Meeting Software Defined Radio cost and power targets: Making SDR feasible

By Manuel Uhm and Jean Belzile

The vision of software reconfigurable radios is finally a reality, but implementations are not as efficient as they could be. Although the radio itself can be programmed to realize multiple waveforms for joint service interoperability, current implementations require redundant hardware for multiple channels.

A better approach is to use higher performing programmable logic that can be partially reconfigured in-system. This shared resources method allows not only the radio to implement multiple waveforms, but eliminates redundant per-channel hardware. Partially reconfigurable FPGAs will save space, weight, power, and cost.

Today, Software-Defined Radios (SDRs) are becoming a reality in the military through programs such as the Joint Tactical Radio System (JTRS). However, two critical issues continue to limit the practical deployment of SDR: power and cost. Specifically, current implementations of SDRs, including JTRS Cluster 1 radios, consume more power than is desirable. This results in lower-battery life, as well as excessive thermal dissipation. Furthermore, they are still too expensive to make widespread deployment cost effective.

In order to move forward with successful deployment, these are issues that must be addressed by all SDR programs, including military and commercial. Fortunately, the technology is now available to address these issues in a significant manner by *decreasing* the number of components while still providing the necessary functionality. What is this magic technology? It is the latest generation of high-density Field Programmable Gate Arrays (FPGAs) that allow in-system partial reconfiguration. Even though the technology is applied in an unclassified SDR modem, it is worth noting that it's also applicable to other critical subsystems of an SDR, such as the RF front end, I/O, and crypto.

Dedicated resources vs. shared resources

The current model for implementing an SDR modem is known as a *dedicated resources* model. It is called dedicated resources because there is a set of processing resources that is dedicated to a radio channel where each channel typically represents one type of radio waveform, such as Single-Channel Ground and Airborne Radio System (SINCGARS). In this case, the processing resources consist of an A/D, D/A, FPGA, Digital Signal Processor (DSP), and General-Purpose Processor (GPP). In order to implement an N-channel radio, N sets of processing resources are required. This is illustrated in Figure 1 for a four-channel SDR modem supporting a Software Communications Architecture Core Framework (SCA CF), as mandated for JTRS. Current four-channel SCA-enabled SDR modem implementation requires a dedicated set of hardware for each channel (Figure 1). The more channels the SDR must support, the more hardware it contains. This adversely affects power consumption and cost.

From a functional perspective, this model of dedicated resources for each channel has been proven to work well. However, it is an inefficient usage of the processing resources, resulting in excess power consumption and cost. For example, with regard to cost in this model, the subsystem level cost of the modem scales linearly with the number of channels being supported. However, from a parts cost perspective, parts for all channels of the radio must be selected for the worst case scenario; that is, the processing resources must be able to support the largest waveform (the Wideband Networking Waveform for Cluster 1 radios), such that if only a small waveform like SINCGARS is instantiated, most of that channel's processing resources are not utilized.

This has a significant impact on driving up the cost of the modem. Obviously, the problem gets exacerbated as one scales the model further. The JTRS Cluster for Airborne, Maritime, and Fixed (JTRS AMF) installations, for example, requires some radios to support eight channels. This also has an obvious impact on power consumption. Even if all the channels are not being utilized, the processing resources are still drawing some amount of power.

A more efficient model for an SDR modem is referred to as a *shared resources* model. Unlike the dedicated resources model,

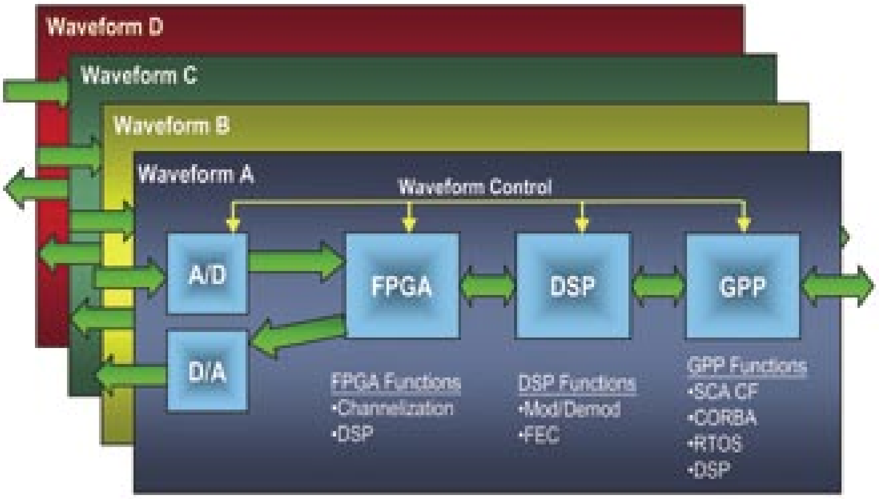

Figure 1

this architecture offers the capability to support multiple waveforms across a single set of processing resources, allowing for much more efficient usage of the resources. The number of waveforms that can be supported is a function of the size of the waveform and the size of the available processing resources. A multichannel SCA-enabled SDR modem using shared resources rather than dedicated resources is shown in Figure 2. In this instance, five channels supporting one wideband waveform and four narrowband waveforms have been implemented, while reducing the part count from 20 to a mere four components, (each functional block, as shown.) Implementation of this technology can result in a production cost and power consumption that is two to three times lower.

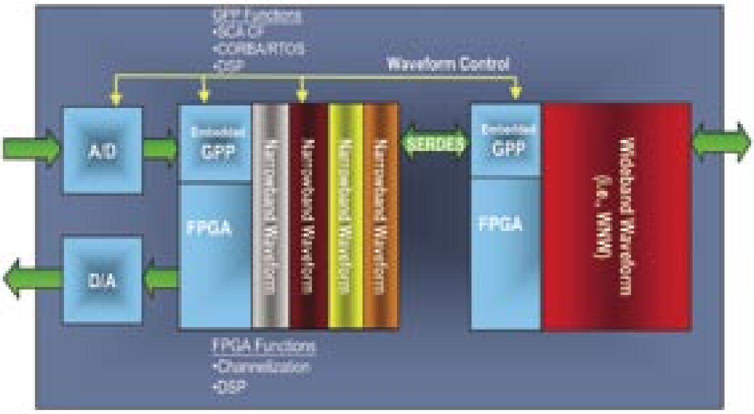

Figure 2

Also notice that in this architecture, the GPP from the dedicated resources model that manages the modem infrastructure and operating environment (which is, POSIX-compliant RTOS, CORBA ORB, SCA CF) has been integrated into the FPGA, and the FPGA is doing all the heavy digital signal processing. The embedded GPP is also a natural fit for the light signal processing, such as synchronization loop control, and the upper protocol layers such as link and network layers. Essentially, the FPGA is an SCA-enabled System-On-a-Chip (SoC). This contributes to the cost and power savings, beyond just using the shared resources model, by removing two parts per channel card: the discrete GPP and the DSP.

It is also worth noting that both models illustrated here are using 100-percent commercially available components. In the shared resources model, the SoC FPGA is a mid- to large-sized Xilinx Virtex-II PRO FPGA with an embedded IBM 405 PowerPC core, but could be a next-generation Virtex-4 FX FPGA, also with an embedded IBM 405 core.

Enabling technology

The technology that enables this shared resources model is partial reconfiguration of FPGAs. Partial reconfiguration is the ability for an application, such as a waveform, to be dynamically configured or reconfigured in a portion of the device, while other portions are either under use by other applications or unused. This allows support for multiple independent applications concurrently in a single FPGA, which is somewhat analogous to dynamic task switching or multitasking of a GPP. Without this capability, it would be necessary to reconfigure the entire FPGA to support a different application, which would result in the loss of all previous applications.

For example, if an FPGA was configured to support a SINCGARS communications link, it would have to be fully reconfigured to support an Enhanced Location Position Reporting System (EPLRS) communications link, resulting in the loss of the SINCGARS link, regardless of how much left over logic there may be in the FPGA. Clearly this is unacceptable for a radio. This is why the current Cluster 1 implementation, which does not support this capability, must use an inefficient dedicated resources model to support multiple channels.

In order to implement partial reconfiguration on an FPGA, three basic elements are required:

- An FPGA which inherently supports dynamic reconfiguration of only portions of the device while leaving the other portions unaffected, such as the Xilinx Virtex family of FPGAs. The Virtex family is column reconfigurable, meaning that individual columns of logic within the device can be dynamically reconfigured, independent of the rest of the columns.
- Partial reconfiguration software development tools which support the development of applications restricted to boundaries which comply with the hardware architecture of the FPGA. For the Virtex family, tools must be available to restrict applications to columns to match the columnar architecture.
- At least a basic controller must be available to dynamically manage the reconfiguration of the FPGA. This could be an embedded GPP, a soft core GPP (such as the Xilinx MicroBlaze core), or an external GPP connected to the FPGA. In this shared resources model, the same embedded GPP that is running the modem infrastructure and operating environment is also managing the partial reconfiguration of the FPGAs.

So why has this Commercial Off-the-Shelf (COTS) technology not been adopted for current SDR modem implementations, such as JTRS Cluster 1? The reason is simple: although COTS hardware has been available to support this technology for some time, COTS software development tools have not been available to make such application designs feasible. This is no longer the case. Standard tools from Xilinx are available by request for Virtex-II PRO today and for Virtex-4 in the fourth quarter of 2005.

Supporting shared resources in an SCA-enabled radio

A shared resources model enabled by partial reconfiguration of an FPGA to support multiple waveforms can be supported by the SCA as mandated for JTRS. In fact, partial reconfiguration can be viewed as creating two classes of firmware: static and dynamic. Static firmware is normally common to all waveforms, such as digital down converters, and/or provides infrastructure services such as accessing chips like ADCs, DACs, and I/Os. This type of firmware is instantiated once at power up and stays valid for the duration of the uptime of the radio. Static firmware maps well to the SCA's concept of a *device*.

On the other hand, *dynamic firmware* is waveform dependent. The dynamic firmware is instantiated when the waveform needs it. Thus, it makes sense to implement waveforms as a mix of software and dynamic firmware. Because it is only present in the FPGA when it is needed, the concept of dynamic firmware is closely related to that of a dynamic linked library. Dynamic firmware maps well to the SCA's concept of a resource. Therefore, it is simple and elegant to support in today's SCA-enabled radios.

This new firmware flexibility also creates a paradigm shift. Now it allows waveforms to take advantage of the strengths of each processor type for the most efficient implementation to minimize power and size. Indeed, the firmware can handle the highly regular signal processing with great power efficiency while the processor can handle the exceptions and other data-related issues such as switching, routing, and so on. The resulting waveform is more power and size efficient than both its firmware only and software only equivalent.

Proven technology

The shared resources model using an SCA-enabled SoC has been proven to work today. Xilinx and ISR Technologies have implemented this model in a COTS Xilinx Virtex-II PRO-based SDR modem from ISR Technologies. The demonstration system uses two modems, each supporting two independent applications: a narrowband, 256 kbps waveform supporting a communications link between two Voice over Internet Protocol (VoIP) phones and a wideband, 1024 kbps waveform supporting a streaming video link between two laptop computers. Using a COTS SCA CF from the Communications Research Centre, the video link can be instantiated and torn down while maintaining the communications link and vice versa. More details on the demonstration can be found in the December 2004 *JTRS JPO Technology Awareness Bulletin* published by the JTRS Joint Program Office at http://jtrs.army.mil/sections/technicalinformation/fset_technical.html.

Figure 3 is a floor plan of a Virtex-II PRO-based SCA-enabled SDR modem SoC supporting shared resources through partial reconfiguration. Yellow areas represent static firmware and do not change from waveform to waveform. This includes the digital down and up converter, internal shared buses (the Core Connect bus for the embedded GPP) and the interfaces to external devices, such as the A/D and D/A. The two applications (waveform A and B) run independently in the partially reconfigurable region in the right hand side of the device. If necessary, a larger waveform or smaller waveforms could run in the same space.

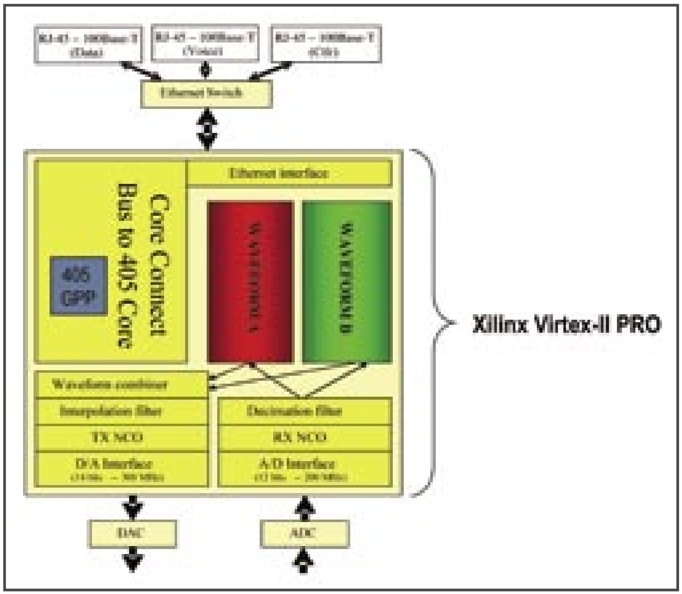

Figure 3

For the demonstration system, the engineering effort included the development of both waveforms, the development of beta software development tools to constrain the applications to partially reconfigurable regions, and integration with the SCA operating environment. This effort took approximately 10 man months in the span of two months of elapsed time.

Reconfigurability realized: SDR's real potential

Cost and power consumption are critical issues that hamper the feasibility and deployment of SDRs today. Moving from a dedicated resources model to a shared resources model can significantly drive down the cost and power consumption of the modem by factors of two and greater by reducing the number of components required to provide a given set of functionality. This model should be seriously considered for all new SDR development. Even for existing SDRs, this capability can be integrated through technology insertion to lower the production cost and increase the radio's battery life. Moreover, future waveforms can easily be instantiated using existing, already-deployed hardware. ⊕

***Manuel Uhm** is the DSP Marketing Manager for Xilinx, Inc. He is responsible for strategic marketing within the DSP Division and is one of the foremost experts in market segmentation of FPGAs in digital signal processing applications. Manuel is also currently Co-Chair of the Markets Committee of the SDR Forum. Prior to joining Xilinx, Manuel worked at Spectrum Signal Processing, Inc., a leader in Software-Defined Radio subsystems, where he was responsible for marketing.*

For more information, contact Manuel at:

Xilinx
2100 Logic Drive
San Jose, CA 95124-3400
Tel: 408-559-7778
Fax: 408-559-7114
E-mail: fpga@xilinx.com
Website: www.xilinx.com

***Dr. Jean Belzile** is a founder and CTO of ISR Technologies. He is responsible for the development of new technologies such as SCA-based SoC wireless radios. Dr Belzile has worked extensively on the OMG SWRADIO DSIG. He has authored or co-authored more than 50 papers and patents in the fields of microelectronics and signal processing for digital communication systems. Dr. Belzile is also professor at École de Technologie Supérieure.*

For further information, contact Jean at:

ISR Technologies
1100 Notre-Dame Ouest
Montréal, Québec
Canada H3C 1K3
Tel: 514-396-8622 • Fax: 514-396-8684
E-mail: jbelzile@isr-t.com
Website: www.isr-technologies.com