The Newsletter of The Kings County Radio Club

April 2018 *The Semi-Ridiculously Abridged Edition* Volume 5, Issue 4A

Minutes of the April 2018 KCRC Meeting,
April 4th, 2018

Our April "Pre-Meeting Question and Answer Session" was a very lively affair this month. Once again, no specific topic took center stage with many topics being discussed.

The monthly meeting was called to order at 7:50 PM, by our President, Mitch N2RGA. Also present at tonight's meeting were Treasurer Richard KA2KDQ, General Secretary Roy AC2GS, Joe AC2AE, Robert AB2LO, Lloyd K2JVX, Alan KD2OMG, Gene KY2MY, Simon KD2LQE, Bob KD2NVB, Al, and our new guest Darren!

Treasurer Report—Richard KA2KDQ, reported that our Treasury currently has $970.10 in assets in our bank account, with $107.18 in our Club PayPal account, for a total of $1,077.28. Since last month Zachary KK6DFJ, and Robert KD2OTP have joined our Club—our Club presently has 69 members, and 50 members have paid their 2018 year dues!

2 Meter Report—Richard KA2KDQ reported approximately a dozen check-ins to recent Nets.

10 Meter Report—Our most recent Net had Howard N2GOT as temporary Net Control operator. The Club is searching for a new Net Control Operator for this Net. Anyone with a decent 10 Meter setup with a free hour or two on Sunday morning should consider volunteering. Perfect reception or a very strong signal is not necessarily needed. The participants of the Net are available to relay messages back and forth, as needed. Please consider volunteering for this position. The Club executive committee will try to cover the Net Control Operator post until a more permanent replacement is found.

KCRC TechNet —Our Net Control Operator and Host, Roy AC2GS, reported that the TechNet is doing well. Roy once again asked for greater participation among club members. Please listen to our TechNet and PARTICIPATE!

KC2RC FusionNet—The FusionNet continues with a few non-local WIRES-X Nodes checking in.

Field Day 2018 Committee Report—James reported, in absentia, that he has acquired a vertical antenna for Field Day 2018 and was still investigating a cost effective, sturdy shelter for our operations.

Old Business: Our most recent successful VE session was March 18th, 2018. Our next VE Session is scheduled for May 6th, 2018 at 1 PM, in room 6B at Wesley House 501 Sixth Street, between 7th and 8th Avenue. The Club is always looking for new VE's to join our VE Sessions. For ANY interested individuals, please contact any Executive Member of the Club or the return email address for these emails of our Club Meeting's minutes. People took time out of their busy lives to help get you licensed – pass on the favor!

Repeater status was reported by Mitch N2RGA—Mitch is awaiting the latest controller firmware before attempting to reconnect the controller to the Repeater. Mitch has set up a streaming service between his shack and the Internet to provide both analog and Fusion Digital transmissions over a web browser dirsted to http://stream.KC2RC.com. Unfortunately the antennas at Mitch's shack are too close together and cause desensing of the streaming signal when he transmits. He is planning to either build a more directional Yagi antenna or relocate his antennas to lessen this desensing problem. If any member would be interested in helping Mitch and can climb roofs, PLEASE contact Mitch!

Work progresses on the new KCRC Club Patch—we are awaiting a proposed sample based upon our most recent design:

New Business: We recognized, with great sadness, the passing of one of our Executive Board, At Large, Howard Fink K2IGJ's passing on March 25th, 2018. Simon KD2LQE was voted unanimously to complete Howards term of office.

Our semi-annual new membership mailing drive was suspended until a further re-evaluation of its return on investment, in September.

We will release our annual membership roster to all members in good standing sometime this month. With the email of these minutes there will be included an option for our members to block their email addresses from entry onto the roster. This roster will NOT be published anywhere on the Internet and will only be sent to club members, but if anyone has a problem with their fellow club member having their email address, their wishes will be respected.

All 2017 members that have not paid their 2018 dues will be stricken from our membership roster and our mailing list.

At 8:40 PM the meeting was adjourned.

These minutes were respectfully recorded and submitted by Roy AC2GS on this day, April 4th, in the two thousandth and eighteenth year of our Lord of Propagation…

The Kings County Radio Club is at www.KC2RC.com or
www.KingsCountyRadioClub.com
KCRC is an ARRL affiliated club (see: www.ARRL.org)

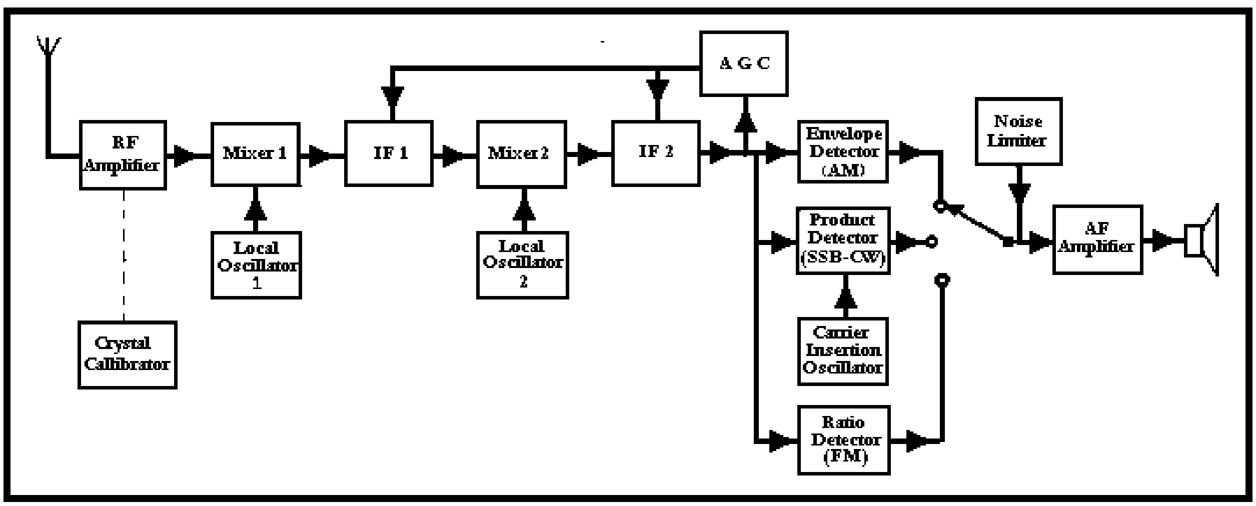
Software Defined Radio – A Technical Discussion

Let’s first return to the definition of a Software Defined Radio – there are many floating about, and more than a bit of confusion about the term. It was so confusing that those bright engineers over at the august Institute of Electronic Engineers (IEEE)) took the trouble to codify their definition of SDRs:

"Radio in which some or all of the physical layer functions are software defined."

Not “software controlled,” or “software enhanced,” but “software defined.”

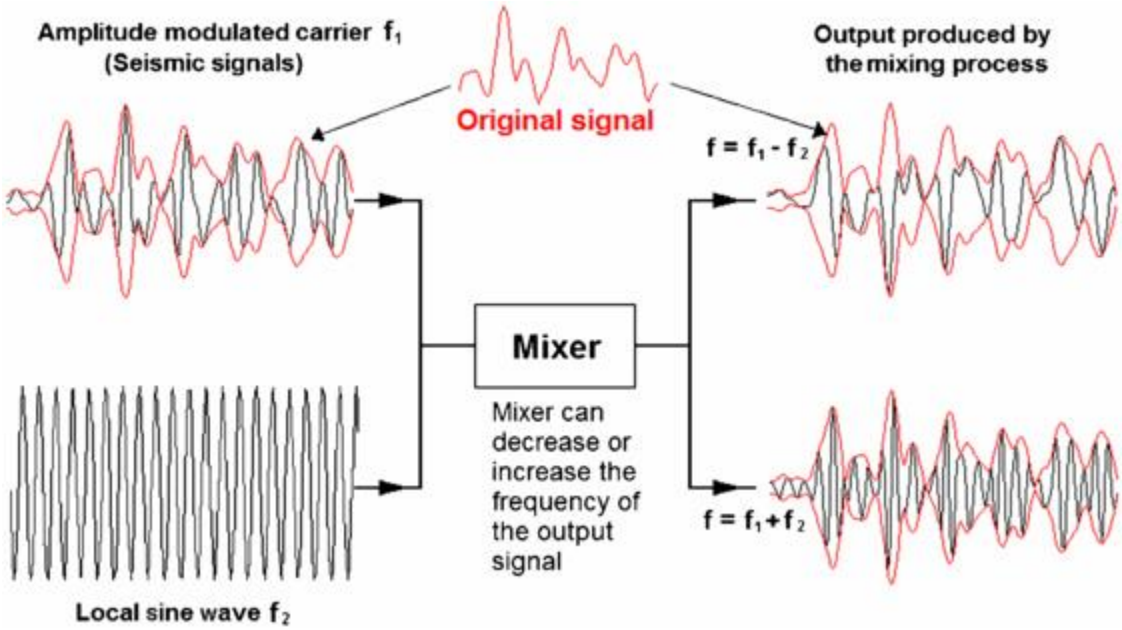
Let’s just review, for a moment, the block diagram stages of an old legacy analog radio – these pretty much define radios from 1928 all the way to your shiny new Icom IC-7851, that I am sure is a steal at the price of $12,499.95, being offered this past Holiday Season!



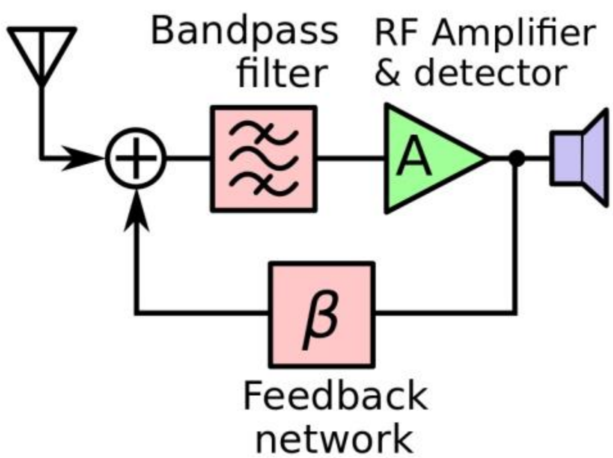
Legacy radios usually are designed similarly – the parts on the high end might be of finer quality, with better accuracy, and better structural integrity, but they are all fundamentally the same.

Radio design was revolutionized by the modified Audion Tube – the first triode vacuum tube, and the first way we figured out how to amplify an electrical signal. There were two other important advances in the design, brought forth by the brilliant if a bit eccentric, and unfortunately suicidal Edwin Armstrong. It is thanks to him that we had FM modulation/demodulation, Superheterodyning, and Superregeneracy.

Superheterodyning mixed the received signal with a local oscillator and then filtered the results, leaving a fixed Intermediate Frequency, for which any good engineer could design filters and amplifiers optimized for that one single signal frequency. You may have noticed that your old transistor radios had these 455 kHz filter cans all over their boards. 455 kHz was an extremely common frequency chosen as an intermediate frequency. If you could afford the extra components and the design costs, you could get a radio with TWO Intermediate Frequencies, one hooked up past the next, to improve your radios technical specifications even more! Or maybe even THREE IF stages!



Superregeneracy was a cute trick – you fed back part of your amplified signal into its own input! Too much and you get an oscillation, like when you use a microphone too close to the amplified speaker output of a public-address device. But if you get the amount of positive feedback just right, you get more gain for your initial received signal.



Almost every high-end radio from 1928 on is basically:

Antenna feed -> Low Noise Amplifier -> Bandpass filters -> Local Oscillator/Mixer -> Intermediate Frequency filter-> Intermediate Frequency amplifier-> another filter -> and perhaps another Local Oscillator/Mixer/Filter/IF Amplifier, if you want a dual conversion radio -> demodulating detector -> audio circuits.

Every mixer, filter, and amplifier add nonlinearity and distortion. Every local oscillator adds noise, no matter how well shielded. This is just the price that Physics demands that we pay.

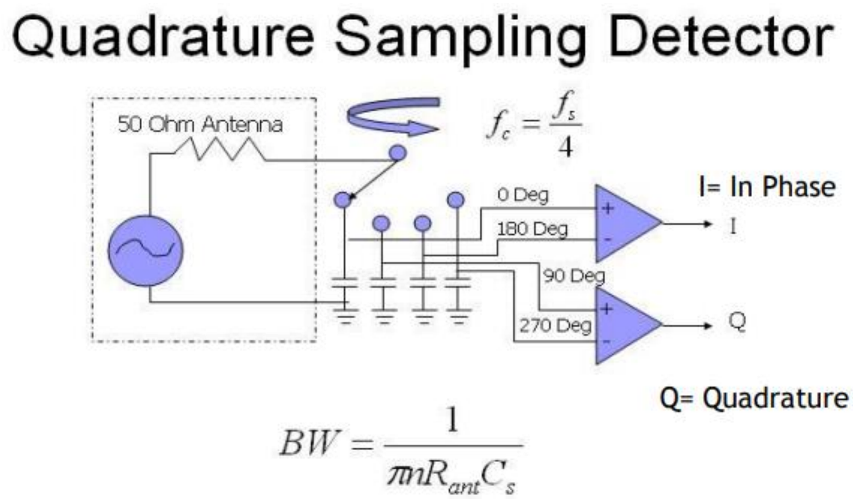
Up until recently, there was nothing more that you could really do, but recently digital signal processing chips have become available, and more importantly, they have become affordable. These are commonly used either in the audio stage or just before the demodulation detector stage to decrease the amount of noise in a given signal. Some work better than others.

Software Defined Radios are stratified into “Tiers” where Tier 0 is an old legacy analog radio that your grandfather would recognize in a heartbeat, all the way up to a mythological Tier 4 radio, where only their specifications exist on paper, waiting for technology and some bright engineers to bring them into being.

We will be discussing the less ethereal Tier 3, fully programmable SDRs.

The very “heart” of the first generation of SDRs is a device known as a Quadrature Sampling Detector, sometimes called a Tayloe Detector, after its designer. Once past the Quadrature Sampling Detector, the signal is fed into an

Analog to Digital Converter, either inside the radio or outside of it, on a PC's sound board. For reasons that you will be glad to learn, we won't get into here, right now, the simplest way to juggle all the bits around efficiently is to have a **pair** of digitalized received signals. The original, in-phase "Incidence" waveform, is usually designated "I," the second waveform is delayed by 90 degrees of its phase or one quarter of the period of the clocking speed of the Analog to Digital Converter (ADC). This quadrature delayed waveform is usually designated "Q," and that is why these devices are called Quadrature Sampling Detectors. By using the digitalized I and Q values, you can determine the input signal's amplitude, for CW, AM, and SSB detection, as well as its phase, for FM or any phase shift keying detection mode. With the information available from the digitalized I and Q signals, you can demodulate any modulation scheme imaginable, all in programmable silicon – no extra hardware required! Where you get to put the Quadrature Sampling Detector determines how much of a Software Defined Radio, you really have!



SDRs also get further classified into "generations." 1st generation SDRs were, obviously, the earliest. In first generation SDRs there was an analog mixer/local oscillator/filter stage before the signal got to the Quadrature Sampling Detector. All of the early commercially available "SDRs" were really legacy analog radios, until you got past the first IF stage – Flex 3000, 5000, 1500, Elecraft KX3! The bandwidth limitations for their panadapter and their other SDR goodies were not, necessarily due to the speed of their silicon, but were limitations of the IF bandwidth that they were given to manipulate (in the Flex 5000 that was 192 kHz). These SDRs were called "Direct Conversion SDRs." This design created a reasonably wide frequency range signal to be digitally manipulated, but it also exacerbated an ever-present problem. These radios required brawny "fat-client" PCs to do all the digital manipulations, and many PCs were not quite "up" for the job!

Some manufacturers offered built in PCs, like the Flex 5000C, but it turned out to be a band aid on a festering wound. No, the engineers would have to be more innovative than slapping one of Intel's finest and calling it a day. They came up with using something called a "Field Programmable Gate Array" (FPGA). In short, it is a piece of silicon that has its hardware re-wired through software, to make a kind of CPU optimized specifically for the type of algorithms that an SDR would need, rather than making generic spreadsheet applications, or watching a movie. All the bit fiddling could be performed in this Third Generation SDR, leaving your PC to serve as a "thin client" something that most PCs can do without really trying!

Somewhere around the 2nd or 3rd generation, the Quadrature Sampling Detector got pushed all the way back. It became incorporated into the Analog to Digital converter and Field Programmable Gate Array, which possessed the digital equivalent of a mixer/local oscillator, just after the antenna feed's analog low noise amplifier and bandpass filters, generating a quadrature delayed signal in digital form. And thus, was born the Direct Digital Sampling SDR".

So, we've touched upon the present third generation of SDR design and two of its required components a Field Programmable Gate Array, and an Analog to Digital Converter (ADC) or vice versa – ADC's work in both directions!

There are a few more components that round out the third generation SDR design. There is the generic Central Processing Unit (CPU) or a Reduced Instruction Set Chip (RISC), for the more boring control duties. There's the Digital Signal Processor (DSP), for the same reason that they are in almost every radio these days – to digitally try to mitigate background noise. Include a couple of odds and ends, some interfaces to communicate via USB, or even better, an ethernet gigabit connection, a power supply, and a few other things, and you have a third generation SDR, like the Flex 6000 series, or the Anan 200 and beyond!

There are some experts in the field that describe FOUR generations – for them the thing that defines a fourth generation SDR is its use of Direct Digital Sampling, as opposed to an analog circuit form of Quadrature Sampling Detection. I didn't want to muddy the waters, but I just want you to be aware of this discrepancy in the literature if you come across it. Whether you want to define 3rd generation SDRs as a very smart 2nd Generation SDRs and fourth Generation SDRs as possessing digital sampling, is up to you.

We shall try to explain how your signal, presented to a 3rd (or 4th?) Generation SDR's antenna input deals with this signal.

Unfortunately, we need a little of that old analog magic at first, a nice old low noise analog RF amplifier, some bandpass filters, and a switchable attenuator, so that very powerful off-frequency signals will not oversaturate our ADC and lower our receiver's sensitivity. De-sensing of the front end of a radio receiver is a widespread problem that has happened to most of us. But after that little bump, its "digital" until we get back to the analog audio output!

Next stop is the Analog to Digital Converter, followed by the Field Programmable Gate Array's software based Local Oscillator/Mixer/Filter/Quadrature generator to generate a digitized stream of 1's and 0's that define the signal that your radio is receiving. The quality of your ADC/DAC will determine many specifications of your radio. It's clock speed limitation will determine the bandwith that you will be able to evaluate at one time, and the number of bits that can be stored will determine your signal's resulting maximum dynamic range. There are tricks to get more out of your DAC/ADC, but you always have to pay Nature back in some way.

The theoretical maximum dynamic range for a 16-bit ADC is 96 db. For a 24-bit ADC, it is 144 dB, although in the real world the best you can expect is 130 dB, which is a lot better than the common dynamic range of 75-80 dB seen in a typical receiver. Old radios make up for this poor dynamic range by using their Automatic Gain Control (AGC) to shift that limited range up or down. SDRs only use AGCs to limit the audio being outputted.

As an example, the Flex 6700's DAC can decode 30 kHz to 72 MHz using a sampling rate of 245.76 Msps (million samples per second). Tricks like "aliasing," which amounts to the removal of a low pass filter before the ADC can trick your receiver into operating on what's called its "second Nyquist frequency" – this is how a Flex 6700 can operate on the 2 Meter band. There are other "tricks" that designers can employ. They can introduce "dither" into the ADC input – pseudo-random noise is injected into the input and subtracted from its output. This dithering can improve IMD (Inter-Modulation Distortion) without too much additional noise being added to the signal. A similar "trick" is to use a random function to randomize the output of the ADC in case its more regular output interacts with other regular local signals, and then converts it back into useable form inside the FPGA. Many times, these tricks are not necessary for good reception on the Ham bands, though. But they are available if needed.

When the signal gets to the FPGA, things get rather technical. First, if randomization was used earlier, the FPGA must "de-randomize" it before it does anything else. Then the FPGA uses an algorithm given the name CORDIC – it generates a mixer, a filter, and a pair of sine and cosine local oscillators in software. It is a Quadrature Sample Detector, purely in silicon bits! Since you are doing this within the silicon of your FPGA, there is no nonlinearity nor phase error that you would get with a real physical Tayloe Detector! Now, we are back to a digitalized stream of "I" and "Q" data, that we mentioned earlier. Usually, you don't need the entire spectrum of HF at one time so that you can get rid of some of it with a benefit. There is a process called "decimation" which effectively gets rid of extra, unnecessary information from the data stream. There is a price, though. It generates spurious noise, but that can be filtered out, in software. Your FPGA uses another algorithm with another lovely acronym called Cascaded Integrated Comb (CIC) filters. They are small and quick, but they are a bit "dirty" and leave a bit of a mess after themselves. We'll need a few computer simulated CIC filters to deal with the mess that the decimation caused. What to do with the mess that the CIC filters caused? Why, use another filter algorithm at the final stage – the Compensating Finite Impulse Response (CFIR) filter. A Finite Impulse Response (FIR) filter is a much more complicated, but much cleaner filter, but you still need to get the mess created from the row of CIC filters used before – that's where a Compensating FIR filter comes in. It is specifically designed to compensate for the mess that the row of CIC filters left behind!

But why go to this trouble? Well, it lessens the amount of math that you need to push all around your silicon, but also decimation has an amazing effect – by taking the digital stream and decimating it, you improve the dynamic range for the smaller frequency range that you are left with! Win-Win!

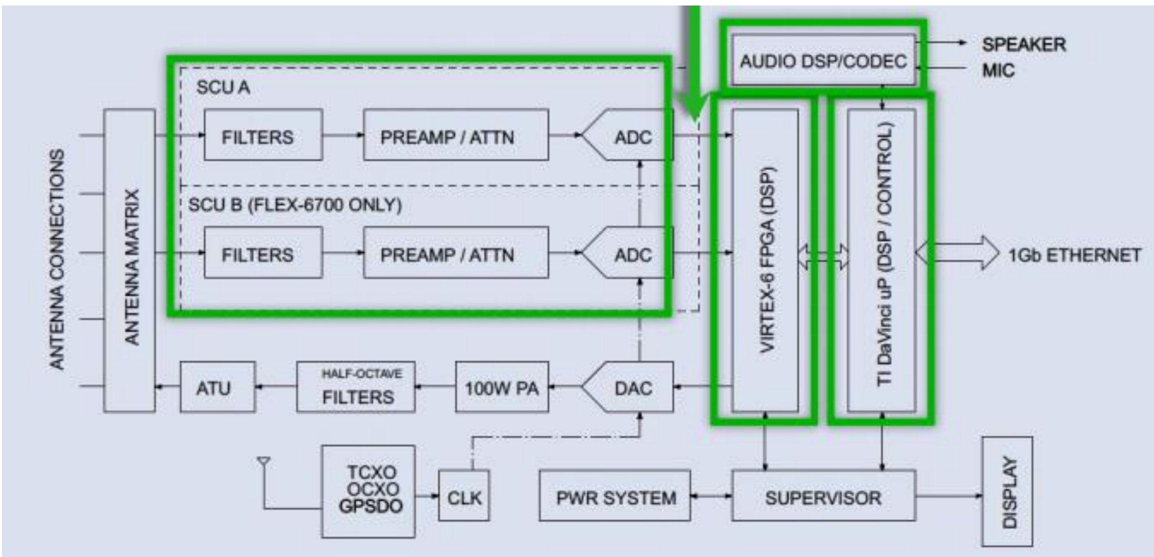
I won't bother you with all the details of how the FPGA deals with your transmitting signal, except to say that it uses another CORDIC mixer/local oscillator/filter to do "up converting" (your receiver did the opposite, "down-converting"). Your FPGA uses another algorithm called an Interpolating Cascade Integrated Comb (ICIC) filter, which does the opposite of decimation, with some loss of your dynamic range (Nature always requires pay back!).

There are other filtering algorithms in an SDR programmer's bag of tricks, like the Infinite Impulse Response (IIR) filter, which is better suited for audio frequency filtering.

There are FPGA algorithms that allow the FPGA to take the resulting decimated and filtered I/Q streams of data and juggle them into a demodulated FM, or AM, or SSB, or almost any other demodulated information stream. Further filtering, equalizing, and noise reduction can be performed within the FPGA or farmed out to a DSP chip that might be included in an SDR's design. From there it's back to analog, where the data stream runs through a DAC and an audio amplifier. That CPU or RISC is sitting in the background, making sure that your connection with your control surface is going fine and handling the I/O stream going into and out of your radio! An ethernet connection, preferably a gigabit ethernet cable will be a thick enough pipe to share as much information as your PC will ever require.

Where the receiver side uses "Digital Down Conversion," the SDR's transmitter uses "Digital Up Conversion," which allows "pre-correction" of any nonlinearity detected, to lower the resulting IMD amount even more! There are just a lot more "tricks" that you can employ in the digital domain, than in the good old analog domain!

What does all this look like? Well here is a block diagram of what's going on under the hood of a Flex Radio 6700:



With analog radios, the price you pay for clever tricks is additional components that introduce nonlinear distortions, or more noise, or BOTH. With SDRs the price one pays, other than the cost of the fancy silicon and using up too many CPU cycles is something called "latency." Latency is the delay introduced by the processing time required for digital devices to do their magic. All things being equal, filters with narrower skirts and little to no ringing artifact add milliseconds of processing delay from that signal being received, to you hearing it for yourself. This latency delay isn't just a problem with SDRs, DSPs have caused this long before SDRs were around. Algorithms with more finesse, or just more powerful, albeit more expensive silicon, are possible solutions. It will all get better, cheaper, and smaller over time. That isn't the case for "boat anchors."

So, can you take these notes that you've scribbled down and build your own SDR? I seriously doubt it, but I hope that you got a sense of what was going on under the hood of this new, and exciting technology, that I believe will be the face of radio technology in the future.

73,

Roy AC2GS