Demystifying software radio

Looking for a "magic bullet" two-way radio that covers all frequency bands, all channel bandwidths, all modulation techniques and all control and signaling protocols? Look no more—it's *software-defined radio.*

By Stephen Bartlett

Taken to its logical extreme, a software-defined radio loaded with every imaginable software—plus frequency coverage and a tiny power pack worthy of a science fiction novel—could be used in any mode, on any frequency, anywhere in the world.

In a software-defined radio, transmitter modulation is generated or defined by a computer, and the receiver uses a computer to recover signal intelligence. Need interoperability among public safety agencies regardless of frequency band and standard or proprietary protocol? Need compatibility, wide-area coverage and data features for business users? That's some of the promise—taken to the extreme, mind you—of software-defined radio.

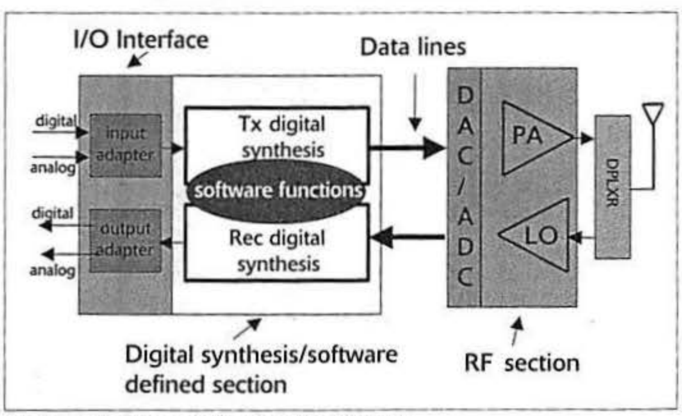

Figure 1. Software-defined radio concept.

Public safety agencies should watch software-defined radio development for its potential to replace expensive network-based interoperability often required for multiple-agency emergency response. Also, it offers first responders the potential to save money on mobile and portable equipment replacement through its compatibility with legacy and future systems.

Commercial mobile radio service providers, as well as business and industrial end-users, can benefit from reduced network management costs, more competitive features and the ability to form wide-area coverage alliances.

If you've recently signed up for the latest cellular or PCS mobile wireless phone services, you've probably had services with choices from: coverage area, free minutes, email service, Web surfing, shoe shining, you name it. Whether you chose all of these, or only a few, you most likely opted for the best coverage you could get for the best price. If so, the phone you now own is one example of a software-defined radio—a programmable, dual-mode (analog/digital), multiband (cellular/PCS) wireless device.

Software-defined radio covers several types of technology and can mean different things. To simplify, three categories can be used to classify SDR technology: multistandard terminals, reconfigurable radios and the flexible architecture radio.

Multistandard terminal — Although not a pure software radio, the MST is a programmable radio (generally a base station terminal) capable of operating on many air interface standards. These terminals support wider roaming areas and provide mobile users with backward- and forward-compatibility as wireless technology evolves.

Reconfigurable radio — Wireless devices with software and firmware that can be reconfigured through programming hardware, including modern dual-mode, tri-band phones and many modern digital two-way radios, fit this category.

Flexible architecture radio — All characteristics of this radio are programmable. An ideal FAR would be able to configure the number of up and down conversion stages, intermediate frequencies, filter bandwidths, modulation and waveforms, and RF bands. This is essentially the Holy Grail—the "one radio fits all"—of software radio. We're definitely not there yet.

The SDR technology vision is to make common hardware to perform many functions and operate with numerous protocols defined by resident software that can be changed when different functions, features or protocols are required. This can reduce costs with less maintenance and easier upgrading.

Fixed terminals, or base stations, normally are restricted to assigned frequencies and protocols for specific coverage areas. Mobile terminals and handsets require more flexibility to roam between coverage areas, with different frequencies and protocols.

Bartlett is a wireless systems engineer and consultant for terrestrial and space applications. His email address is *radio.bartlett@verizon.net.*

The software radio can be divided into three sections (see Figure 1): input and output interface, digital synthesis and software-defined section, and RF section. Designers want to keep the information signal digital until it is absolutely necessary to convert to analog. Ideally, the signal stays digital right up to the antenna, where transmit and receive RF stages perform signal conversions associated with RF amplification.

One way to do this in base stations is to string fiber optic transmission line up the tower into the RF "black box" attached to the antenna array. This method would promote the use of more efficient, lower-power, lower-temperature RF circuitry. Without coaxial cable signal losses, less RF power would be necessary for transmission and low-gain, low-noise amplifiers would give satisfactory reception. System downtime and repair or replacement costs caused by moisture that penetrates coax would also be eliminated.

Digital hardware limits — Several technologies must mature before the ideal FAR can be made. The most critical need is for ultra-high-speed digital hardware that can host software with low latency.

When the SDR receiver samples signals digitally to convert them from analog to digital (ADC), or vice-versa (DAC), the sample rate must be at least *twice* as fast as the highest frequency or bandwidth sampled—so says the Nyquist Theorem. The capability to filter the signal in the minimum acceptable bandwidth and the availability of speed-compatible converters (ADC and DAC) dictate where and how sampling is done.

Direct IF sampling

Another approach, direct IF sampling, samples the incoming received RF signal at a rate that is twice the information bandwidth frequency. This approach is also referred to as *undersampling*, and is well-suited for speed-limited devices. Undersampling is an especially clever approach for detecting broad-bandwidth signals (high-speed data and video), but it has limited use in typical, very narrow bandwidth applications (low-speed data and voice). Direct IF sampling can eliminate much of the costly analog hardware needed for demodulation followed by lowpass filters to feed the baseband sampling circuit.

Most common ADC technology limits the rate of passband and direct receiver sampling to an upper bound of 30MHz—about 60 million-samples per second (Msps). Manufacturers of newer high-speed ADCs are quoting sampling rates up to 300Msps or greater, which could sample up to the middle of the VHF band, about

150MHz. For higher frequencies, the RF signal must be down-converted to a lower frequency.

For now, software-based radios need special hardware for up and down IF conversion, filters and IF amplifiers because of the limitations of high-speed digital electronics. With densely packed integrated circuits, placing enough gates requires unrealistic power and size growth to accommodate the ideal FAR. Most SDRs include high-density application-specific integrated circuits (ASICs) and field-programmable gate arrays (FPGAs). But even these devices are being stressed by the 3G cellular technologies pushing for release. Although these new 3G technologies have been delayed for several reasons, foremost is a lack of practical hardware.

New hybrid digital signal processors, *associative string processors*, incorporate large numbers of specialized processors. Several manufacturers claim to have such devices ready for 3G systems elsewhere in the world.

Software latency – Software will provide ultimate SDR flexibility. The more processors in the hardware, the more software can be a part of the radio—for a price.

Processors generally call for service requests from specific software objects. Making these calls internal to a processor, or externally from another processor in the radio, takes valuable run time. Although internal processor calls have delays in tens of microseconds, external calls may have throughput delays on the order of milliseconds.

Millisecond delays are not usually an issue for typical data traffic, but such delays can be a serious problem for real-time communications such as emergency voice calls. As more features are programmed into the FAR—features are always added—these delays will grow. Delays are not usually an issue for typical data traffic, but can be a serious problem for real-time communications.

One solution is to move as many of the unchanging FAR functions as possible into hardware and firmware. Then, fewer software functions requiring service requests will need to be accommodated. Even so, as the newer, high-speed, processor-dense hardware becomes available, these systems will become feature-rich, requiring software latency to be closely managed during development to prevent uncontrolled latency growth—a computer evolution lesson learned.

Software radio has passed its newborn stage of development, but it still has a long journey ahead. We will watch it grow up as new commercial wireless systems are deployed. Maybe it's time to take some refresher software courses? ■