PROMISE, REALITY, AND LIMITATIONS OF SOFTWARE-DEFINED RADIOS

Lawrence Goeller and Patricia Bronson

The Problem

The goal of using software-defined radios to provide interoperable communications equipment across U.S. military forces has proven to be technologically challenging to implement.

The vision of an SDR—a communications device that can synthesize any waveform, including new waveforms, solely by changes to the software—has been, and remains, highly attractive... but, it seems evident, we are not there yet.

A long-standing Department of Defense goal has been to achieve interoperability. Some of the most challenging problems in this area involve the acquisition of interoperable communications equipment. Since the Services depend on radios that have been acquired over decades, it is infeasible to start over and replace all of them with new, more interoperable systems; the transition would have to be gradual.

The Ups and Downs of the Joint Tactical Radio System (JTRS)

Considering the constraints, and given the rapid progress in the 1980s and 1990s in powerful, inexpensive Personal Computer (PC) technology, one way forward seemed especially promising: if radios could migrate from the traditional mixer-based modulation technique to one based on a programmable microprocessor, the *waveforms* could be loaded and run on the underlying hardware much like word processing and spreadsheet programs were run on a PC (Figure 1). Such devices were referred to as *software-defined radios* (SDRs). These new SDRs needed to be, first and foremost, backward-compatible with the existing, fielded radios that each Service had previously purchased. Second, they had to

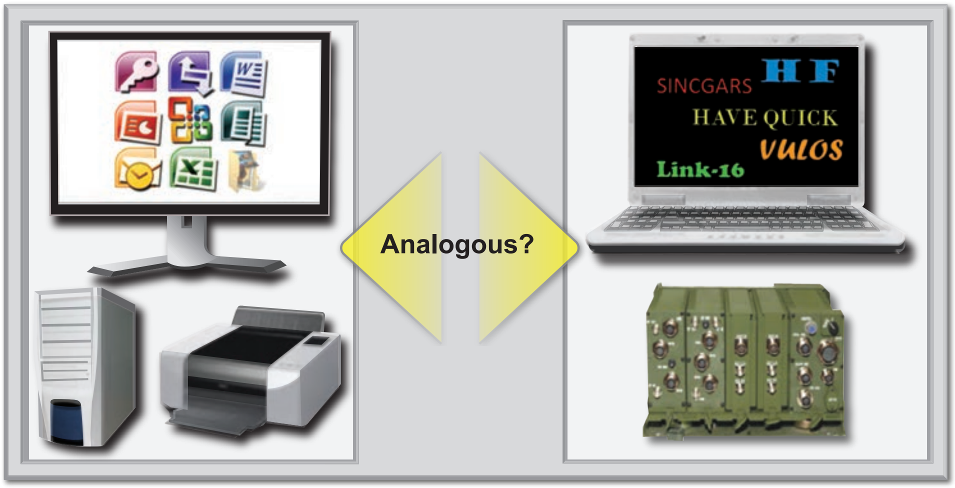

Figure 1. SDR Envisioned as a PC Running "Waveforms" Instead of Applications

be programmable to “run” at least some of the waveforms of different Services, allowing interoperability without the presumed excessive cost for additional dedicated circuitry for each additional waveform. Third, this approach offered the promise that new and better waveforms could be readily ported to SDR equipment after it had been acquired and fielded. In 1997, the Joint Tactical Radio System (JTRS) SDR program was born, and the Office of the Secretary of Defense directed that, after a certain date, the Services would not be permitted to acquire any non-JTRS radio without a waiver.

Unfortunately, the JTRS program has struggled. Despite a number of program changes, requirements easing, management restructures, and the investment of more than a billion dollars, the Ground Mobile Radio (GMR) version of the JTRS radio was terminated in 2011 after declaring a Nunn-McCurdy breach.[1] The Airborne Maritime Fixed (AMF) JTRS prime contract was ended a year later, with no fielded hardware produced. The Handheld, Manpack, and Small Form Fit (HMS) JTRS and Multifunctional Information Distribution System (MIDS) JTRS programs have produced some hardware as of this writing, and other parts of the program are continuing, but, on the whole, the program has been a disappointment. The question is: are there underlying technical issues that fundamentally preclude success, or can the promise of the SDR still be met with some combination of newer technology and different management? This is the question that the Office of the Under Secretary of Defense for Acquisition, Technology and Logistics’ Performance Assessments and Root Cause Analyses (PARCA) organization asked IDA to investigate.

The Basics of Radios and Waveforms

With few exceptions, the function of any radio is to use a baseband signal to modulate a carrier wave and then amplify and transmit the resulting signal—and then reverse the process at the receiving end. The baseband signal could be analog or digital, and the carrier wave can be at almost any frequency. Since the modulation/demodulation process has traditionally been performed via non-linear mixing and amplification of the baseband and carrier signals, such devices are sometimes called *mixers*. Other operations may be performed on the baseband signal before modulation (such as channel encoding, if the signal is digital) and after modulation (such as multiplexing, or combining, this signal with others transmitted at the same time). Collectively, the combination of channel encoding, modulation, multiplexing, and other processes is referred to as the *waveform.* The waveform may be thought of as the information the receiver needs to know about the transmitted signal in order to correctly recover the baseband information.

At the heart of most modern radios is the replacement of the mixer, and often other components, with some sort of digital processing device.

[1] The Nunn–McCurdy Amendment requires the Secretary of Defense to notify the Congress if the cost per unit of a program grows more than 15 percent beyond what was originally estimated and calls for the termination of programs with total cost growth greater than 25 percent.

This is a revolutionary change; the task of modulating a carrier has been changed to "synthesizing" what the modulated signal *would have looked like* if conventional hardware had been used via what is, in practice, a digital signal processor. (Both mixer-based and microprocessor-based radios are capable of transmitting and receiving either analog or digital waveforms.) These devices are not necessarily SDRs, however; in many designs, the digital logic circuit synthesizes only one or a few waveforms, and these cannot be modified or added to, once fabricated. For a logic-based radio to be considered an SDR, the waveforms it synthesizes must be controlled externally by a software program; hence, the name.

In the 1980s and early 1990s, it was envisioned that a so-called general purpose processor (GPP), such as an Intel Pentium, would be able to perform the waveform synthesis. It rapidly became apparent, however, that it could not, in practice, support such tasks. To make a long story short, GPPs were found to be unsuitable for the real-time and multiplication-intensive requirements associated with signal processing. Even in PCs, such operations are performed by digital signal processors (DSPs) located in math coprocessors or graphics cards. Although many military radio systems were developed that synthesize waveforms via DSP-based circuits, DSPs are not nearly as flexible as GPPs. They are designed to perform a specific, limited set of operations very quickly, but it is in general difficult to add new algorithms to them after fabrication. This constrains their utility to the SDR vision. This gap between the flexibility of a GPP and the operational performance of a DSP was, to many, an indication that synthesizing waveforms differed *technically* from running applications. Nonetheless, the attractiveness of the SDR concept remained.

Field Programmable Gate Arrays

The Defense Advanced Research Projects Agency (DARPA) came up with a compromise solution: the use of hardware devices called field programmable gate arrays (FPGAs) to synthesize the waveforms. FPGAs, a new technology at the time, are a third kind of processing architecture (although like both GPPs and DSPs, they are commonly composed of Complementary Metal-Oxide Semiconductor (CMOS) transistor technology). An FPGA is composed of a large number of small identical logic elements, each typically consisting of a small lookup table and some memory. These logic blocks do not connect directly to each other as in a DSP, but rather to one or more of a series of parallel wires called *routing channels* that run around all sides of each block (see Figure 2). Connections between the internal logic of each block and the wires of each routing channel are controlled by transistors that can be opened or closed on command; there are more of these transistors in each region where horizontal and vertical routing channels cross each other. By opening and closing selected transistors, the small logic blocks can be connected in such a way that they can emulate virtually any other logic element, from a simple digital logic gate to a block of hardware memory to a multiplier or even a small microprocessor. Best of all, these connecting transistors can be commanded by a software program.

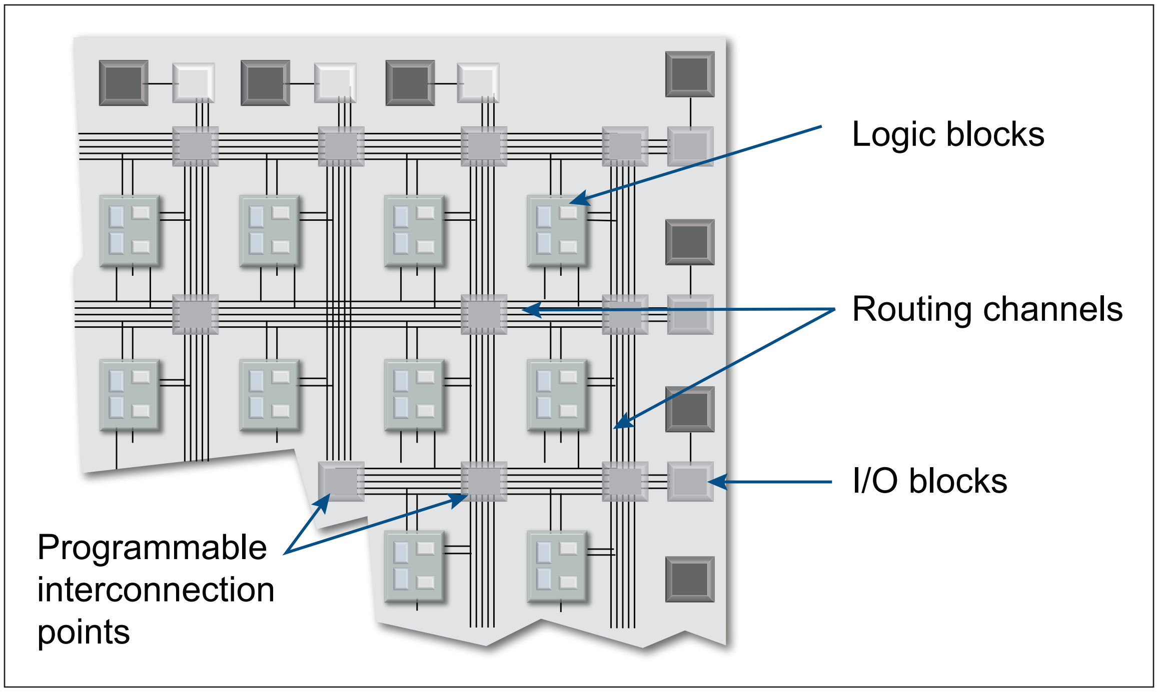

Figure 2. Notional FPGA Logic Architecture

Software packages called *hardware description languages* (HDLs) that look a lot like C or Ada could be used to "write circuits" on a given FPGA; this code could then, in principle, be recompiled to operate on different FPGAs, thus effectively offering the ability to port waveforms from one device to another. That is, the FPGA offers the SDR community a technology with comparable performance to a DSP, and yet is reprogrammable—perhaps more accurately, reconfigurable—like a GPP. After DARPA's SpeakEasy program was considered to be a great success, the FPGA-based JTRS program was initiated.

Unfortunately, it quickly became apparent—to the engineers, if not the larger acquisition community—that the "sea of configurable logic blocks" approach was also technically inadequate. FPGAs require many times the number of transistors to perform the same function as a dedicated DSP, and each of these transistors draws electrical power and produces heat that must be dissipated. FPGA-based SDRs were found to draw a large amount of power—a problem for battery-powered tactical radios—and produced a large amount of heat. At some point, it also became apparent that, despite the superficial similarity of HDL languages to portable, high-level languages like C, they were in fact quite different. The portability of GPP-based languages depends on a certain similarity of the architecture in all cases and on the ability of the compilers to find some way, if not the optimal way, of implementing each program statement. HDL compilers, by their nature, are much more dependent on the underlying hardware. The JTRS community responded by trying to develop an intermediate layer between the waveform software and the FPGA-based hardware called Software Communications Architecture (SCA)—essentially an operating system and middleware. This was uncharted territory. When coupled with the real-time constraints of signal processing, it became clear that the amount of effort to port a waveform from one device to another was significantly greater than hoped.

The technology continued to evolve, however. New generations of devices that were still called FPGAs became available, but these new devices not only were larger in terms of the number of configurable logic blocks they contained, they also contained many pieces of intellectual property (IP). These IP elements were composed of anything from dedicated blocks of fast memory to dedicated hardware multipliers to microprocessors (see Figure 3). The "catch" is that, while these IP elements run faster and draw less power than strings of configured logic blocks, they are themselves essentially DSP elements, with the corresponding lack of flexibility. In fact, the trend in FPGA design has been for each vendor to create a wide diversity of products, each aiming at a different specialized market. And therein lies the problem: the key idea of the SDR is to rely on a single underlying card to generate any one of dozens of different waveforms, including some not yet invented. What we are finding is that FPGAs are not, apparently, a one-size-fits-all technology; in fact, they are almost the opposite.

But what about the flexibility-vs.-power consumption trade space of the newer generations of FPGAs, sometimes referred to as "System on a Chip" or SOC? Alas, the JTRS program tried these as well; the AMF JTRS design, when it ended, was based on a Xilinx Virtex 5 FPGA with hundreds of embedded IP elements. Although a number of different waveforms were

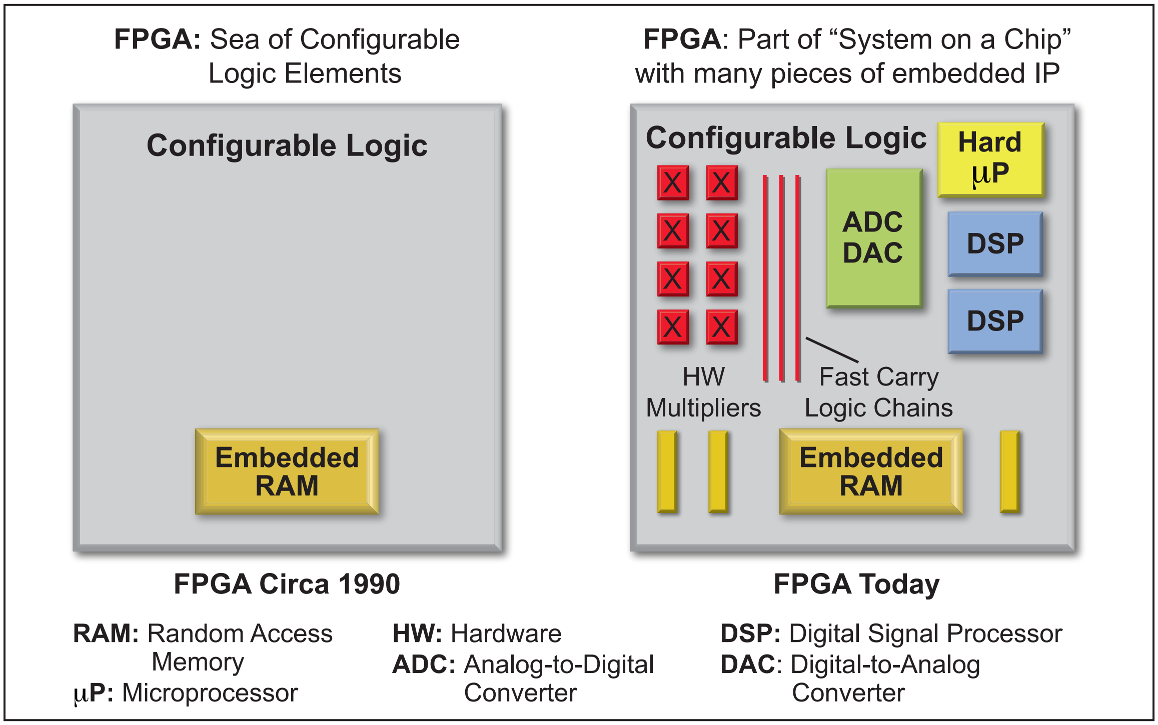

Figure 3. FPGA Architectures, Then and Now

demonstrated on the development hardware, the power consumption remained large, and porting waveforms remained challenging.

What about the future? As logic element sizes shrink, both the power that the transistors require to change states (the so-called *active power*) and the time it takes them to do it decrease. Newer designs come with clever ways to remove the clock signal from transistors that are not in use. There are even new manufacturing technologies in which the transistors are grown "vertically" on the substrate as opposed to the planar approach of CMOS today; this offers the opportunity to lower power consumption even further. Can these improvements lead to a viable approach to the SDR? Perhaps, but every advance in this trade space seems to come with additional constraints. For example, as transistors get smaller, the active power consumption goes down, but the so-called static power goes up; the transistors essentially "leak" as long as they are powered up. Further, it appears that there is a tradeoff between this static power loss in small transistors and switching speed. It must also be noted that none of these new technological approaches addresses the problem of waveform portability, which is a primary driver in the preference for an SDR over, for example, a radio containing a different DSP (or different parts of a larger DSP) for each supported waveform.

Conclusion

The vision of an SDR—a communications device that can synthesize any waveform, including new waveforms, solely by changes to the software—has been, and remains, highly attractive. Although there have been some limited successes in the decade and a half since the initiation of the JTRS program, it seems clear that the task of creating an SDR that is both power-efficient and flexible enough to support new waveforms remains challenging. This is in part due to technical reasons: specifically, the difficulty of producing hardware that is both powerful enough to perform signal processing tasks in real time *and* flexible enough to support growth to future, undetermined waveforms. The original vision of the SDR might be technically achievable, but, it seems evident, we are not there yet.

Dr. Goeller is a Research Staff Member in IDA's Cost Analysis and Research Division. He holds a doctorate in physics from Rice University.

Dr. Bronson is a Research Staff Member in IDA's Cost Analysis and Research Division. She holds a doctorate in applied physics from Old Dominion University.