicking on "Sync Release" from the "Tools" menu.

Simulation models for components don't necessarily cover the full behavior of the real device. In general, it is safe to assume that the component will display the correct behavior under standard operating conditions (room temperature, nominal supply voltage…) while second order effects such as distortion, crosstalk, etc. might not be included in the models.

For ADCs, the models focus on analog behavior so most of the component behavior, pins, etc. related to digital I/O are not present in the model. The LTspice model will not be giving out the coded output over a digital interface (SPI, LVDS) as the actual device does. Instead, the ADC output is provided as a regular (analog) signal that is quantized and restricted to the maximum input span of the ADC. This makes it easy to seamlessly perform signal chain analysis, especially noise and AC, up to and including the ADC output (which is a digital domain signal) in analog units of volts and hertz. For TRAN analysis, note that transient variations in the voltage reference of an ADC may not have a clearly identifiable effect in this analog version of the output, even though they could indeed cause code changes in the digitally coded output of the real device.

Simulation speed

Full signal chains might experience long simulation times. First because the signal chain will be composed by multiple components, but also because ADCs and DACs can have complex models, so it is not uncommon for them to be the main cause for a slow simulation. To alleviate this, consider removing the ADC temporarily from the schematic when exploring aspects of the signal chain that are not impacted by it.

For information on the topic of speeding up simulations see this article [9].

References

[1] G. Alonso, "Get Up and Running with LTspice," Analog Devices, [Online]. Available: https://www.analog.com/en/analog-dialogue/articles/get-up-and-running-with-ltspice.html.

[2] Analog Devices, "ADI EngineerZone LTspice forum," [Online]. Available: https://ez.analog.com/design-tools-and-calculators/ltspice/.

[3] I. Analog Devices, ""Precision Low Power Signal Chains: optimized noise," [Online]. Available:," [Online]. Available: https://www.analog.com/en/applications/technology/precision-technology/precision-low-power.html.

[4] P. Karantzalis, "Webinar: Signal Chain Filtering: Beginning with the Basics," [Online]. Available: https://www.analog.com/en/education/education-library/webcasts/signal-chain-filtering-beginning-basics.html.

[5] G. Alonso, "LTspice: Using the .STEP Command to Perform Repeated Analysis," [Online]. Available: https://www.analog.com/en/technical-articles/ltspice-using-the-step-command-to-perform-repeated-analysis.html.

[6] G. Alonso, "LTspice: Generating Triangular & Sawtooth Waveforms," [Online]. Available: https://www.analog.com/en/technical-articles/ltspice-generating-triangular-sawtooth-waveforms.html.

[7] Analog Devices, "AD8237 Micropower, Zero Drift, True Rail-to-Rail Instrumentation Amplifier datasheet," [Online]. Available: https://www.analog.com/media/en/technical-documentation/data-sheets/AD8237.pdf.

[8] R. Delaney and P. Delizia, "Step-by-Step Noise Analysis Guide for Your Signal Chain," [Online]. Available: https://www.analog.com/en/analog-dialogue/articles/step-by-step-noise-analysis-guide-for-your-signal-chain.html.

[9] G. Alonso, "LTspice: Speed Up Your Simulations," [Online]. Available: https://www.analog.com/en/technical-articles/ltspice-speed-up-your-simulations.html.

[11] Analog Devices, "Precision wide bandwidth signal chain: optimized noise, bandwidth and area," [Online]. Available: https://www.analog.com/en/applications/technology/precision-technology/precision-wide-bandwidth.html.