Ray Mack, W5IFS

17060 Conway Springs Ct, Austin, TX 78717; w5ifs@arrl.net

SDR: Simplified

Reader Feedback

I got a note from Peter Anderson, KC1HR, who wrote a *QEX* article in 1994 about using the Harris Digital Down Converter ICs for an SSB receiver.[1] These ICs implement the sample rate reduction using decimation and filtering that we looked at last issue. His observation is that the ICs from Harris and Texas Instruments are fairly old and do not have enough internal precision to handle large dynamic range for narrow band applications. Many modern implementations (including the TAPR software defined radio) use an FPGA with more precision to implement the dedicated functions for sample rate reduction. The AD9864 is much newer and uses 24 bit precision internally. This part also starts its work from a much lower starting frequency than the Harris parts. We will look at using this IC in a future column.

I have heard from some of you and have started an e-mail update list. I am also planning to set up a Web site, so we have a place to post updates, corrections, and software resources between issues of the magazine. I will continue to provide files to post on the ARRL *QEX* files Web site.

Nyquist Meets Real World

The Nyquist theorem says that you can exactly recreate an analog signal that has been sampled by passing the signal back into a digital to analog converter (DAC) as long as you select the sample rate greater than twice the highest frequency component of the input signal. The assumption is that the input analog to digital converter (ADC) and the output DAC perform identically to the sample theorem that is central to the mathematics. The real world is never so easy. In general, the ADC comes pretty close to converting the analog signal into a sequence of samples that matches how an actual impulse sample sequence would work. Of course, there are noise sources that add noise to the system, but those are just a normal part of electronic systems. We'll look at noise sources in ADC operation in the future.

The DAC system, on the other hand, suffers from a serious disconnect between the math and the electronics. The math assumes that we are going to convert a sequence of infinite height pulses with an area equal to the sample value into a

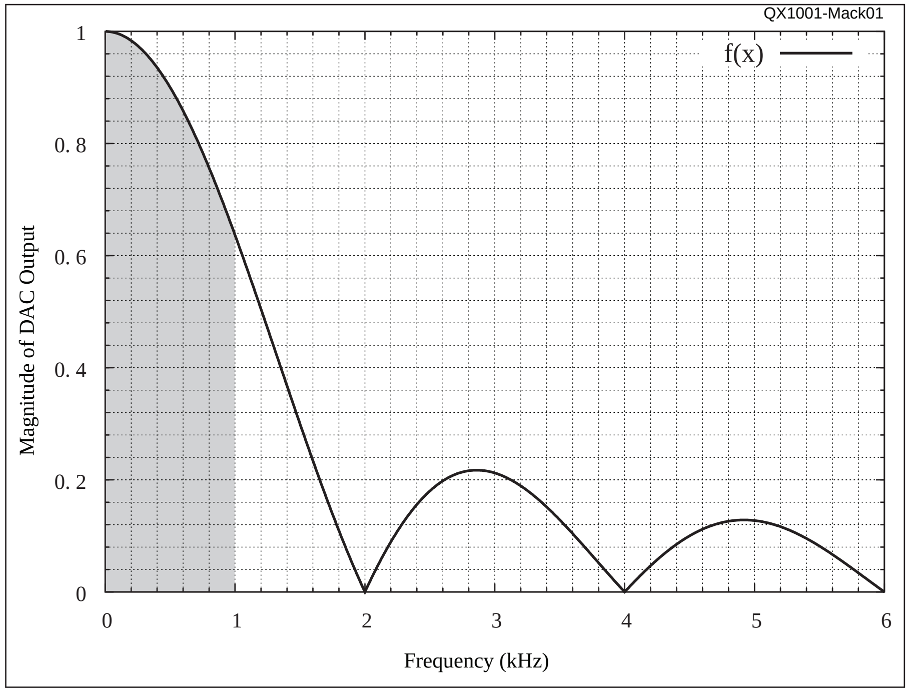


Figure 1 — This is a graph of the frequency spectrum for 2 kHz sample rate and frequencies from dc to 1 kHz. The shaded area shows the RMS output that will be achieved at each frequency in the first Nyquist zone. The envelope is the magnitude of the sinc function.

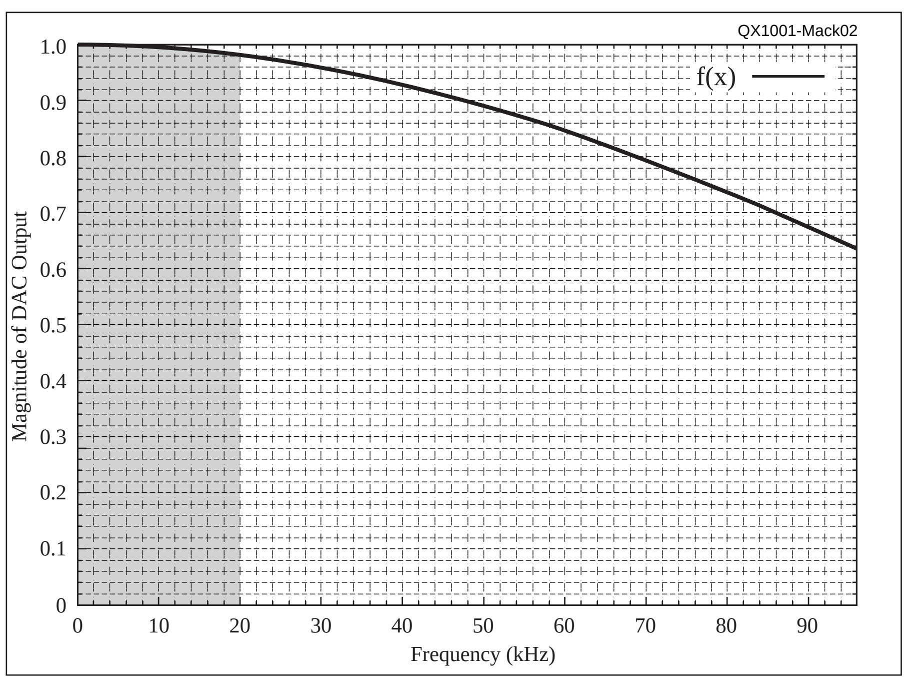


Figure 2 — Here is a graph of the frequency spectrum for CD audio that is sampled at 192 kHz. The shaded area shows how the sinc function affects frequency. Without compensation, the frequencies near 20 kHz are only attenuated by 2% (–0.18 dB).

[1]Peter Traneus Anderson, KC1HR, "A Simple SSB Receiver Using a Digital Down Converter," *QEX*, Mar 1994, pp 17-23.

continuous signal. A low pass filter (especially a brick wall filter) will convert those impulses into the appropriate sine waves with the proper amplitude and phase. As the low pass filter becomes less than a brick wall, the conversion allows more of the energy at higher frequencies to leak into the output and distort the waveform. So our first problem on output conversion is the fidelity of the low pass reconstruction filter.

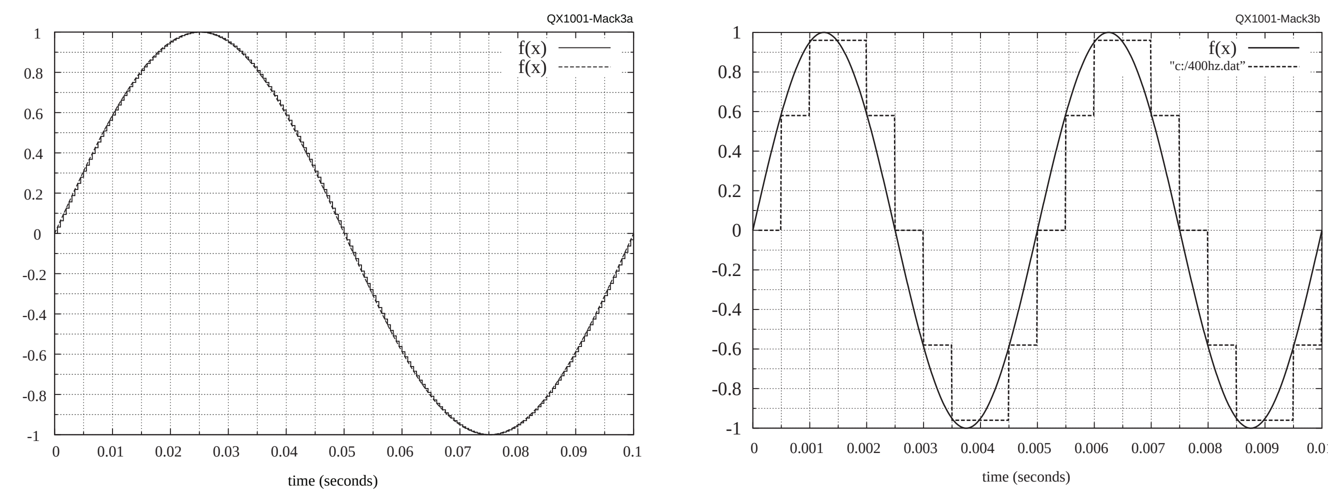
The next problem comes from the way a DAC works. Mathematicians call this a *zero order hold* circuit. You can think of the zero order hold function as an electronic transform that converts the rectangular impulse (infinite height, zero width, but finite area) into a rectangular pulse that is one sample period wide with the same area as the input impulse. Remember that the impulse sequence created a comb in the frequency domain, with all of the elements having the same height and containing all harmonics of the sample frequency, out to infinity. When you run the impulse sequence through a zero order hold (the DAC), you change the frequency content of the resulting signal. The signal still extends to infinity, but the phase and amplitude vary with frequency. That variation is defined by the function (sin x)/x. This function is called the ***sinc*** function. Enter ***wiki dac sinc function*** in your search engine to see a mathematical explanation.

Let's look at an example in which we have our DSP use a sample rate of 2 kHz, so our Nyquist area extends from dc (zero hertz) to just below 1 kHz. We will assume our system has a brick wall filter at 1 kHz. We can have our DSP create a direct digital synthesis sine wave with 1 V RMS, and vary the frequency from 1 Hz up to 999 Hz. Figure 1 shows the actual RMS output voltage we measure versus frequency. Figure 1 also shows several Nyquist zones beyond the first, so you can see the shape of the sinc function over a larger frequency range. Notice that the output voltage from the DAC and low pass filter is only 0.637 V for the 999 Hz waveform.

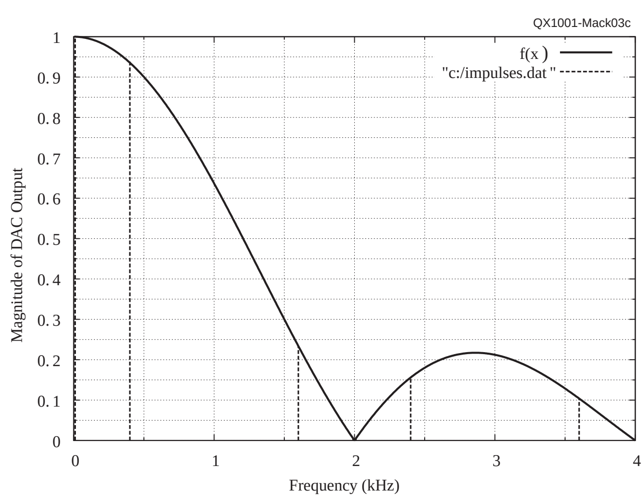
One way to solve the problem so that all frequencies from dc to 999 Hz have 1 V RMS is to change the low pass reconstruction filter to have a shape that is the inverse of the sinc function. This method is frequently used but hard to implement with analog filters. The preferred option is to apply the inverse sinc compensation to the digital samples.

The sinc function issue was a problem for CD audio (and computer audio since it originally used CD technology) when it was first introduced in the mid 1980s. The CD sample rate of 44.1 kHz was about the fastest that could easily be implemented at 16 bit resolution 25 years ago. This put the Nyquist frequency at 22.05 kHz, and allowed reasonable analog reconstruction filters that were able to cut off at the upper audio range of 20 kHz. CD audio players had to implement filters to compensate the higher frequencies because of the sinc problem.

You will notice that the sinc waveform is essentially flat very near dc. If we sample

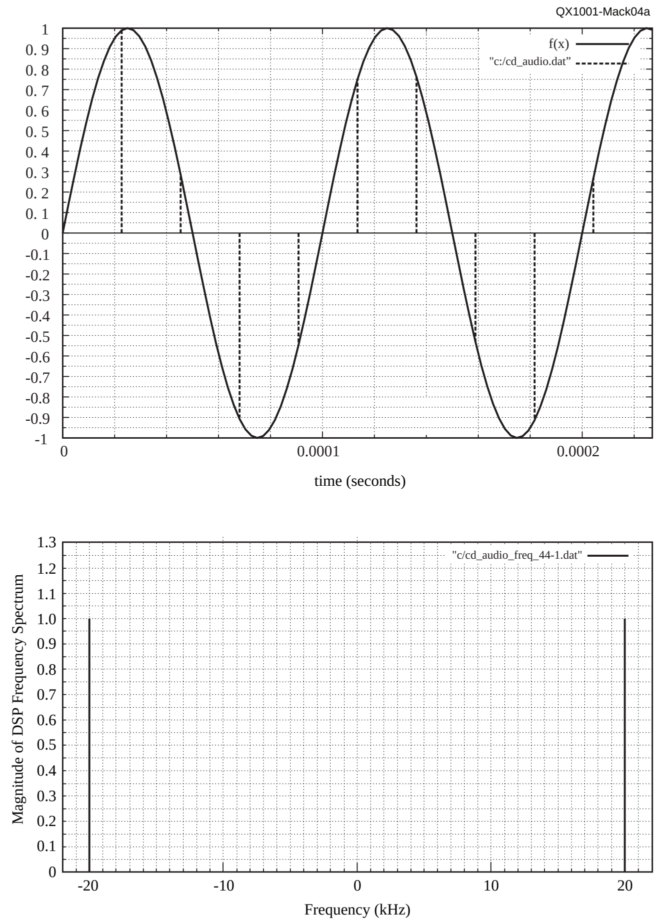


(A) (C)

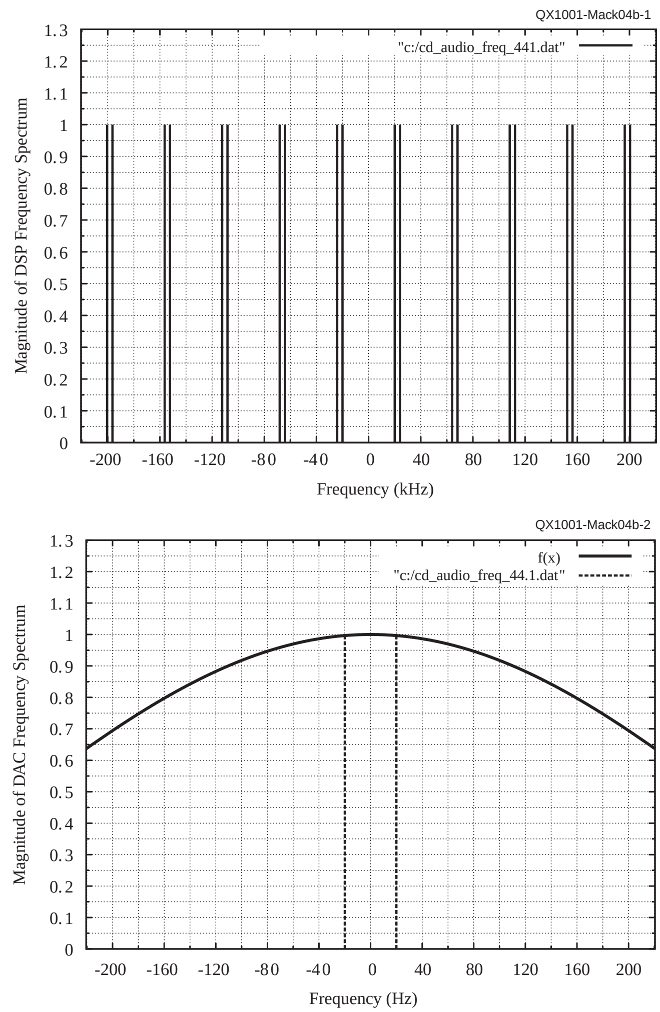


(B)

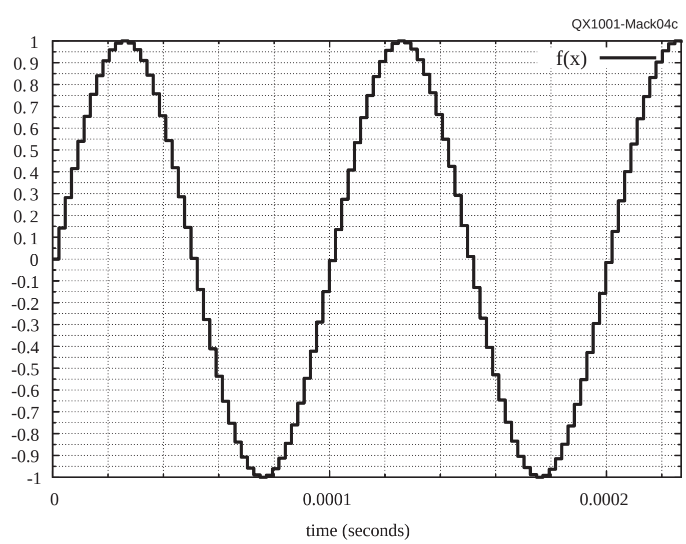
Figure 3 — Part A is a graph of a 10 Hz sine wave and the DAC output of the 10 Hz sine when sampled at 2 kHz. The area between the two plots is extremely small, so very little energy appears at the DAC output at harmonics of the sample frequency. At B is a graph of a 400 Hz sine wave and the DAC output of the 400 Hz sine when sampled at 2 kHz. The area between the two plots is quite large, so there is substantial energy at the DAC output at harmonics of the sample frequency. Part C shows the DAC output spectrum for a 10 Hz signal and a 400 Hz signal when sampled at 2 kHz. The energy for the 10 Hz signal appears at full strength just to the right of the axis. The 400 Hz signal is down 0.5 dB from full strength. The 10 Hz signal is almost imperceptible in the second, third, and fourth Nyquist zones. The 400 Hz signal has substantial energy in each of the higher Nyquist zones. It is only 12 dB below the fundamental in the second Nyquist zone.



(A)



(B)



(C)

Figure 4 — Part A is a plot of the samples and original waveform of a 20 kHz CD audio tone when sampled at 44.1 kHz, along with the DSP spectrum. Part B shows a plot of the spectrum of a 20 kHz CD audio tone after inserting nine zero samples between each original sample. Note that the spectrum contains equal height tones that are two 20 kHz sidebands around each harmonic of the original 44.1 kHz sample frequency. The bottom graph shows the spectrum and sinc reduction after the top spectrum is passed through the interpolation filter. The strength of the output has improved from 3.9 dB down to only 0.18 dB down. In Part C we have a plot of the DAC output for the 20 kHz CD audio tone after a 10× interpolation filter.

at a higher frequency, we can move the highest frequency of interest closer to the top part of the sinc waveform and lower the amount of compensation needed or ignore compensation altogether. In the case of CD audio, our highest frequency of interest is approximately 20 kHz. If we sample at 200 kHz, we move all of our information to the bottom 20% of the sinc function. Figure 2 shows the effects of using 192 kHz sampling (the current high end in common computer sound cards). The amount of compensation needed for excellent fidelity is much smaller. A good rule of thumb is that sampling at least ten times more than the required Nyquist rate will essentially eliminate the need for compensation.

We have looked at the math reason for sinc and seen a way to mitigate its effects, but why does it happen? Figure 3A shows a sine wave and its DAC waveform for 10 Hz with 2 kHz sampling. Figure 3B shows a sine wave and its DAC waveform for 400 Hz with 2 kHz sampling. Notice that the 10 Hz sampled waveform is a very close approximation to the sine wave. There is almost no variation from the actual sine wave. The 400 Hz waveform is another matter. Notice that there is significant area between the true sine wave and the DAC waveform. All of the area between the DAC waveform and the actual sine wave represents energy at frequencies other than 400 Hz. That energy shows up in the second, third and higher Nyquist zones. For the 10 Hz waveform, we see that the harmonic energy in the upper Nyquist zones falls under a very low part of the sinc function. This fits with our earlier observation that the size of the difference from the actual waveform is proportional to the energy in the higher Nyquist zones.

Sample Rate Up Conversion

If we could take our CD audio at 44.1 kHz sample rate and increase the sample rate to 441 kHz (a 10× increase in sample frequency), we could eliminate a lot of the error in the signal compared to the original. One way to accomplish this sample rate conversion is to take two adjacent samples and draw a straight line between them. For our 10× increase in sample rate we would add 9 new samples with values between the two original samples. This is called interpolation, so the process of sample rate up conversion is called interpolation. This method does not really solve the problem. In the case of a sine wave at 20 kHz, we will have a sequence of sloped lines. The error is less but the error is still substantial.

We looked at sample rate down conversion (also called decimation or digital down conversion) in the Nov/Dec '09 installment. The first application was to reduce the sample rate using a low pass filter, which made the first Nyquist zone smaller in frequency range. This allowed the frequency range to more closely match the capabilities of our digital signal processor. There is an inverse operation that uses the same principles to increase the sample rate, and therefore increase the size of the first Nyquist zone. Let's look at our 20 kHz CD audio tone again.

I mentioned the limited frequency range of the DSP view of our signals last time, but did not go into detail. When we sample a waveform in the real world and convert it into the DSP world, we reduce the frequency domain from the real world infinite to the perfect, exact, and limited DSP frequency domain. The CD audio example has frequencies that extend from –22.05 kHz to +22.05 kHz and *only* those frequencies exist for the purposes of the math. The boundaries are entirely a consequence of the sample frequency. If we do a frequency domain plot (Discrete Fourier Transform) of our 20 kHz audio tone, we get a single line at –20 kHz and another at +20 kHz, and no other spectrum lines show up while we are working with DSP.

DSP interpolation uses the math to do a couple of transformations on the data. Figure 4A shows The input sine wave, the samples, and a frequency plot for a 10 sample sequence of the 20 kHz signal sampled at 44.1 kHz (22.7 µs between samples). Figure 4B shows what happens if we create a new sequence by adding nine zero samples between each of these original samples with 2.27 µs between each sample (441 kHz sample rate). Notice that the frequency plot now has equal height lines at –200.5 kHz, –196.4 kHz, –156.4 kHz, and so on, up to +200.5 kHz. These frequencies represent the fundamental at 20 kHz as well as the sum and difference between each harmonic of 44.1 kHz and the 20 kHz signal. It is a relatively simple matter, using DSP techniques, to create a brick wall low pass filter that will pass all frequencies below 22 kHz. The resulting sequence of samples is now 100 samples long, with each sample being a reasonable approximation to the 20 kHz sine wave. The process of adding the zero samples and low pass filter can be combined in one operation, and is called an interpolation filter.

Notice that our brick wall filter has to be pretty good, since it must pass 20 kHz and completely reject 24.1 kHz. A system with a signal so close to the top of the first Nyquist zone is a candidate for doing a 2× interpolation first, to move the first two sets of signals further apart and make the filter less critical. We haven't looked

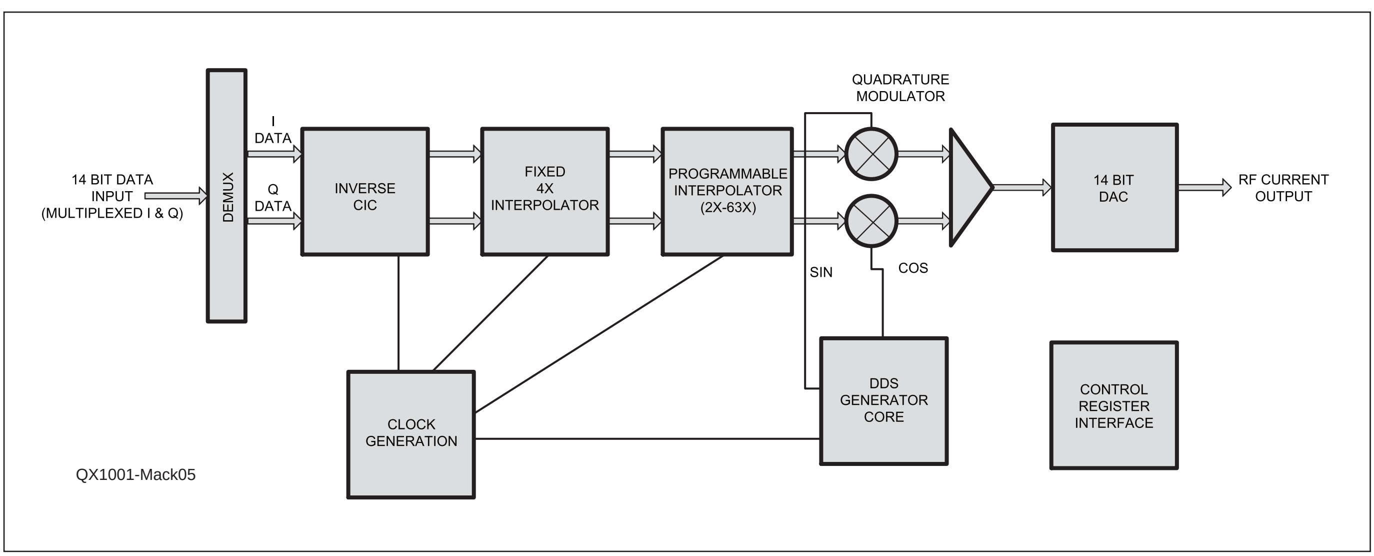


Figure 5 — A simplified block diagram of the AD9857 transmitter IC showing two stages of interpolation and a final translation to RF using an image reject mixer. All functions occur in digital circuits prior to the output DAC. Down- load the data sheet from analog.com to see a more detailed block diagram.

at filters in detail yet, but the ratio of filter bandwidth to total bandwidth affects the computer resources. A filter with a ratio of 0.5 requires significantly less CPU power than a filter with a 0.1 ratio. This is similar to the analog world, where a 500 Hz CW filter requires a lot more crystals for sharp skirts than a 6 kHz AM filter.

We looked at band pass decimation in the Nov/Dec '09 issue, where we were able to increase the ratio of our signal bandwidth to total DSP bandwidth and also change the frequency of the signals. Again, the inverse function works for us. It is very easy to start with audio from 300 Hz to 3 kHz and create a DSB suppressed carrier signal in DSP at 200 kHz, for example, with an 800 kHz sample rate. Then we can create a DSP brick wall band pass filter to create either upper sideband or lower sideband by filtering. Now we can use the same zero sample insertion method we talked about above to raise the sample rate. We could insert four zero samples between each original sample to increase the sample rate to 4.0 MHz. This would result in new transmitter signals at 200 kHz, 600 kHz, 1000 kHz, 1400 kHz, and 1800 kHz. We can create a band pass filter centered on any one of these signals to simultaneously increase the sample rate to avoid sinc issues as well as translate the SSB signal to a new frequency. An interesting example would take the new 200 kHz signal and create a 2× interpolation filter to increase the sample rate again to 8.0 MHz. The band pass interpolation filter can be centered at 3.8 MHz to move our original 200 kHz SSB signal to the 75 m band without any analog operations, other than the final DAC step with its reconstruction filter. Another interesting example would band pass select the 1000 kHz signal and use a 3× interpolator to generate an SSB signal at 5.0 MHz for the 60m band.

The interpolation and digital filtering for the higher frequencies is frequently performed by an FPGA in modern systems. A large FPGA can contain the logic to insert the zero samples and perform the multiply/accumulate functions of the digital filtering. The interpolation is a fixed function and does not lend itself to the general purpose capabilities of a DSP computer.

The AD9756 and AD9857 are interesting ICs from Analog Devices. These parts implement the core of a digital radio transmitter. Both parts implement a 200 MHz sample rate system, but the AD9856 has a 12 bit data path and the AD9857 has a 14 bit data path. The block diagram for the parts is shown in Figure 5. The intent of these parts is that you will create a baseband version of your transmitter signal with both *I* and *Q* versions. The sample rate of the baseband signal is then interpolated from perhaps 1 MHz or 10 MHz up to the final sample rate of the system (potentially as high as 200 MHz). The interpolation is the low pass baseband variety rather than the band pass up conversion type. The chip then uses the *I* and *Q* output of a direct digital synthesizer to create an image reject mixer that operates entirely on digital samples. The result is a transmitter signal that is entirely digital until the DAC connects to the antenna. This would be a QRPP signal, since the DAC can directly generate about 10 dBm at its output. The IC actually has three modes. The first is the transmitter mode just described. It can also be used as a DDS single tone generator for applications like a local oscillator. The third mode just operates as a baseband interpolation filter and DAC to raise the sample rate.

Future Issues

Thanksgiving and Christmas breaks are coming, so my intention is to use the extra time to create experimental boards for both the AD9857 and the AD9864. The goal is to create an SSB transmitter for 160 m through 6 m (including 4 m for the UK) where the generation is done entirely in the digital world, with the exception of the final power amplifier. The AD9864 is billed as a 10 MHz to 300 MHz digital receiver IC. It will be interesting to see if it is possible to rearrange the connections to create a companion receiver that will receive the same frequency range. This IC first converts the input to a frequency between 1.6 MHz and 3 MHz, so it is not a true "connect to antenna" software defined radio IC. It samples the output of the mixer with a 24 bit ADC and processes it in a set of decimation filters to drop the frequency down to the 200 kHz range.

Another task is to develop some software to demonstrate both decimation and interpolation. I have finally realized that I cannot avoid having a *Linux* development system. So much of the resources for the Stamp are developed only on *Linux* and must be adjusted to work with *Windows* based tools. My system isn't all the way built yet, so that is another task for the holidays. QEX