SDR Architecture

Contributed by Lee Pucker, Spectrum Signal Processing

Introduction

Software defined radio (SDR) is an enabling technology, applicable across a wide range of areas within the wireless industry, that provides efficient and comparatively inexpensive solutions to several of the problems inherent in more traditional radio architectures. Simply put, software defined radio (SDR) is the term used to describe radio technology where some or all of the wireless physical layer functions are software defined.

To understand this, consider the high level functional model of a radio presented in Figure 1.1[1]. On the receive side of this model, the front end processing consists of an RF subsystem that extracts the channel or channels of interest from a pre-defined spectral band, converts these channels to baseband, and forwards them onto a modem subsystem for demodulation and decoding. The modem subsystem passes the resulting analog signal or digital bitstream bearing information onto the link/network layer processing or security processing subsystem, as appropriate. This process is reversed on the transmit side, with the modem receiving an analog signal or digital bitstream bearing information, encoding this input as appropriate and creating a modulated signal bearing the associated information suitable for transmission. This signal is then passed to the RF subsystem for insertion into the wireless channel.

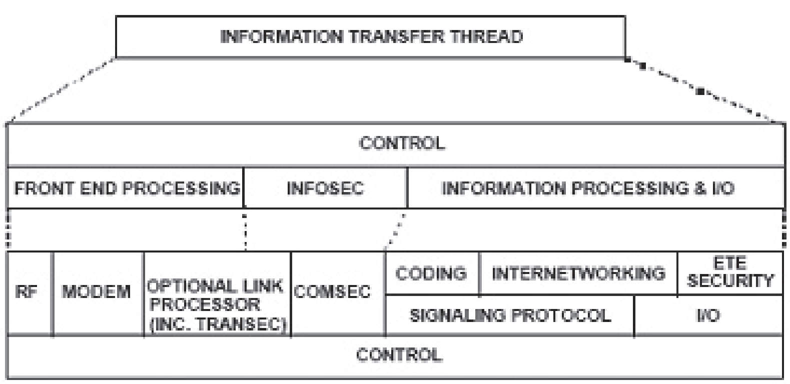

Figure 1.1 SDR Forum High Level Functional Model

In a traditional hardware radio, these functions would all be supported through a fixed-function hardware architecture employing an array of application specific integrated circuits (ASICs), RF integrated circuits, and discrete RF and digital components to support the target air-interface standard [2]. These traditional hardware elements might allow some flexibility through a software control interface, such as choosing the channel of interest or setting the transmit power level, but the basic function of the radio is relatively fixed. In a software defined radio, however, these functions are performed through modifiable instructions that are executed on programmable processing devices such as a field programmable gate array (FPGA), a digital signal processor (DSP), or a general purpose processor (GPP). This allows for significant flexibility in the radio's functionality, which provides multiple benefits for wireless original equipment manufacturers, service providers and end-users, such as:

- Reducing the costs associated with product development, since a single development project can now potentially support multiple market segments, and software development based on reusable waveform components can often replace a costly ASIC development cycle,
- Improving time to market in supporting new revenue generating services or features, since these upgrades can be provided through software download, potentially over the air, versus requiring a new hardware platform,
- Reducing installation and support costs, since a common set of inventory can be utilized for multiple markets, and "bug fixes" can be facilitated through software download versus hardware redesign, and
- Improving the ability of the radio to interoperate on multiple independent networks, allowing the user to seamlessly roam across network boundaries and achieve true mobility [3, 4].

Market-Specific Architectural Models Supporting Software Defined Radio

While the high-level functional model for a software-defined radio presented in Figure 1 provides a useful tool for discussing SDR technology in general, the model is often insufficient in addressing the precise needs of specific wireless markets. This high-level model must be tailored, therefore, to provide an appropriate model for each specified market. For example, consider the base station generalized functional SDR architecture presented in Figure 1.2 [5]. In this architecture, which is consistent with the base station architectures supported by the Open Base Station Architecture Initiative (OBSAI) and the Common Public Radio Interface (CPRI), the INFOSEC and Information Processing Blocks of the of the generalized high level architecture are encapsulated in a single "Call/Message Processing and I/O" block while the RF and modem subsystems have been decomposed into four separate functional blocks as follows [6]:

- An Antenna subsystem, which may include specialized processing supporting frequency diversity, smart antenna, or beam forming.
- An RF subsystem converting one or more frequency bands of interest to an analog or digital IF signal. This subsystem may incorporate multiple RF front ends operating in parallel to support the base station's operational requirements.
- A Channel Selector/Combiner subsystem providing digital frequency tuning, channel selection, and digital sample rate conversion, as appropriate, to support the target air interface standard associated with each active channel. Functionality in this subsystem may be provided through one or more application specific standard processors (ASSPs) or through a programmable device such as an FPGA.

- Baseband DSP Processing, providing modem and channel codec processing for one or more channels utilizing programmable signal processing devices. For multi-channel systems, this subsystem may be provided in the form of a “DSP farm” providing a pool of DSP resources that can be dynamically allocated to specific channels based on the current network load.

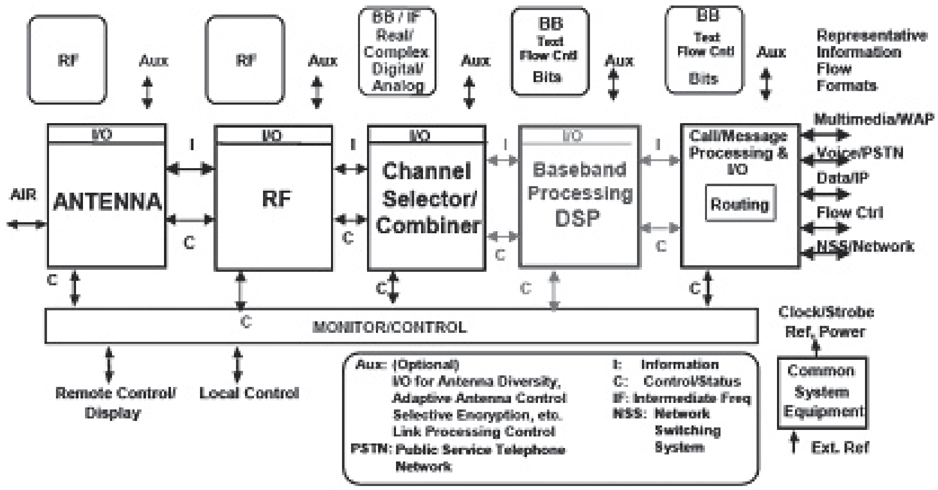

Figure 1.2 Base Station Generalized Functional SDR Architecture

Handsets also have architectural requirements that can be defined by tailoring the high level functional model provided in Figure 1.1. One such tailoring is illustrated in Figure 1.3 [7]. In this model, a common baseband processing engine can service multiple RF front ends, each of which supports a specific air interface standard operating in a specific frequency band of interest. The baseband processing engine in this architecture is dynamically loaded with code supporting the mode of operation required by the handset operator at any given time. The interface between the baseband processing engine and the RF front end is then switched to connect to the appropriate RF front end supporting this mode of operation. The baseband processing engine in this model may be provided through a combination of technologies such as an ASSP, FPGA, or DSP, or may be provided through a software defined system-on-a-chip (SoC) integrating the technologies into a single programmable device optimized for power and cost.

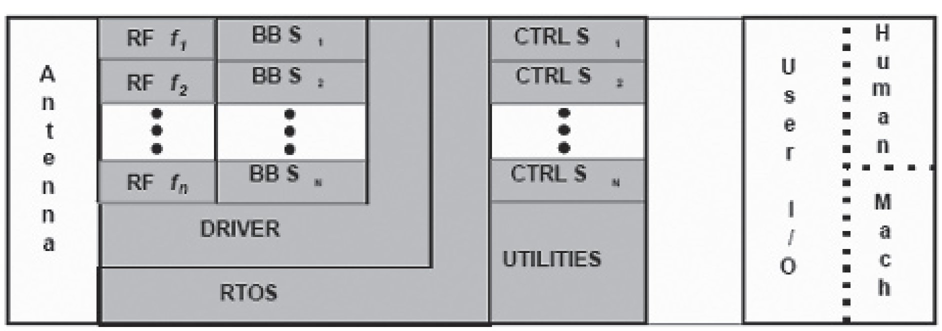

Figure 1.3 Multiband, Multimode Handheld Functional Model

Standardization in SDR Architectures

The benefits of SDR for OEMs, such as code reuse and in-service upgrades are often achieved through standardization of key elements within the SDR architecture, providing a common platform for the development of SDR technologies. The standards supported may be proprietary to the OEM, incorporating intellectual property that differentiates the OEM in the market, or they may be industry standards developed through a consensus process to commoditize the technology, allowing support by third parties in creating the radio platform to achieve specific business objectives. Typical areas of standardization, all of which are highly inter-related, are as follows:

- Application Frameworks – An application framework provides the standard framework for setting up, tearing down, configuring and controlling waveform applications running on the SDR platform. These frameworks typically specify the software operating environment, including the base application interfaces, necessary to support the waveform applications. Examples of application frameworks relevant to SDR systems include the Software Communications Architecture (SCA) supported by the SDR Forum's SCA Working Group, as well as the frameworks from other industry consortia, such as the Open Mobile Alliance™ and the Service Availability Forum ™ [8, 9].
- Hardware Abstraction Layer – A hardware abstraction layer, as described by the SDR Forum's Hardware Abstraction Layer Working Group, "assists with the portability of high speed signal processing code (FPGA, DSP, or other) in the signal processing subsystem (SPS)" (see Figure 1.4) [10]. Elements of a hardware abstraction layer include standardized elements of the SPS operating environment that reduce the changes required in porting a waveform or its constituent components from platform to platform, and standardized methods for configuration and control that reduce the costs of modifying the code in the rest of the system to accommodate a change in the SPS architecture.
- Radio Service APIs – Radio Service APIs are defined by the SDR Forum's System Interface Working Group as "an agreement of services provided and required behavior among related software and/or hardware modules" [11]. Radio service API's are generally provided by the radio platform as services to the waveform application, and often include the following:
    - Timing Service APIs – These APIs are provided by the radio platform to support time synchronization between waveform components. These APIs are is especially important in radios supporting waveforms or air interface standards requiring temporal synchronization on a time division multiple access (TDMA) or frequency hopped network, and must be accurate to within the specifications of that network.
    - Digital IF APIs – These APIs are provided by the radio platform to facilitate the transfer of signal and control data between the RF front end and waveform components operating in the baseband processing subsystem.
    - Smart Antenna APIs – These APIs are provided by the radio platform to provide the functionality required to support a smart antenna sub-system, including APIs supporting distributed processing, configuration, and control.
    - Audio Service APIs – These APIs are provided by the radio platform to facilitate transfer of signals between waveforms components and the radio's audio subsystem.

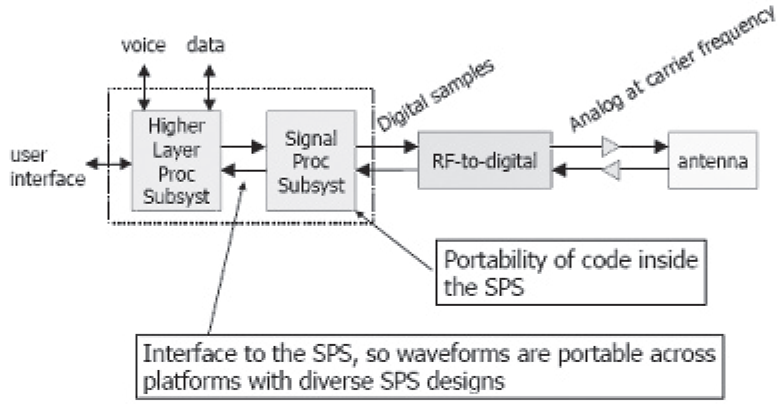

Figure 1.4 SDR Architecture Model Showing Areas for Hardware Abstraction

Network Technologies Supporting SDR Architectures

Standardization may also be required in software defined radios in areas that extend beyond the base radio functionality and into the network. For example, protocols may be established to manage radio software download. Radio software download has been described by the SDR Forum's Download Working Group as "the process of delivering reconfiguration data and/or new executable code to a SDR device to modify its operation or performance" [11]. Radio software download, which may include over the air reprogramming, provides manufacturers and service providers the ability to update the radio's software while in service, including both the application software and the radio software, as illustrated in Figure 5. This latter ability is of primary concern to regulators as the download of new radio software may cause the software defined radio to behave inappropriately, disrupting the operation of other radios and networks. Radio software causing this type of behavior may be the result of any number of things, including poor software design, poor testing, or malicious intent on the part of a "hacker". Regardless of the cause, the ability to support software radio download necessitates the need to standardize the policies and protocols associated with the download process to prevent these types of issues.

Support for cognitive radio applications also may require additional standardization extending beyond the base radio functionality. Cognitive radio technology allows a radio to automatically adjust its behavior or operations based on an awareness of its environment, internal state, and location. Cognitive radio may be utilized to allow a radio to choose the network it wishes to operate on based on an assessment of available networks. This assessment can be performed following a "listen before talk" model to sense the networks that are available in a specific location, or utilizing a network construct such as the cognitive pilot channel proposed by the End-To-End Reconfigurability Program to provide network information to the radio using a radio software download protocol [12]. In either case, a policy engine supporting network selection and managing the radio's operation in the context of the cognitive function is generally required to ensure appropriate network behavior.

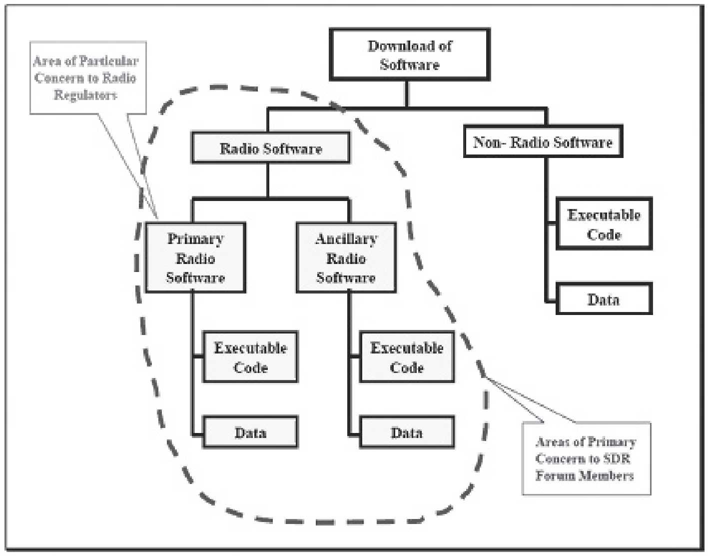

Figure 1.5 Characterizations of Software Download

References

1) SDR Forum, “TECHNICAL REPORT 2.1: Architecture and Elements of Software Defined Radio Systems as Related to Standards”, November 1999
2) SDR Forum, “FAQ”, http://www.sdrforum.org/pages/aboutTheForum/faqs.asp
3) Lee Pucker, “Software Defined Radio Technology in Commercial and Military Applications: A Contrast in Requirements”, Proceedings of the SDR ’02 Technical Conference, November 2002
4) Stephen M. Blust, “Perspective on Software Defined Radio - Download and Reconfigurability for Radio Software” ITU Seminar on IMT-2000 and Systems Beyond Ottawa, 28 May 2002
5) SDR Forum Base Station Working Group, “Base Station System Structure”, SDRF-01-S-0006-V2.00, 15 January 2002
6) Lee Pucker, “OBSAI and CPRI Update”, SDRF-04-I-0038-V0.00, 21 April 2004
7) SDR Forum, “Software Defined Radio Commercial Handset Guidelines”, SDRF-04-A-0006-0.00, 25 August 2004
8) Open Mobile Alliance, “OMA Service Environment Architecture Document Version 1.0.3”, 22 June 2006
9) Service Availability Forum, “Introduction to the Service Availability Forum”, http://www.saforum.org/about/saforum_introduction/index.html
10) SDR Forum Hardware Abstraction Layer Working Group, “Hardware Abstraction Layer Working Group Report on Results of Request For Information”, SDRF-04-A-0009-V0.00, 4 October 2004
11) SDR Forum System Interface Working Group, “API Position Paper”, SDRF-03-I-0030-V1.00
12) SDR Forum Download Working Group, “Overview and Definition of Software Download for RF Reconfiguration”, SDRF-02-A-0002-V0.00, 6 August 2002
13) Dr. Didier Bourse, “E2R Reconfigurable Radio Resource and Spectrum Management”, WWI-MOC-CA Workshop, 30 March 2006