A High-Performance Digital-Transceiver Design, Part 1

Data-converter technology has made tremendous strides in the past several years. Let's take a look at how we can achieve high performance in an almost-all-digital radio design.

By James Scarlett, KD7O

There has been much discussion in *QEX* recently on the need for, or the achievement of, high performance in our radios.[1, 2, 3] This has particularly been the case with receiver strong-signal performance. It has often been shown that amateurs can improve upon existing commercial design by paying proper attention to the analog front end.

The introduction of cost-effective digital signal processing (DSP) has improved performance in some areas.[4] With respect to strong-signal capability, however, good analog designs are still king. For several years, I have been looking at approaches to an analog front end from a system level. As the time to build the radio approached, I began to look at applying digital techniques as well. What I learned about the state of digital technology was surprising.

The goal for digital radios has always been to get the converter as close to the antenna as possible, for both receive and transmit; however, data converters have been the limiting factor in getting the necessary performance. They lacked the dynamic range and were noisy. This means we must employ analog circuitry to overcome converter limitations.

The state of the art has changed enough that we can now convert directly between the digital realm and the HF bands.[5] We still need to amplify the inputs to overcome converter noise. To prevent overload, we need to use band-pass filters instead of a low-pass filter above 10 meters–which high-performance analog designs do anyway. We cannot yet reach the strong-signal performance of the best analog front-end designs, but it is now possible to exceed the performance of even high-end commercial gear using a direct-to-RF conversion scheme.

This first article in the series will look at the design process for such a transceiver. The emphasis will be on the areas that differ from typical analog designs. In future segments, we'll look at the detailed implementation and performance measurements.

Architectural Overview: The Receiver

By performing conversions directly at RF, receiver and exciter architectures may be greatly simplified. Fig 1 shows the basic block diagram of a

[1]Notes appear on page 44.

kd7o@arrl.net

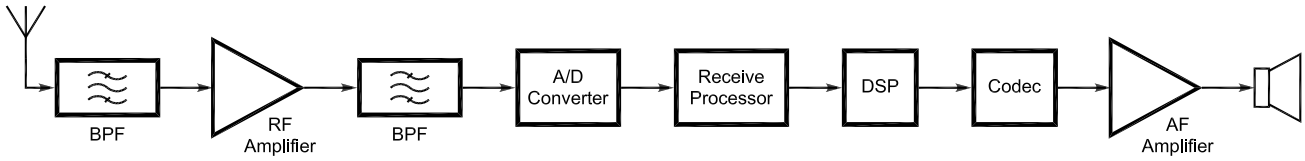

Fig 1—Receiver Block Diagram.

receiver. As you can see, it's missing many stages we're used to seeing!

Since we will be doing all frequency translations digitally, there is no need for an analog mixer. The only analog filters are the band-pass filters for the entire amateur band of interest. All narrow filtering is done digitally. We need a preamplifier to overcome noise in the analog-to-digital converter (ADC).

The weakest link in this architecture is the ADC. A little later, we'll look at these weaknesses and see how we can mostly work around them. The sampling frequency must be high enough to satisfy Nyquist at the widest bandwidth we'll use. There are advantages to getting the sampling rate much higher, though. Therefore, we'll be using a sampling clock on the order of 65 MHz.

The problem is this: DSPs can't accomplish much at that sampling rate! Therefore, we must have some sort of interface between the ADC and the DSP. This is the job of the *receiving signal processor* (RSP), also known as a digital down-converter (DDC). This nifty chip converts the signal to baseband and reduces the sampling rate to a speed that can be handled by the DSP. It also provides extensive filtering and splits the signal into its analytical components for demodulation. The output is then sent to the DSP for further processing.

Architectural Overview: The Exciter

Fig 2 shows the basic exciter block diagram. This is essentially the receiver in reverse order, and conceptually it is just as simple.

The DSP hardware communicates an analytical signal pair representing the baseband information to a *transmitting signal processor* (TSP). This device is essentially an RSP in reverse. It interpolates the signal up to the final sampling rate, while providing the necessary filtering. The analytical pair is then translated to the desired output frequency and combined.

The TSP output is converted to analog at the desired RF frequency by the digital-to-analog converter (DAC). Like ADCs, DACs have come a long way in recent years. Again, higher sampling rates help us to meet the Nyquist criterion, avoid aliased signals and improve noise performance.

The output of the DAC must be filtered to eliminate images created by aliasing. As the output frequency lowers with respect to the sampling frequency, this task becomes easier. Given the same sampling rates, any filters that meet spurious requirements for the receiver will be more than adequate for the exciter. Therefore, it's convenient to simply use the same band-pass filters for both.

The filter output is then sent to the amplifier chain to achieve the final output power. This output is also filtered to ensure low spurious levels.

I have avoided much of the theory in the previous discussion. It has been well covered in *QEX*, so this article will concentrate on the actual design process, with emphasis on areas unique to this architecture. We will revive the theory as needed to make design decisions.

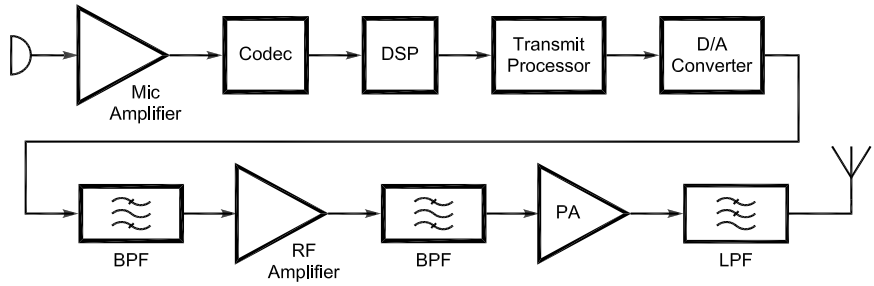

Fig 2—Exciter Block Diagram.

Table 1—Performance Specifications

RX		*TX*	
20 m *IP3*	> +30 dBm	Output *IMD3*	< –30 dBc
10 m *IP3*	> +20 dBm	Output *IMD9*-11	< –60 dBc
20 m *NF*	22 dB	Output Spurs	< –55 dBc
10 m *NF*	12 dB	Output power	50 W
Image Rejection	> 100 dB	Carrier rejection	< –60 dB
Filter Bandwidth	programmable	Sideband rejection	< –60 dB
Audio Power	4 W, < 10% THD	—	—
Modes	programmable		programmable

Receiver Design

We will begin the design discussion with the receiver and then move on to the transmit side. Before tackling the actual design, we will review the required performance characteristics. Several of these are outlined in Table 1.

A few notes are in order for some of the specifications. Notice that the two-tone dynamic range is not specified. This can be calculated from the given specifications, using the formulas in Chapter 17 of any recent *ARRL Handbook*.[6] The problem with this number is that it does not give adequate information about strong-signal capability. A 20-meter receiver can have a 100-dB dynamic range (*DR*) with a noise figure (*NF*) of 5 dB, but 10 dB of that dynamic range is wasted on atmospheric noise. A receiver with 90 dB of *DR* and a *NF* of 20 dB would handle strong signals just as well while retaining adequate

sensitivity for 20 meters. Thus, the third-order intercept point (*IP3*) is a far more useful measure of signal-handling capability.

No number is given for the compression point. That is because of the nature of this architecture. As an ADC is a voltage device, we will not get compression *per se*. The ADC has a maximum input range, outside of which it will clip the incoming signal. Therefore, the total incoming power of signals and noise within the bandwidth of the front-end filter, at the ADC, must not reach this hard limit. Because of the low gain used ahead of the ADC in this architecture, this can be assured with a high level of confidence.

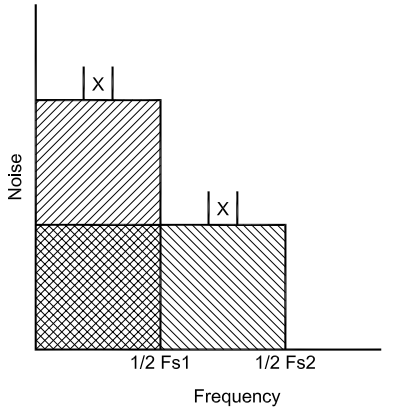

Fig 3—An illustration of processing gain. An ADC creates quantization noise that is spread from dc to half the sampling rate. In this case, sampling rate Fs2 is twice sampling rate Fs1. Both examples have the same total noise power, but it is spread over a larger frequency range with the higher sampling rate. Therefore, the noise power in bandwidth 'X' for sampling rate Fs2 is one half the noise power for Fs1, for a 3-dB improvement in the SNR.

The noise figure of this receiver in its default state will vary from band to band. As John Stephensen, KD6OZH, showed in his *QEX* article (see Note 2), external noise varies across the HF spectrum. For this reason, Table 1 reflects the *NF* requirements for both 20 and 10 meters. The receiver is capable of achieving a lower *NF* on the lower bands to take advantage of those occasions when less external noise is present.

Notice the programmable filter bandwidths and modes. Since this receiver is software/DSP based, it can be adapted to whatever characteristics we want!

A Dynamic-Range Spreadsheet

When I was first looking at receiver architectures, I developed a spreadsheet for analyzing noise-figure and dynamic-range tradeoffs. More recently, the spreadsheet was adapted to include ADC calculations. Table 2 shows this spreadsheet with data for the 20-meter band.

The cascade chain in the spreadsheet is calculated based on formulas found in Chapter 17 of recent *Handbooks* (see Note 6). The input is at the top of the cascade, and the ADC is at the bottom. Therefore, these calculations show that for 20 meters, we have a noise figure of about 22 dB, and an *IP3* of about +35 dBm.

The lower left corner contains user-defined information for the system level, as well as for the ADC. The bandwidth is self-explanatory. The external excess noise ratio (ENR) is the expected level of the atmospheric, cosmic and man-made noise. This is discussed in Note 2, which includes a chart that I used to get my numbers. The ADC information will be discussed later, as will the calculated ADC information found in the lower right of the spreadsheet.

The calculated information in the upper right includes the MDS, dynamic range and front-end gain. The external ENR is taken into account with the external-noise dynamic range and with the noise figure shown below it. This allows you to see the dynamic range you get with a given noise level at the antenna, in addition to what you would measure in the lab. Lab measurements will always be better, unless the receiver is not sensitive enough. I did my design for a *NF* of about 6 dB less than the external noise, which causes 1 dB to be added to the noise at the input.

ADC Selection

As I mentioned earlier, this is currently the weakest link in the receiver. Any improvements in the ADC can be translated directly to a better *IP3*. This is why traditional IF-DSP receivers have used low IFs. It allows the use of 16- to 24-bit ADCs that have excellent specifications.

If one samples fast enough, though, 16 bits are not necessary, though they would be nice. It took me a while to appreciate that. By using some easily applied techniques, we can achieve excellent performance using today's high-speed 14-bit converters.

The converter I chose for this application is the Analog Devices AD6645.[7] This is a 14-bit converter designed for IF-sampling at rates of up to 80

Table 2—Dynamic Range Spreadsheet

		Stage				MDS	DR	DR including
Stage	NF	gain	OIP	Total NF	Input IP	(dBm)	(dB)	external noise (dB)
LNA / Atten	0	0	180	22.09	34.8	–117.9	101.8	97.2
RF filter #1	0	0	180	22.09	34.8			
Preamplifier #1	0	0	180	22.09	34.8	**Total gain**		**NF**
RF filter #2	1.5	–1.5	57	22.09	34.8	7		28.99
Preamplifier #2	3.5	10	44	20.59	33.3			
RF filter #3	1.5	–1.5	57	30.51	51.8			
Input *IP*	0	0	51.4	29.01	51.4			
input *NF*	29.01			29.01				

			ADC				
	External		Maximum	Sample	Noise		EXT/FE noise
BW	ENR	SNR	Signal	Frequency	Floor	NF	at ADC
(Hz)	(dB)	(dB)	(dBm)	(Hz)	(dBm)	(dB)	(dBm)
2400	28	74.5	4.8	6.50E+07	–111.02	29.01	–99.95

megasamples per second (MSPS). It provides excellent performance with analog inputs up to the lower-VHF range, which allows future expansion to 6 meters without complication.

SNR and Processing Gain

The specified signal-to-noise ratio (*SNR*) of the AD6645 is approximately 74.5 dB. The maximum input to the ADC in the configuration we'll be using (with the ADC looking back at 200 Ω) is +4.8 dBm. By itself, this doesn't provide an adequate noise floor to operate in a direct-to-digital system. This brings us to our first simple technique, *oversampling*, and its result, *processing gain*.

QEX has already introduced readers to the concept of processing gain (see Notes 4 and 5). This is how we will take advantage of the very high sampling rate. As shown in Fig 3, the total quantization-noise power is the same regardless of the sampling frequency. For higher sampling frequencies, noise power is spread over a greater frequency range. Thus, the noise power in an identical bandwidth is lower when using a greater sampling rate. The *SNR* will be improved based upon the formula:

$$processing\ gain = 10\log\left(\frac{F_s}{2BW}\right)\text{dB} \quad \text{(Eq 1)}$$

where F_s is the sampling frequency in hertz and *BW* is the final bandwidth in hertz. For example, with a sampling rate of 65 MHz and a bandwidth of 2400 Hz, the improvement in the *SNR* would be 41.3 dB.

This information is used in the spreadsheet of Table 2. In the lower left part of the spreadsheet, we enter a nominal *SNR* of 74.5 dB, a sampling frequency of 65 MHz, and a maximum input level of 4.8 dBm. The data in the lower right gives a noise floor based upon this information and the bandwidth. A value for the noise figure—29 dB in this case—is also calculated from the noise floor. Though noise figure is not normally associated with ADCs, which are voltage (not power) devices, doing so makes the job of the spreadsheet easier. Also included in the lower right of the spreadsheet is the total front-end noise power at the ADC input. I have used this primarily as a sanity check for other calculations. Notice that the front-end noise dominates, not the ADC.

The *SNR* will also be affected by both aperture jitter within the ADC and clock jitter. This is illustrated by the following formula:

$$SNR = 20\log\left(2\pi F_{in} t_{RMS}\right) \quad \text{(Eq 2)}$$

where F_{in} is the analog input frequency in Hertz and t_{RMS} is the RMS jitter in seconds. This formula gives the best possible *SNR* for a given frequency and level of jitter. Note that jitter performance that would be acceptable for 160 meters could be totally unacceptable for 10 meters. We need to maximize the performance at the highest frequency used.

The aperture jitter of the AD6645 is nominally 0.2 ps. For best *SNR* performance, we don't want the clock jitter to add appreciably to this. To accomplish this, and to allow for future upgrades to 16 bits, we'll shoot for a clock that has less than 0.1 ps of jitter. This can be translated to SSB phase noise by the formula:

$$L = 10\log\left(\frac{4\pi^2\left(\frac{t_{RMS}}{T_o}\right)^2}{f2 - f1}\right) - 3 \quad \text{(Eq 3)}$$

where *f1* and *f2* define the frequency range of interest in hertz, and T_o is the clock period in seconds. In all frequency bands of interest, we will be looking at the phase-noise floor of the oscillator. In the worst case, we'll have a 2-MHz bandwidth. Thus, to achieve less than 0.1 ps of jitter, we need the SSB phase noise to be better than –154 dBc/Hz. This will be no problem to achieve with a good crystal oscillator.

SFDR Performance

Another major concern for the ADC is the *spurious-free dynamic range*

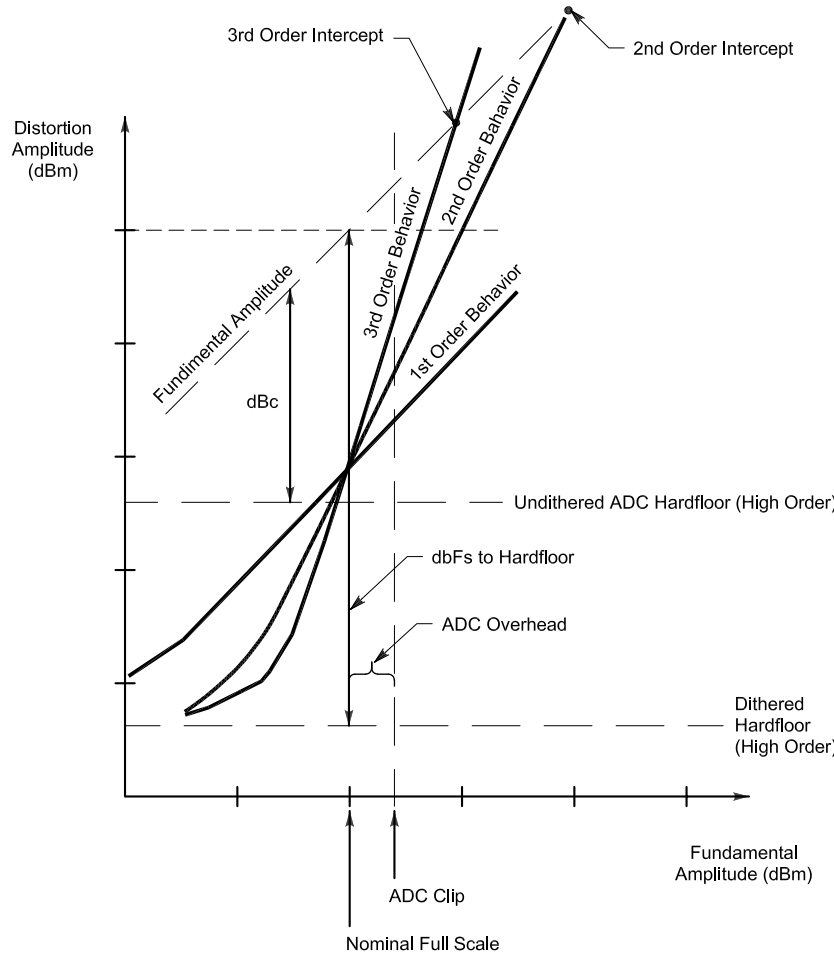

Fig 4—Distortion characteristics and an illustration of the effect of dither in reducing the distortion "hard floor."

(*SFDR*). For the AD6645, this value is approximately 100 dBFS (relative to full scale) for both single- and two-tone signals. This is very respectable performance, but it can be improved further by using our second simple technique, dither.[8, 9]

Dither is a technique that injects random noise into the ADC to overcome converter nonlinearities. I will not get into all the detailed theory behind dither here, but I will cover some basic concepts as they apply to this design. For a more detailed description, see Notes 8 and 9. In short, the *differential nonlinearity* (*DNL*) within a converter is a major source of spurs. By adding random noise, the *DNL* is "averaged out" and virtually eliminated. This reduces spurious outputs.

Fig 4 illustrates this point. It shows the familiar concept of the intercept point, but also introduces a new concept. A given converter will have a distortion "hard floor" that generally shows up as higher-order spurs. These spurs do not decrease with a reduction of the input level. We lower this hard floor by injecting dither into the ADC. The lower-order terms are also improved somewhat. The amount of improvement depends on how much the *DNL* contributes to the spurious content within a given converter. Fig 5 shows the single-tone improvement for the AD6644, the predecessor to the AD6645.[10]

Dither will do nothing to improve the spurs generated within the ADC's analog front end. These spurs will generally be lower-order terms and will therefore determine the *IP3* for the ADC. In Note 9, this is referred to as the "soft floor." Spurs generated in the analog section of the converter will decrease with a reduction of the input. Therefore, if the analog section is good enough (as in the AD6645) the ADC should not be the limiting factor in the receiver's IP3.

For the spreadsheet, I used an *IP3* value for the ADC of approximately +51 dBm. I got this value by assuming a distortion product level of –105 dBm with input tones at approximately –2 dBm (–7 dBFS). This should be achievable, given the assumption that the *DNL* limits the distortion performance, not the analog section. If not, the *IP3* could degrade by a couple of decibels.

The dither level recommended for the AD6645 is –19 dBm. This is the total noise power applied, which we will limit to a bandwidth of less than 500 kHz. Doing this minimizes the *SNR* reduction caused by the addition of dither. It also allows us to accomplish this task using a handful of inexpensive op amps, as we'll see when we get to the actual implementation.[11]

ADC Resolution

The bit-resolution requirement for the ADC created the biggest mental block for me, with regard to ADC performance issues. In this architecture, the ADC does *not* require a least-significant bit (LSB) that is smaller than the minimum signal expected. Because of the application of dither, the input signals will be spread over a vast number of quantization levels—more than 1000 of them. This means that the extra resolution is not necessary until we begin to filter out the dither signal in a later stage. It is interesting that even without injected dither, the analog parts of current 14-bit ADCs generate enough thermal noise to provide self-dither at the LSB level.

ADC Overload

In an analog system, one of the measures of receiver performance is the blocking dynamic range. This is a measure of the level of desensitization created by the presence of large signals. With an ADC, however, we are less worried about "de-sense" than with overload. If the incoming signal exceeds the ADC input range, the result is a clipped signal that can generate massive distortion. In his article (see Note 4), Doug Smith, KF6DX, used AGC ahead of the converter to prevent clipping. In his implementation, though, there was enough gain in front of the ADC so that there was little sensitivity lost if the gain were reduced in later stages. In this architecture, any gain reduction prior to the ADC results in a corresponding loss of sensitivity.

With a +4.8-dBm maximum input to the ADC and 7 dB of gain in the front end for 20 meters, the ADC will clip with about –2 dBm at the receiver input. We need a little over 17 dB of gain to achieve the desired noise figure at 10 meters, which results in a clipping level of about –13 dBm. I spent quite a bit of time convincing myself that this level would not be reached.

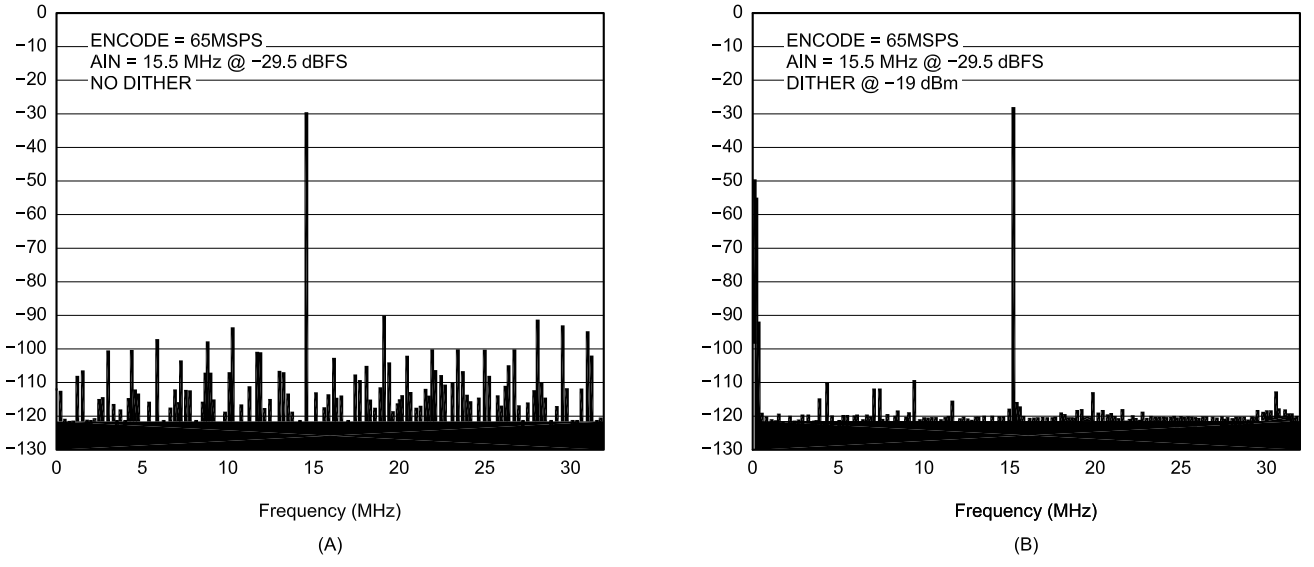

Fig 5—The effect of dither on AD6644 spurious performance.

Early on in the project, I decided that I would make this a ham-bands-only receiver. This allows me to use individual band-pass filters in the front end, which limit the energy that can reach the ADC. It also greatly improves the second-order intercept point for any receiver. To assure myself that the system wouldn't need gain control in the analog section, I did some worst-case calculations of the total power present at the ADC.

Although the probability that the signals present would be correlated is basically zero, I did calculations for both correlated (voltages add) and uncorrelated signals (powers add). I found that even if the band were jammed to the gills with S9+ *correlated* signals (highly unlikely), the clipping point would not be reached. I showed my worst case numbers to Bill Sabin, W0IYH, who termed them "virtually impossible." I then did some rough probability calculations, which indicate that I might clip for a second or so every few years. I'm willing to take that risk in exchange for a simple, clean front-end design. Therefore, all automatic gain-control functions will be performed in DSP.

Digital Down-Conversion

To be able to use the output from the ADC, we need some way to translate the data to a sampling rate that can be handled easily by the DSP. The RSP does this for us. These parts are available from a couple of sources. For my receiver, I chose the Analog Devices AD6620.[12] This part is capable of handling up to a 67 MSPS input data rate and an input path of up to 16 bits. This leaves a future upgrade path when 16-bit ADCs get fast enough.

Fig 6 shows a simplified block diagram of the RSP. We'll use the "real" mode, which takes a single input per sample period. Other modes are available for diversity reception or to allow cascading of RSPs. Take care to ensure that the clock/data setup and hold requirements are met. I'll show a simple way to do this when we discuss the implementation. The input data are split and fed to a quadrature numerically controlled oscillator (NCO). The NCO is followed by two programmable "cascaded integrator comb" (CIC) stages that perform decimation with digital filtering. The CIC stages are followed by a programmable FIR filter, with up to 256 taps, and further decimation. The data are then sent to the DSP via a 16-bit parallel or a 16-, 24- or 32-bit serial link. The incident and quadrature data words are sent in succession.

The NCO uses a 32-bit frequency tuning word. Phase and amplitude truncation to 18 bits limits the spurious level to better than –100 dBc. The largest spurs occur when the programmed frequency is an integral submultiple of half the clock rate, with the larger fractions (that is, 1/2) generating larger spurs. These spurs should be no problem for this design, as most frequencies fall outside the amateur bands; those that fall in band are higher-order (9th in 80 meters, 17th and 18th in 160 meters). Spurs can be further improved by activating phase or amplitude dither, if necessary. There is a tradeoff here, as the dither will raise the noise floor slightly.

The multiplier output is truncated to 18 bits. Using formulas given by KF6DX in part 1 of his article (see Note 4), it was determined that this truncation does not noticeably increase the noise level in the bandwidth of interest.

There are two separate CIC filters. CIC2 is a second-order filter that has a decimation ratio programmable from 2 to 16. The filter itself is fixed, based upon the decimation ratio. This type of filter is optimized to generate a minimum amount of noise, so again the noise level is not noticeably increased, even with the reduced sampling rate. CIC5 is similar to CIC2, except that it is a fifth-order filter with a programmable decimation ratio from 1 to 32. Truncation after each CIC filter is 18 bits.

Signals are then processed in a programmable FIR filter. This filter can have up to 256 taps, depending on the amount of decimation used in the CIC

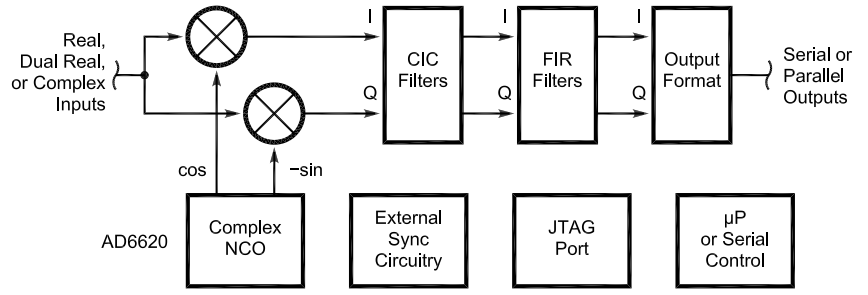

Fig 6—Simplified block diagram of the AD6620 RSP.

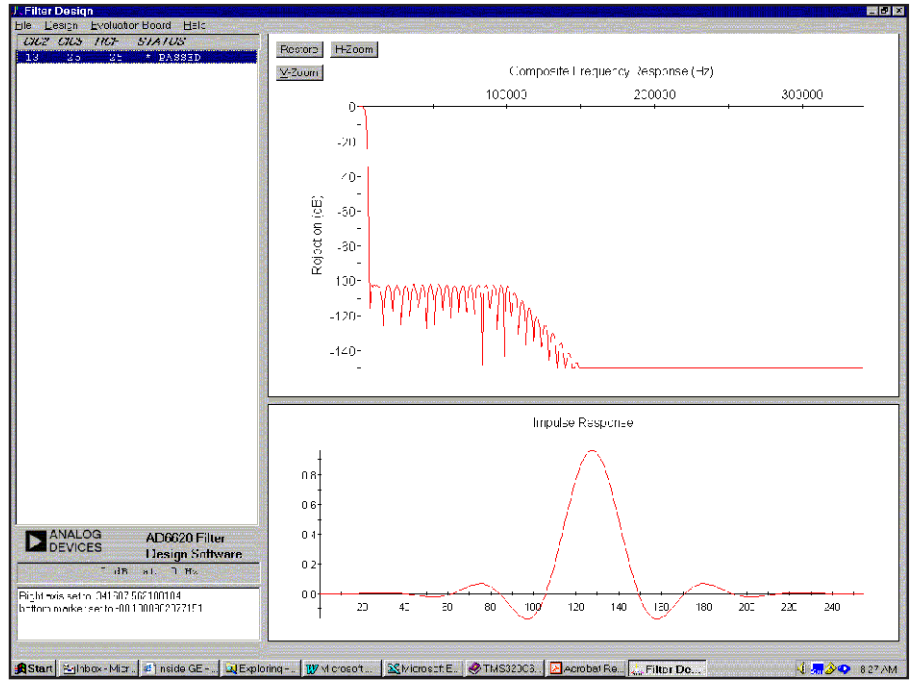

Fig 7—Sample screen from the AD6620 filter-design software.

filters. Although FIR filters generate additional truncation noise in each tap, the noise level remains insignificant by keeping 23 bits of data. Further decimation (programmable from 1 to 32) is performed before the data are formatted and sent to the DSP.

Decimation and FIR filter coefficients can be optimized using software provided by Analog Devices.[13] This software allows the user to set all necessary requirements, such as decimation ratio, alias rejection and filter response. It then gives possible solutions and allows the user to simulate them. A sample is shown in Fig 7. The upper-left window shows the possible decimation combinations (only one in this case) and whether each combination meets the requirements specified. The two right-hand windows give the frequency and impulse responses for the entire RSP system. A text file contains the FIR filter coefficients the user programs into the part.

The data from the AD6620 can be sent to the DSP in either parallel or serial format. The parallel interface is, however, limited to 16 bits. Since we have now removed the dither from the signal, we need more resolution than that. Therefore, we will use the serial interface. This interface allows data transfer in 16-, 24- and 32-bit words. We'll use the 24-bit mode, which allows all 23 data bits to be sent while maximizing efficiency. The serial port can operate at clock speeds of up to half the input clock rate, so it is no problem to get both the I and Q data words transferred between output samples.

Analog Front End

The analog front end is simplified in this architecture. No mixers are included! Although there are multiple filter copies, the only circuit blocks in the front end are the band-pass filters, a low-pass filter and the preamplifiers.

The combination of these blocks is dependent upon the band in question for two reasons. First, as has already been mentioned, the external noise is different for each band, resulting in different noise-figure requirements. Therefore, at 20 meters and below, there is one preamplifier in line. There are two preamplifiers above 20 meters.

I have also included a pair of resistive attenuators in the front end. One provides 5 dB of attenuation and the other, 10 dB. There are those who might argue that switched attenuators merely mask an inferior design. I would not dispute that argument and it may very well apply to this design. However, they give a certain amount of flexibility in optimizing noise figures for different bands. For example, the default setting for 40 meters uses the 5-dB attenuator to account for a typically higher external noise level than occurs on 20 meters. This adds 5 dB to the *IP3* on a band that can often use it. The attenuator settings can be changed easily enough, or the first preamplifier can be switched in, to adapt to changing conditions.

The other reason for differences in block use pertains to the band-pass filters. With a sampling rate of 65 MSPS, images begin to occur at 32.5 MHz. Therefore, we must deal with images as if we were using a 32.5-MHz local oscillator. Obviously, this makes image rejection a challenge for the higher bands—10 and 12 meters in particular. To address this, there are three band-pass filters for 10 and 12 meters and two band-pass filters for all other bands. An additional low-pass filter is used on all bands. The resulting analog block diagram appears in Fig 8.

I developed my band-pass filters using the procedures outlined by Mr. Sabin in his *QEX* article.[14] I used a combination of series- and shunt-coupled filters for the lower bands to provide a more symmetrical response. For the higher bands, there is no need for that because of the low-pass filter's contribution to the response.

I adapted the component values to suit my own preferences. For instance, my interests have not taken me to the upper 500 kHz or so of the 10-meter band. Therefore, this portion of the band has a few decibels of additional attenuation (> 3 dB). This buys the extra image rejection I want for the part of the band I do use. I'm always free to change the capacitor values to include this part of the band later. The inductors for these filters are the same values Bill Sabin used, as they are a known, measured quantity. He also measured the intercept point of these filters to be about +57 dBm, which appears in the dynamic-range spreadsheet.

The block that limits performance in the analog section is the preamplifier. It should have as large an *IP3* as possible. Jacob Makhinson, N6NWP, attained outstanding performance using a push-pull amplifier with "noiseless" feedback.[15] Simulating with Ansoft's *Serenade SV*,[16] I got a +50-dBm output *IP3* value with 40 mA in each of a pair of MRF5811 devices. The noise figure is not critical in this application, but it was around 2 dB.

My simulations with *ARRL Radio Designer (ARD)*, however, showed that the common-base configuration created difficulties when I sandwiched it between the band-pass filters. This is because of the low reverse isolation inherent in the amplifier design. That low reverse isolation would also tend to increase IMD (and decrease the *IP3*) because of reflections from the filters. I therefore adapted the design to a common-emitter configuration. The result is a loss of 5 to 6 dB on the intercept point, but the much greater reverse isolation ensured proper termination for the filters. The result is that I now have an *IP3* for the entire front end that is 4 to 5 dB lower than the original simulation. I considered this a reasonable tradeoff, though I would obviously prefer the higher *IP3*. On the bands using two preamplifiers, the impact is not as great, since the *IP3* of the first preamplifier (that closest to the antenna) does not have as great an effect as the second preamplifier.

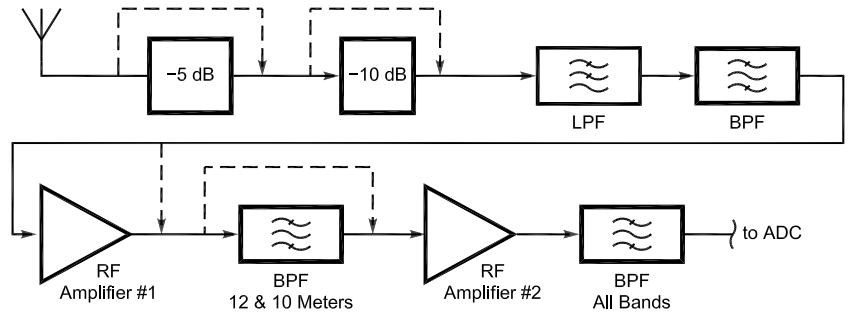

Fig 8—An analog front-end block diagram. Default settings are shown; the attenuators and RF amplifier #1 can be switched in and out as desired. RF amplifier #1 and the second band-pass filter (10 and 12 meters) are switched out during transmit. The switching is not shown for simplicity. The alternate paths are shown as dashed lines. The default settings switch in: –5 dB for 40, 17 and 15 meters, –10 dB for 160 and 80 meters, and RF amplifier #1 in for 17, 15, 12 and 10 meters.

DSP Selection

The first requirement I had for the DSP was that the serial port must accept words of at least 24 bits. As described above, this is the minimum number of bits available from the AD6620 that retains the full resolution. In my case, this led to a bit of overkill. I wanted to use a prefabricated DSP kit, since my strong suit is not designing DSP hardware. In addition, DSPs very often come in ball-grid array (BGA) packages that are not exactly designed for hand soldering.

The popular Analog Devices ADDS-218X-EZLITE kit[17] (inexpensive at $89) was my first thought, as I had one on hand. However, the serial port on this device allows only 16-bit words. I also had a DSP Starter Kit (DSK) from Texas Instruments for the TMS320C6211 available,[18] which I decided to dedicate to this project. The board includes a 16-bit codec for the audio input/output and a host-interface port to communicate with either a PC or a microcontroller. It also provides far more processing power than I will ever need for this application, at 1200 million instructions per second (MIPS). The DSP is optimized for use with *C* code, so we won't need to mess with assembly code—unless you really like it.

This DSK has been replaced with the TMDS320006711DSK, which includes the floating-point equivalent to the '6211. *C* code written for the 6211 is fully compatible with the 6711. The 6711 DSK is available from TI for $295. It includes a DSK-only version of their *Code Composer Studio*, which includes the *C* compiler, simulator and a real-time debugging utility. Analog Devices offers a kit for its 32-bit floating point DSPs (ADDS21065L-EZLITE) for $299. Analog Devices ADDS-2106X-EZKIT will also do fine and its price is $179. I am not familiar with the current Motorola kits, but that may be another option.

Exciter Design

Transmit Processing

Sixteen-bit I/Q data is sent from the DSP to the TSP. The TSP in this design is the Analog Devices AD6622.[19] This part actually has four independent channels that are summed together prior to the output, which is useful in a number of commercial applications. In this case, we'll use only one of the channels.

The part receives the I/Q data over a serial link from the DSP. The design of the AD6622 requires that the TSP be the serial master. As the DSP serial port has independent transmit/receive and can be a serial slave, this is not a problem.

The process of the TSP is very close to the RSP in reverse (see Fig 9). The data pass through a programmable FIR filter, followed by CIC5 and CIC2 filters, which in this case, provide interpolation. An NCO then translates the signal to the desired output frequency. I and Q signals are summed together and summed with any signals from the other channels (disabled in our application). An 18-bit data word is output with a sampling rate of approximately 65 MSPS. By doing the summation at 18 bits in the digital

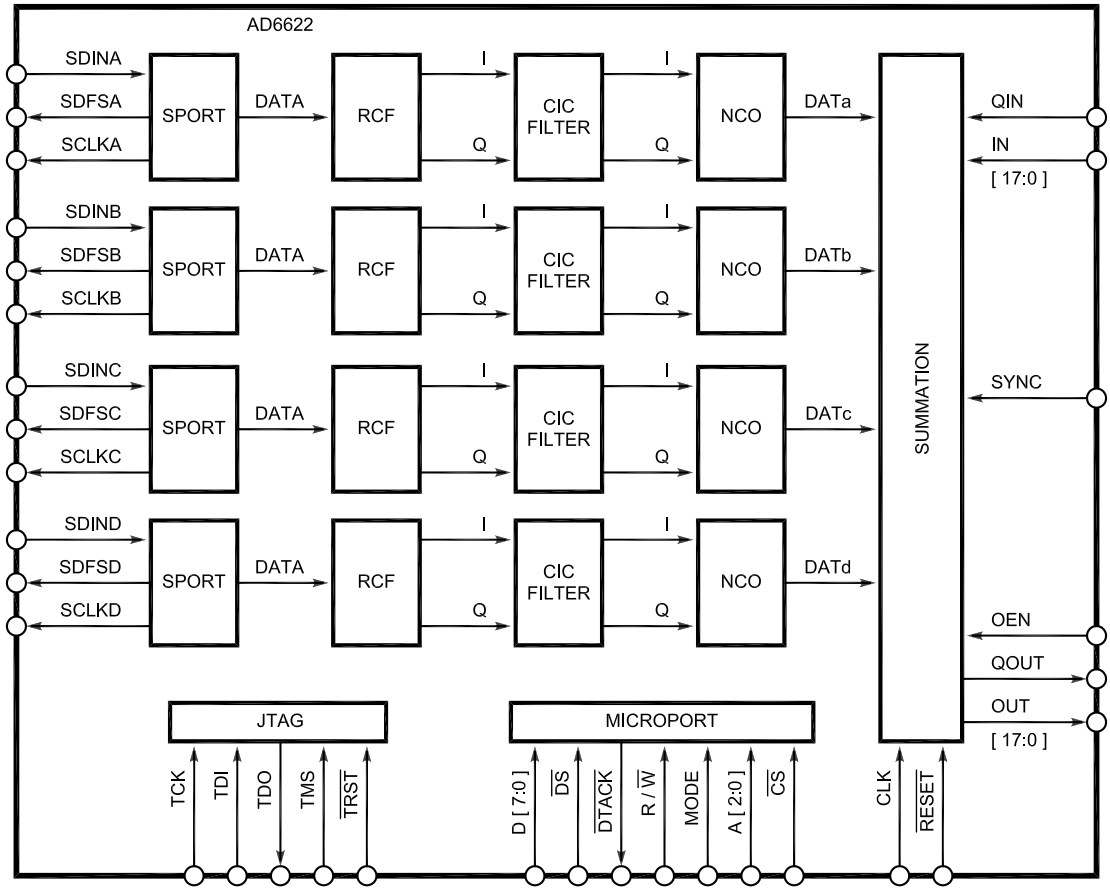

Fig 9—Simplified block diagram of the AD6622 TSP.

realm, we retain the carrier and sideband suppression we need.

Notice that the TSP only provides for summation of the I and Q signals. This requires that we look at SSB generation in a slightly different manner. Normally, the Q signal is shifted by +90° and added to the I signal for LSB, or subtracted from the I signal for USB (see Note 4). Since we cannot subtract in the TSP, we'll take a different approach for USB and shift the Q signal by –90° prior to sending it to the TSP. Try it: The math works out correctly.

Transmit DAC

Like the ADC, the transmit DAC must be able to handle RF signals. This includes the sampling rate, since we must conform to Nyquist and the analog section, which usually consists of differential current sources. The number of converter bits limits the *SNR,* and therefore we must account for it.

The best high-speed DACs readily available today are 14-bit devices, though there are some 16-bit DACs in the pipeline. These should be perfectly adequate for our design, since we're not trying to do CD-quality work. Also, since we have already summed the I and Q signals digitally, we don't need to worry about truncation effects on carrier or sideband suppression. Since 14 bits are adequate, I chose the AD9772A for this application.[20]

The AD9772A allows an input data rate of up to 160 MSPS. In addition, it provides for 2× interpolation of the input data, which eases the burden on analog filters. The *SNR* is on the order of 70 dB throughout the HF band, which is fine for our needs. Spurious performance will easily meet our requirements, especially after analog filtering. The two-tone distortion products will have no impact when compared with the nonlinearity of the power amplifier's output.

Performance will be somewhat dependent upon the quality of the clock, so we will use an oscillator similar to that of the receiver. The oscillator will operate at twice the input data rate, which will minimize noise by not using the on-board PLL. This adds an additional synthesizer, but I think the tradeoff is worthwhile to get a clean transmit signal with respect to noise.

The output compliance level—the amount of voltage that can safely be generated at the DAC output terminals—is much larger than we use here. The data sheet recommends using only part of the compliance range to minimize distortion. By loading the DAC with 50 Ω, we ensure low distortion levels and allow easy interface to existing analog circuitry.

The output of the DAC is transformed into an unbalanced output, which is buffered and sent to an amplifier with a pair of band-pass filters. The amplifier and filters are actually the ones used in the receiver. This saves space and dollars at the price of a little flexibility and insures low spurious content. The output to the driver and PA is about +15 dBm, which generates very low distortion products.

Audio Section

The audio section of the transceiver interfaces with the DSP via the TI TLC320AD535[21] codec on board the starter-kit PC board. Some key specifications of the codec are shown in Table 3. The block diagram of the audio section (both transmit and receive) is shown in Fig 10.

The sampling rate of the codec limits our audio bandwidth to a little more than 3 kHz. This is perfectly adequate for our needs. Active audio low-pass filters are used in both transmit and receive paths to prevent aliasing. Both filters have a gain of unity.

The audio from the microphone can have a very wide variation in signal level. To maximize the *SNR* of the audio input, we will need to amplify the signal until it approaches the maximum level allowed by the codec. Since this will require many different gain settings, we will use a preamplifier chip designed for use in the computer market. The SSM2166[22] provides variable gain dependent upon the input signal level. The designer specifies the desired output level and the input threshold by means of external resistors. We will set the output level to –10 dBu (0 dBu = 0.776 V RMS, 1mW at 600Ω). The additional gain needed to maximize the *SNR* will come from a second amplifier with a manual gain control.

The codec can provide ±2 V to a 60-Ω load. This output is buffered and then filtered. Volume is controlled using an audio voltage-controlled amplifier (VCA), such as the SSM2018.[23] These devices are designed for professional audio applications and are th refore overkill here, but they are inexpensive and the fidelity is excellent. For the power amplifier, I chose to use a discrete class-B amplifier,[24, 25] which will provide much more output than needed, with low distortion.

Power Chain

I specified a power output of 50 W for this project. This is adequate for my needs and suitable for driving many different tube or solid-state amplifiers if more power is needed. If you want more power, further information has been presented in *QEX*.[26] (Also see Note 2, Part 3.)

I chose to use the MRF151 FET for the PA. It has excellent gain and bandwidth characteristics, and is operated conservatively at 50 W. The amplifier also operates at a conservative 40 V to maximize component life. The driver operates in class A to provide excellent IMD characteristics. The driver gain required is fairly low, only 10 to 13 dB. The MRF151 provides the rest.

Since the amplifier is single-ended, we will use filters similar to those described by Stephensen (Note 2, Part 3). These filters are optimized for high

Table 3—CODEC Parameters

Sampling Rate	8 kHz
Maximum Input	3 V pk-pk
Input Impedance	50 kΩ
Full Scale Output	±2 V
Output Impedance	60 Ω

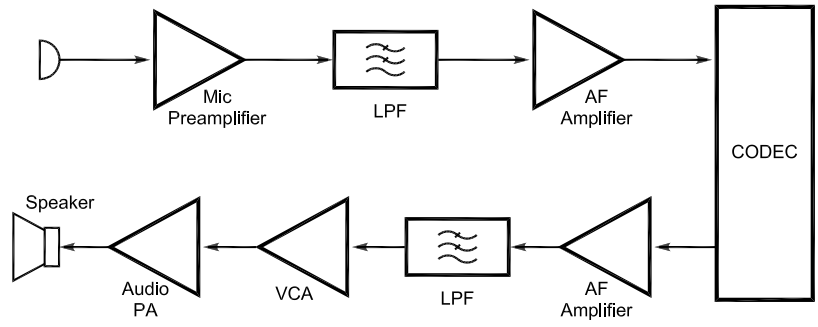

Fig 10—Audio-section block diagram.

attenuation of the second harmonic. They also provide good return-loss characteristics at the harmonic frequencies to reduce IMD.

Summary

This article has described the thought processes behind a high-performance transceiver design that interfaces directly between the digital domain and the desired operating frequency. Signal-handling issues were addressed and tradeoffs explained. The expected receive performance of the transceiver has been modeled to be better than that of commercially available radios. Later articles will cover the detailed design of the transceiver.

I would like to thank Brad Brannon, N4RGI; Bill Sabin, W0IYH; and Doug Smith, KF6DX, for their assistance during various definition stages of this project.

Notes

[1]U. Graf, DK4SX, "Performance Specifications for Amateur Receivers of the Future," *QEX*, May/Jun 1999, pp 43-49.
[2]J. Stephensen, KD6OZH, "The ATR-2000: A Homemade, High-Performance HF Transceiver," *QEX*, Part 1 Mar/Apr 2000, pp 3-15; Part 2 May/Jun 2000, pp 39-51; Part 3 Mar/Apr 2001, pp 3-8.
[3]M. Mandelkern, K5AM, "A High-Performance Homebrew Transceiver," *QEX*, Part 1 Mar/Apr 1999, pp 16-24; Part 2 Sep/Oct 1999, pp 3-8; Part 3 Nov/Dec 1999, pp 41-51; Part 4 Jan/Feb 2000, pp 47-56; Part 5 Mar/Apr 2000, pp 23-37.
[4]D. Smith, KF6DX, "Signals, Samples, and Stuff—A DSP Tutorial," *QEX*, Part 1 Mar/Apr 1998, pp 3-16; Part 2 May/Jun 1998, pp 22-37; Part 3 Jul/Aug 1998, pp 13-27; Part 4 Sep/Oct 1998, pp 19-29.
[5]B. Brannon, N4RGI, "Basics of Digital Receiver Design," *QEX*, Sep/Oct 1999, pp 36-44.
[6]P. Danzer, N1II, Editor, *The ARRL Handbook for Radio Amateurs*, (Newington, Connecticut: ARRL, 1997), pp 17.5-17.11.
[7]Analog Devices, AD6645 datasheet, Rev PrC, 12/01. At the time of writing, this part was not yet released. Samples are available. See **www.analog.com** for the datasheet and further information.
[8]B. Brannon, "Overcoming Converter Nonlinearities with Dither," AN-410, Analog Devices. This application note can be found at **www.analog.com**.
[9]Agilent Technologies, "The Dynamic Range Benefits of Large-Scale Dithered Analog to Digital Conversion in the HP89400 Series VSAs," Product Note 89400-7, 1994. Available at **www.agilent.com**.
[10]Analog Devices, AD6644 datasheet, Rev 0, 5/00.
[11]Analog Devices, *High Speed Design Techniques* (Norwood, Massachusetts: Analog Devices, 1996), pp 5.41-5.48.
[12]Analog Devices, AD6620 datasheet, Rev A, 6/01.
[13]Further information on the AD6620 filter design software can be found at **www.analog.com/techSupport/designTools/evaluationBoards/Ad6620sw1.html**. The software is a free download.
[14]W. Sabin, W0IYH, "Narrow Band-Pass Filters for HF," *QEX*, Sep/Oct 2000, pp 13-17.
[15]J. Makhinson, N6NWP, "A High Dynamic Range MF/HF Receiver Front End," *QST*, Feb 1993, pp 23-28.
[16]D. Newkirk, W9VES, "Simulating Circuits and Systems with *Serenade SV*," *QST*, Jan 2001, pp 37-43.
[17]Further information about the EZ-Kit Lite for various DSP products is available at **www.analog.com**.
[18]Information for the compatible TMS320C6711 DSK is available at **www.ti.com**.
[19]Analog Devices, AD6622 datasheet, Rev 0, 5/00.
[20]Analog Devices, AD9772A datasheet, Rev 0, 3/01.
[21]Texas Instruments, TLC320AD535 datasheet, Rev B, 2000.
[22]Analog Devices, SSM2166 datasheet, Rev A, 6/99.
[23]Analog Devices, SSM2018 datasheet, Rev A, 7/94.
[24]P. Cope, W2GOM/7, "A Class-B Audio Amplifier," *QEX*, Mar/Apr 2000, pp 45-47.
[25]D. DeMaw, W1FB, and W. Hayward, W7ZOI, *Solid State Design for the Radio Amateur*, (Newington, Connecticut: ARRL, 1977), pp 79, 137-138.
[26]W. Sabin, W0IYH, "A 100-W MOSFET HF Amplifier," *QEX*, Nov/Dec 1999, pp 31-40. □□