Maximizing the Dynamic Range of Software-Defined Radio

by Bob Clarke, systems applications engineering manager, and Kevin Kreitzer, field applications manager, Analog Devices, Inc.

IDEA IN BRIEF

In broad terms, software-defined radio (SDR) is any radio where the signal chain is partially software. In practical terms, it will have some or all of these features: wideband, multiband, multimode, multidata rate, software reconfigurable, and its digital conversion (receive or transmit) is as close to the antenna as possible. Note that this description can also describe such RF instruments as a modern signal (spectrum) analyzer.

E-Systems (now Raytheon) in Garland, Texas, is believed to have built the first software-defined baseband receiver in 1984. The first software-defined baseband transceiver was probably the WSC-3(v)9, built by E-Systems' St. Petersburg, Florida, division circa 1987 for Patrick AFB. Haseltine and Motorola developed a later radio, SpeakEasy, for Rome AFB circa 1989. Modern examples include satellite and terrestrial radio, military joint tactical radio system (JTRS), and almost any cellular or land mobile radio terminal or base station.

For digital conversion and signal processing to work, theory says we should have a linear, time invariant system. Practice tells us that putting the analog pieces together is a series of compromises. By careful component selection and gain distribution, however, you can maximize the dynamic range of your SDR while still maintaining sensitivity. It doesn't matter whether the SDR is a communications receiver base station or a signal analyzer, the same rules still apply.

In some standards-based communications systems, that is cellular systems, SDRs operate in a controlled environment; the standard spells out the requirements for the receiver and transmitter, and the carrier adds margin to the standard. In other systems, such as military, amateur, and land mobile radio, the environment is uncontrolled; the nearest emitter can be next door and the furthest can be less than a whisper over the horizon.

Thus, when starting a design, you need to develop a checklist:

- What does the standard require?
- What are the desired minimum and maximum signal levels?
- How much filtering is needed?
- What image filter, channel filter, and antialiasing filters are available?
- Is group delay in the filters an issue?
- What architecture are you using? Zero IF, single, dual, or triple conversion?
- How are you generating quadrature?
- Are you working in the analog or digital (IF sampling) domains?

Choosing an ADC is a discussion in and of itself. The dynamic range of the ADC determines the systems architecture, and vice versa. Start by looking at the signal bandwidth and sampling frequencies (the exact sampling frequency is usually determined by such digital signal processing requirements as clock or frame rate, or both). And to get the full SNR of the ADC, especially when sampling high input frequencies, can you generate a good enough clock to sample at the desired frequency without degrading the ADC's specified SNR? For the system to be a linear time invariant system, the ADC must provide enough dynamic range to accommodate the desired signal plus interferers, as well as add margin to allow for signal fading and AGC response time.

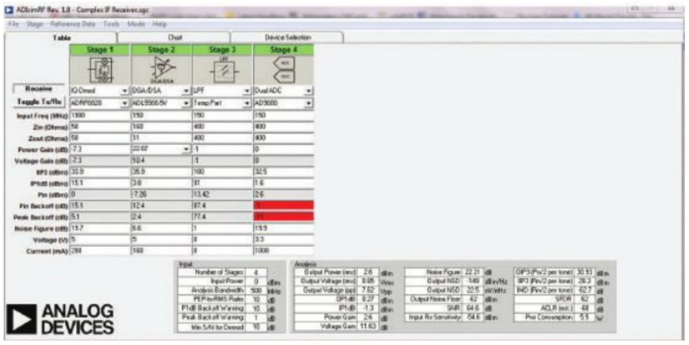
So how much dynamic range is enough? The highest performance software-defined radios (and RF lab instruments) typically use 14-bit to 16-bit, high speed ADCs to sample at as high a frequency as possible for signals as wide as 250 MHz. For testing the widest signals in standards such as the alphabet soup that is 802.11, the industry preference is to use dual high speed ADCs such as the 14-bit AD9680 for quadrature sampling of I/Q signals at baseband with I/Q bandwidths of 500 MHz or more. Some applications need less dynamic range and will typically use a 12-bit GSPS ADC such as the AD9625 to grab a 500 MHz wide chunk of spectrum and bring it down to baseband using an integrated digital downconverter.

The dynamic range of the ADC is the fundamental trade-off between analog and digital filtering. More analog filtering reduces the amplitudes of the interferers and the required range of the ADC, which must digitize both the desired signal and interferers to maintain a linear system. Analog filters, however, are nonideal and can exhibit group delay and phase. At the systems level, lots of filtering in the analog domain also means lots of potentially expensive mechanical shielding to maintain the filter's isolation as well as the possibility of cascading multiple filters at multiple IFs to minimize leakage around the filters. In contrast, digital filters can have great shape factors, no leakage, and near ideal characteristics, but require increasing the ADC's dynamic range to accommodate both the signal and interferers.

It may seem obvious, but you must design the receiver to keep the input to the ADC linear under all operating conditions. This requires, incorporating the AGC's response time into the ADC's margin. To do so, allow a certain number of dB as a margin to account for the change in the input signal while the AGC is reacting so that the receiver is not overloaded by a change in signal level.

In addition, you may also want to add additional margin for signal fading, whether due to atmospheric conditions in the case of lower frequencies or the signals being blocked by buildings or foliage in the case of UHF and microwave signals. On top of these, you also need to include C/N ratio for demodulation plus margin for adjacent and alternate channels interferers and possible PA feed through in full-duplex systems.

Another thing to keep in mind is that narrow-band receivers have a wider AGC range than wideband receivers. Wideband receivers basically move a large slice of spectrum up or down a bit in amplitude—typically less than 10 dB—to keep it centered in the ADC's linear window, as would be the case when digitizing an entire cellular band. In contrast, narrow-band receivers rely heavily on filtering to minimize the number of signals in the pass band, but must accommodate larger interferers, generally in uncontrolled environments, and their AGC works on signals in the much narrower pass band.


ADIsimRF Rev. 1.8 - Complex IF Receiver.agc
File Stage Reference Data Tools Mode Help
Table
Chart
Device Selection
ANALOG DEVICES

	Stage 1	Stage 2	Stage 3	Stage 4
Receive	IQ Dmod	DGA/DSA	LPF	Dual ADC
Toggle Tx/Rx	ADRF6820	ADL5566-5V	Temp Part	AD9680
Input Freq (MHz)	1900	150	150	150
Zin (Ohms)	50	160	400	400
Zout (Ohms)	50	11	400	400
Power Gain (dB)	-7.3	22.07	-1	0
Voltage Gain (dB)	-7.3	10.4	-1	0
IIP3 (dBm)	33.9	35.9	100	32.5
IP1dB (dBm)	15.1	3.8	91	1.6
Pin (dBm)	0	-7.26	13.42	2.6
Pin Backoff (dB)	15.1	12.4	87.4	[illegible]
Peak Backoff (dB)	5.1	2.4	77.4	[illegible]
Noise Figure (dB)	15.7	8.6	1	19.9
Voltage (V)	5	5	0	3.3
Current (mA)	280	160	0	1000

Input

Number of Stages	4	
Input Power	0	dBm
Analysis Bandwidth	500	MHz
PEP-to-RMS Ratio	10	dB
P1dB Backoff Warning	10	dB
Peak Backoff Warning	1	dB
Min S/N for Demod	10	dB

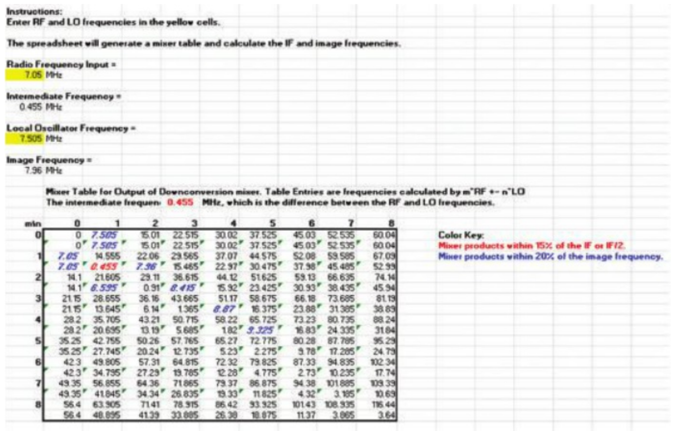
Analysis

Output Power (rms)	2.6	dBm
Output Voltage (rms)	0.85	Vrms
Output Voltage (pp)	7.62	Vpp
OP1dB	0.27	dBm
IP1dB	-1.3	dBm
Power Gain	2.6	dB
Voltage Gain	11.63	dB
Noise Figure	22.21	dB
Output NSD	-149	dBm/Hz
Output NSD	22.5	nV/√Hz
Output Noise Floor	-62	dBm
SNR	64.6	dB
Input Rx Sensitivity	-54.6	dBm
OIP3 (Po/2 per tone)	30.93	dBm
IIP3 (Pin/2 per tone)	28.3	dBm
IMD (Pin/2 per tone)	-62.7	dB
SFDR	62	dB
ACLR (est.)	-68	dB
Pwr Consumption	5.5	W

Figure 1. Screenshot for ADIsimRF™ Modeling Tool Showing a Direct Conversion Receiver

When you set up a cascaded noise figure and intercept model for a receiver design, you have to model the system three times: once for minimum signal level—that is, AGC off with maximum gain; again for maximum signal level—that is, AGC on with maximum gain reduction; and finally, for the receiver's nominal input level. You also have to account for intermodulation effects in all three models. Fortunately, free tools such as ADI's ADIsimRF (Figure 1) will help do this for you; such tools often come with a built-in library of models for RF gain blocks, mixers, attenuators, baluns, filters, and high speed converters.

Frequency planning is another fun issue that requires extensive analysis. Not only do you need a mixer table (Figure 2) for each mixer, but you may also want to create a similar DAC table for the transmit path. You also have to consider which Nyquist zone to operate the converter in (either ADC or DAC). The system clock is often a multiple of the frame rate (which is why multiples of 1.2288 MHz and 13 MHz are commonplace). With luck, you can use a frequency high enough that its harmonics do not fall into a band or on a signal of interest. You need to minimize internal and external interference by careful selection of the system clock, intermediate frequencies, and local oscillator (LO) frequencies because these frequencies will mix with unforeseen consequences.



Instructions:
Enter RF and LO frequencies in the yellow cells.

The spreadsheet will generate a mixer table and calculate the IF and image frequencies.

Radio Frequency Input =
7.05 MHz

Intermediate Frequency =
0.455 MHz

Local Oscillator Frequency =
7.505 MHz

Image Frequency =
7.96 MHz

Mixer Table for Output of Downconversion mixer. Table Entries are frequencies calculated by m*RF +- n*LO
The intermediate frequen 0.455 MHz, which is the difference between the RF and LO frequencies.

m\n	0	1	2	3	4	5	6	7	8
0	0	7.505	15.01	22.515	30.02	37.525	45.03	52.535	60.04
	0	7.505	15.01	22.515	30.02	37.525	45.03	52.535	60.04
1	7.05	14.555	22.06	29.565	37.07	44.575	52.08	59.585	67.09
	7.05	0.455	7.96	15.465	22.97	30.475	37.98	45.485	52.99
2	14.1	21.605	29.11	36.615	44.12	51.625	59.13	66.635	74.14
	14.1	6.595	0.91	8.415	15.92	23.425	30.93	38.435	45.94
3	21.15	28.655	36.16	43.665	51.17	58.675	66.18	73.685	81.19
	21.15	13.645	6.14	1.365	8.87	16.375	23.88	31.385	38.89
4	28.2	35.705	43.21	50.715	58.22	65.725	73.23	80.735	88.24
	28.2	20.695	13.19	5.685	1.82	9.325	16.83	24.335	31.84
5	35.25	42.755	50.26	57.765	65.27	72.775	80.28	87.785	95.29
	35.25	27.745	20.24	12.735	5.23	2.275	9.78	17.285	24.79
6	42.3	49.805	57.31	64.815	72.32	79.825	87.33	94.835	102.34
	42.3	34.795	27.29	19.785	12.28	4.775	2.73	10.235	17.74
7	49.35	56.855	64.36	71.865	79.37	86.875	94.38	101.885	109.39
	49.35	41.845	34.34	26.835	19.33	11.825	4.32	3.185	10.69
8	56.4	63.905	71.41	78.915	86.42	93.925	101.43	108.935	116.44
	56.4	48.895	41.39	33.885	26.38	18.875	11.37	3.865	3.64

Color Key:
Mixer products within 15% of the IF or IF/2.
Mixer products within 20% of the image frequency.

Figure 2. Sample Mixer Table Showing the Many NF1 ± MF2 Products Created in the Mixing Process, Where F1 and F2 Are the Mixer's RF Input and Local Oscillator Input Frequencies, Respectively

Once you have set up your cascaded noise figure and intercept model for the number of stages and type of functions, such as filters, mixers, and amplifiers, you need to perform some side calculations.

First, you'll need to calculate the ADC's noise figure (NF) using the equation

$$NF = P_{FS} + 174\text{ dBm} - SNR - 10\log_{10} B \text{ (at 300°K)}$$

Where P_{FS} is the ADC's full-scale input power in dBm, P_{FS} (dBm) = 10 $\log_{10}$ [P_{FS} (mW)/ 1 mW], SNR is the ADC's signal-to-noise ratio in dB, and B is the bandwidth being digitized, taking into account the noise bandwidth of the input filter (Figure 3).

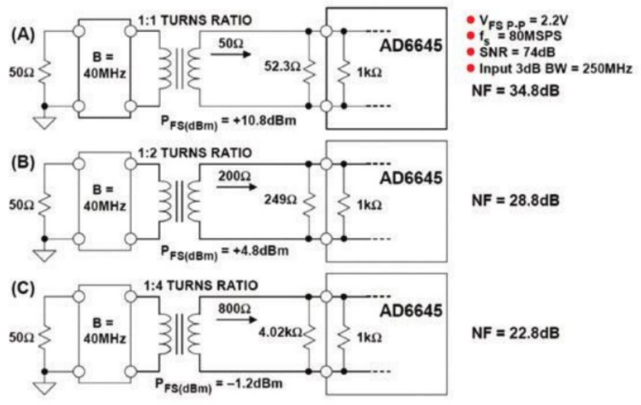


Figure 3. The Relationship Between Noise Bandwidth and 3-dB Bandwidth for a Butterworth Filter

Note that this bandwidth may be wider than the signal bandwidth if you want to digitize the desired signal plus interferers in order to filter the interferers out in the digital domain.

Fortunately, you can improve an ADC's noise figure by oversampling the input signal; in this case, the equation for noise figure becomes $NF = P_{FS} + 174\text{ dBm} - SNR - 10\log_{10} B - 10\log_{10} [fs/2B]$, where fs is the sample clock and B is still the signal bandwidth (or bandwidth to be digitized). Some IF sampling ADCs, such as the AD9874 and AD9864, use both oversampling and noise shaping in a band-pass Σ-Δ architecture. These ADCs are actually complete IF subsystems—accepting an IF input and providing SNRs approaching 100 dB and 16 or 24 bits of decimated I/Q data at their output.

Oversampling isn't the only way to improve the ADC's noise figure. You can also use a transformer to increase the ADC's input voltage with noiseless gain, as shown in Table 1 for the AD6645.

Table 1. Applying Noiseless Voltage Gain Can Improve Performance—But Don't Overdrive the ADC!

Number of Poles	Ratio of Noise Bandwidth to 3 dB Bandwidth
1	1.57
2	1.11
3	1.05
4	1.03
5	1.02

The last side calculation you need to do is for the conversion clock. The ability (or lack thereof) to generate low jitter clocks is why ADCs are not yet placed at the antenna for high dynamic range receivers, as we shall see below. The equation for an ADC's theoretical SNR as a function of clock jitter is $SNR = 20\log_{10}[1/(2\Pi f t_j)]$, where f is the frequency being sampled in Hz and t_j is the clock jitter in seconds.

Alternately, given a desired SNR in dB and maximum input frequency f in Hz, $t_j = 1/(2\Pi f 10[SNR/20])$.

When IF sampling, for example, if you want to sample a 20 MHz wide signal at an IF of 240 MHz, the highest frequency component being sampled is the IF plus half the signal bandwidth, or 250 MHz. Solving for 80 dB SNR at a maximum input frequency of 250 MHz, this yields a maximum clock jitter of 63.66 fs. You can simulate the performance of a PLL/VCO and calculate the jitter for various types of loop filters and circuit configurations using such free tools as ADIsimPLL™ and ADIsimCLK™. Using these tools, you can optimize the design of a loop filter for the best phase noise and thus minimum jitter at the expense of increasing the filter's settling time—generally not a problem for a fixed frequency clock for a high speed ADC.

Thus far, we have reviewed just some of the circuit elements, calculations, and simulation tools that go into maximizing the dynamic range of a software-defined radio, with emphasis on the performance of the ADC and frequency planning. Due to space limitations, we've ignored the effects of gain and gain distribution on distortion, but that can be a topic for another day.

RESOURCES

Share this article on