A GENERIC ARCHITECTURE FOR SMART MULTI-STANDARD SOFTWARE DEFINED RADIO SYSTEMS

S.A. Bassam, M.M. Ebrahimi, A. Kwan, M. Helaoui, M.P. Aflaki, O. Hammi,
M. Fattouche, and F.M. Ghannouchi
iRadio Laboratory, Electrical Engineering Department, University of Calgary, Calgary,
Alberta, Canada, T2N 1N4.

ABSTRACT

One of the main challenges in designing a reconfigurable radio is to develop an efficient procedure to reconfigure the radio system to any new waveforms. Proper implementation procedure would allow the radio system to work for different standards while using the same hardware platform. It will also give the opportunity for upgrading the software of the radio system to new waveform standards as soon as they are available and needed. While several attempts have been proposed to implement co-existing waveforms of different standards, it is believed that standardizing the definition of waveforms will lead to a more generic way of implementing smart multi-standard software defined radio (SDR) terminals. Based on this conception, this paper proposes a solution for a generic SDR architecture that can be used to implement smart multi-standard SDR terminals.

1. INTRODUCTION

The concept of software defined radio (SDR) consists in controlling and configuring the radio operation by software in order to migrate easily from one standard to another and to serve multi-standard applications. The ideal SDR architecture provides a system capable of reconfiguring itself to any waveform standards through a software interface [1]. In other words, users or vendors should have the capability to program their radio systems to work with any waveform standards by setting the waveform parameters and without changing any hardware components. Software communications architecture (SCA) based reconfigurable SDR architecture is a well-known solution for designing an SDR-based transceiver [2]. A schematic block diagram of this architecture is presented in Figure 1. It consists of the use of software communication architecture (SCA) and common object request broker architecture (CORBA) [3] to standardize the interface between the hardware (radio) and software (waveform development application). Indeed, it was initially developed to satisfy US military need in upgrading the hardware of their communication infrastructure with minimum cost.

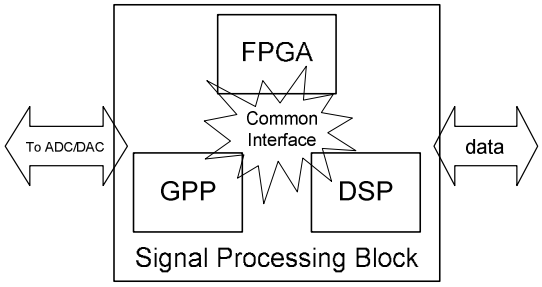

Figure 1. SCA-enabled interface

The idea consists in developing an application independently from the hardware, which facilitates its portability to different hardware systems without the need to modify the application.

However, when dealing with the same hardware, the use of such an interface is not relevant. Because, it introduces unnecessary complexity by loading the general purpose processor (GPP), digital signal processor (DSP) and probably the field programmable gate array (FPGA) with standardized communication protocols (as shown in Figure 1), which results in reducing the performance of the system. Therefore, an SCA-based solution is expensive since it over-sizes the hardware (GPP, DSP and FPGA) in order to handle the extra tasks. In addition to the higher hardware cost, acquiring the licenses for commercial development infrastructures, such as SCA and CORBA based middleware solution, increases the cost of the proposed solution.

When dealing with the same hardware, the main challenge in designing a reconfigurable radio is to develop an efficient procedure to reconfigure the radio system to any new waveforms. Proper implementation procedure should allow the radio system to work for different standards while using the same hardware platform. It should also give the opportunity for upgrading the software of the radio system to new waveform standards as soon as they are available and needed. While several attempts have been proposed to implement co-existing waveforms of different standards, it is believed that standardizing the definition of waveforms will lead to a more generic way of implementing smart multi-standard software defined radio (SDR) terminals.

Based on this concept, this paper proposes a solution for a generic SDR architecture that can be used to implement smart multi-standard SDR terminals.
This paper is organized as follows. The proposed generic architecture is detailed in section two. The implementation of this proposed solution is demonstrated in section three. Finally, section four reports the validation results for wireless LAN standard and section five is the conclusion.

2. GENERIC ARCHITECURE FOR MULTI-STANDARDS SDR SYSTEMS

2.1. Architecture of the proposed solution

The proposed generic architecture offers a standardized way to define the different parts of the physical layer and MAC layer for a given radio system and sets the parameters in order to reconfigure the hardware and software in a relatively short time to comply with the standard considered for implementation. The design and connection of the different blocks as well as the validation of the implemented radio are not required to be done on the mobile terminal. On the contrary, its achievement in a centralized processing unit would reduce the implementation cost of the terminal by reducing the computational requirements for end-users terminals. Moreover, the centralization of the standard design and configuration saves the implementation costs since most of the third party software licenses will not be required to be installed in the end-user terminals.
Figure 2 shows the block diagram of the proposed solution. The radio solution consists of four blocks; a GPP, a DSP board, a FPGA board and a RF front-end circuit. Standard communication protocols and interfaces were chosen to communicate between different blocks in order to reduce the complexity of the implementation, since the solution is not required to be software independent. Indeed, the GPP has access to each of the other three blocks to download the configuration files. The DSP controls the FPGA in order to download and/or upload data and control information. The FPGA is an interface between the DSP and the RF front-end. The communication between the DSP, FPGA and RF front-end is achieved through two physical channels; one for data and one for signaling. Below is a description of the main function of each of the blocks:

- *The General Purpose Processor*

The GPP block main role is to provide the tools necessary for configuring the radio to a given standard rapidly and automatically. This task is achieved through the following steps:

- Configure the different blocks of the radio using the radio design application and user interface.

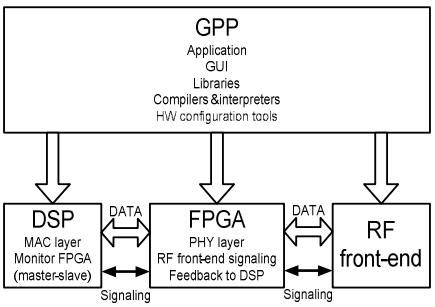

Figure 2. Block diagram of the propose solution

- Store and easily extend the libraries used in the design of the radio. Two different libraries, MAC-S and COR-LIB, will be proposed in the following section to facilitate the fast implementation and reconfiguration of the MAC and physical layers respectively.
- Compile the design using the DSP and FPGA compiling tools.
- Test and validate the implemented standard. A validation environment (VE) is used for this purpose.
- Generating the configuration files in order to send them to each of the other three blocks.

To achieve an effective cost solution, a distributed GPP, which consists of a central GPP for the network and a remote GPP for each end-user terminal, is preferable. In this case, the central GPP is in charge of running the user interface and the radio design application in order to define the new standard. Moreover, it compiles the designed radio and tests its validity. Once the new standard is validated, the configuration files for the DSP, FPGA and RF boards of the end-user terminal is generated and downloaded to the remote GPP. This latter is responsible for configuring the DSP, FPGA and RF front-end boards. In addition, it can offer the option to store in a flash memory the configuration files for different standards to allow for switching, upon the user request, between standards in a multi-standard environment.

- *The Digital Signal Processor*

The DSP block controls the FPGA to provide access to the data. A master/slave configuration can be considered for the communication between both blocks, where the DSP initiates and controls the communication in both ways. The DSP is also in charge of implementing the software part of the radio, mainly the MAC layer and higher level layers. Since MAC-S is stored in the central GPP, less memory constraints are imposed on the implementation of a given standard because only the components necessary for that

particular standard will be downloaded in the configuration file.

- *The digital hardware (FPGA)*

The FPGA implements the physical layer of the radio, which contains most of the components that require parallel processing or dedicated hardware. In addition, it is used as an interface between the digital signal processing (software in the DSP) and the analog wireless radio (hardware). The digital-to-analogue and analogue-to-digital converters as well as FIFOs and RAMs dedicated for this purpose ensures the necessary signal conditioning for this interfacing operation. Indeed, in transmission mode, the FPGA takes the digital data provided by the DSP, processes it and converts it to analogue data using the in-board digital-to-analogue converters, which will be fed to the RF board. Oppositely, in the receiving mode, the analogue signal from the RF board is received and converted to digital data in the FPGA, which processes it before storing it in the memory that the DSP will access to copy the data.

2.2. Reconfigurability options

Figure 3 shows an illustration of the architectural organization of the proposed solution. The lowest level, the hardware, consists of the platform that contains three boards; a DSP board, an FPGA board and an analog board implementing the RF front-end. This hardware is configured by the mean of programming tools using the files generated in high level programming environments after processing the designed radio. The radio design is achieved by selecting and configuring the appropriate components from the configurable libraries MAC-S and COR-LIB. New components can be introduced to these libraries when needed. The designed radio is tested and validated through a validation environment.

The Open System Interconnection Model defines seven layers for the communication networking protocol. The MAC layer acts as an interface between the second layer (Data Link Layer), and the first layer (Physical Layer). The MAC layer's main goal is to establish commands through the physical layer to allow multiple access with the channel.

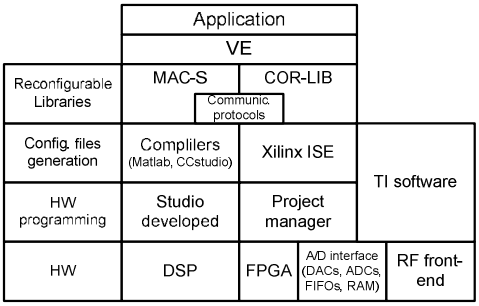

Figure 3. Architectural description of the proposed solution

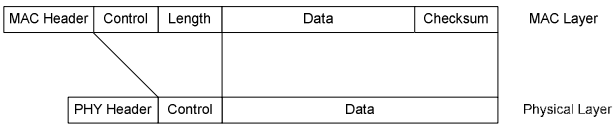

Figure 4. Example of translating MAC layer frame to Physical layer frame

If needed, management commands (association with a base station) may be used in addition to the data transmission commands. The MAC layer data frame consists of common key field such as control, source address, destination address, length, and checksum. Each protocol can define more key fields according to the required specification. These data fields should be processed and passed to the physical layer in a format that the physical layer can process. Figure 4 shows a generic model for converting a MAC data frame to a physical data frame.

The DSP in the proposed SDR will handle the duties of the MAC layer. In a reconfigurable SDR, there should be support for multi-standard configurations. The DSP can be reconfigured to process the incoming MAC frame to a target physical frame by using different user defined functions for each wireless standard. The MAC frame incoming parameter for the DSP function should be constant; however, each field in the frame should support variable size parameters. Figure 5 shows the interface diagram for processing one MAC frame to different wireless Physical frame standards.

The physical layer determines the hardware specifications, encoding, and data transmission over the communication channel. The re-configurability aspect for the FPGA is more stringent than that of the DSP. In a normal operating scenario, the FPGA cannot reconfigure certain processing elements (for example, swapping two blocks) unless that scenario has been previously taken into account. The classification of online and offline classes of components is needed to determine which, if any of the blocks can be suited for online re-configurability. Online blocks should be designed with parameterization in mind, so that they can be adjusted without significant modification of the FPGA program (for example, changing the code rate of a convolutional encoder from ½ to ¾).

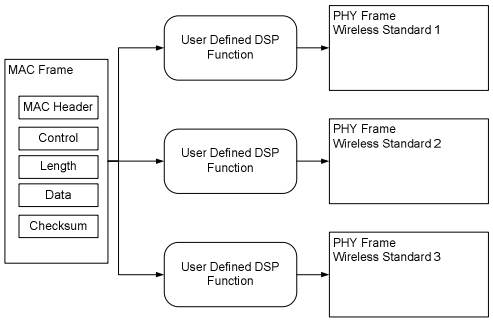

Figure 5. Multi-standard re-configurability

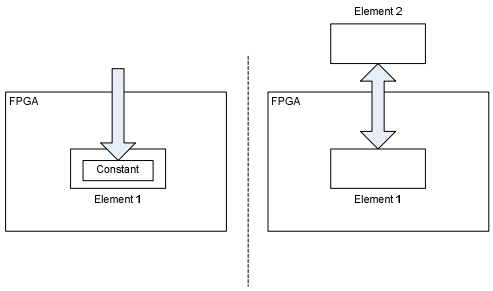

Figure 6. Example of online re-configurability by parameterization (left), and offline re-configurability (right)

Other blocks, such as switching and adding blocks, have to modify the main implementation and therefore require recompilation. Then, the recompiled firmware can be reloaded by the GPP to allow the new standard to run as shown in figure 6.
In the proposed SDR system, the FPGA handles the physical layer aspect of the protocol, since the timing of the data needs to be processed in a real time environment.

3. VALIDATION OF THE GENERIC ARCHITECTURE WITH WIRELESS LAN IMPLEMENTATION

3.1. Hardware implementation of the proposed solution

An SDR platform based on the proposed architecture was developed. It is composed of the following modules:

- Digital signal processing module
- Data conversion module
- RF module

The first two modules are part of the Small Form Factor (SSF) SDR platform from Lyrtech [4] and the RF module is a wireless transceiver module from Texas Instruments (TI) designed to operate at a carrier frequency equal to 2.4 GHz. The digital signal processing module contains two distinct processing units:

- FPGA : Virtex-4 from Xilinx Inc.
- DSP/GPP : TMS320DM6446 DMP SoC from Texas Instruments

The use of either one of the processing units depends mainly on the type of signal processing required. Indeed, parallel processing is more suitable for FPGA implementation, which optimizes the execution time. While sequential processing is more convenient for DSP implementation since less resources will be needed for the same speed of execution.

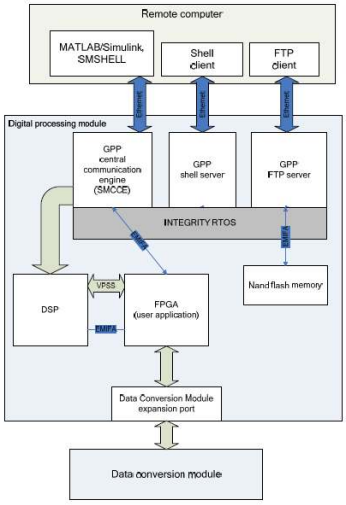

Figure 7. Communication and software communication of the platform

Figure 7 illustrates the possible communication links available through the platform. An Ethernet link provides the communication between the host computer and the GPP processing unit on the digital processing module. The GPP provides the link between the host computer and the DSP and FPGA to download the codes and to send the demodulated signal from the digital processing unit to the host computer for analysis using the validation environment. The data conversion module is equipped with two 14 bits, 125-MSPS analog-to-digital converter and dual channel 16-bits, 500 MSPS digital-to-analog converters.
The RF module consists of the TSW5002 board from TI, which can up-convert the signal from baseband to RF around a carrier frequency of 2.45 GHz and vice versa.
The RF board can be controlled by software, which controls and adjusts the frequencies of the different up-conversion stages, filters' bandwidths, and gain and power of the different components..

3.2. Configuration of a WLAN waveform

The wireless LAN (WLAN) standard [5] was considered for implementation in the proposed generic architecture. Figure 8 presents the block diagram of the WLAN standard. The block diagram consists of two major units:

- Baseband processing unit
- RF unit

The baseband processing unit receives the raw data and prepares it for transmission. The output of this unit is still a baseband signal however data processing operations are done in order to be able to retrieve the transmitted information at the receiver. At the receiver side, the opposite processing is done to extract the data. The RF units are used

to up-convert (down-convert) the baseband (RF) signal to RF (baseband) for transmission (reception) purposes.

Analyzing the WLAN block diagram in Figure 8 reveals that there are blocks which can be used in several standards with minor reconfiguration. For example, the convolution encoder and Viterbi decoder are used in more than one standard.

In addition, some blocks are common for different standards with similar transmission method. For instance, FFT/IFTT, pilot insertion, guard interval, and OFDM symbol assembler are used in OFDM based systems such as WLAN, digital video broadcasting (DVB), and World Interoperability for Microwave Access (WiMAX) [6].

The implementation approach consists of starting from a general block diagram which is able to cover any waveform. Figure 9 shows the proposed generic block diagram of the SDR radio used in the implementation of the WLAN waveform. The hardware of the baseband-to-RF part is independent from the standard being considered. Only minor changes such as filter bandwidth and sampling rates are needed to be modified when switching from one waveform to another. This part is the only part where the analog components with less re-configurability capabilities are used.

The other three parts can be developed in the digital processing units such as FPGA and DSP depending on the size, speed, and processing criteria. In order to implement multi-standards capabilities, libraries that contain the most commonly used components in wireless communication standards were created.

Each of the components in the library are assigned to the specific signal processing unit (DSP or FPGA) and by using that component the proper code for that unit will be developed. Therefore, the user just needs to drag the necessary components from the library, adjust its parameters, and finally configure the system for the specific waveform being transmitted / received. Finally the configured waveform is downloaded and coded into the platform. This is a systematic approach to develop the waveform. Depending on the standard definition, the user selects the appropriate building blocks, and configures the system accordingly.

It is always possible to augment the library components list and add new components using the programmable components (user-defined) available in each section of the library. In this case, the user needs to write the proper code for these new components. The user-defined components are coded for the specific processing unit according to their categories.

4. CONCLUSION

This paper proposed a standardized architecture able to reconfigure SDR communication terminals from one standard to another. It is mainly developed to reach the need of reconfigurable SDR platforms where the hardware of the terminal is fixed. In this regard, we proposed a generic architecture that consists of four blocks; a GPP, a DSP board, a FPGA board and a RF front-end circuit where standard communication protocols and interfaces were chosen to communicate between different blocks in order to reduce the complexity of the implementation. The proposed architecture was used to generalize the implementation of the wireless LAN protocol. The results show the possibility of using the proposed generic architecture to rapidly reconfigure the SDR platform for a specific standard.

5. REFERENCES

[1] J. Mitola, "The software radio architecture," IEEE Communication Magazine, vol. 33, no. 5, pp. 26-38, May 1995.

[2] JTRS Standard, "Software Communications Architecture Specification," Version 2.2.2, 15 May 2006 (http://sca.jpeojtrs.mil/downloads.asp?ID=2.2.2).

[3] CORBA, "Common Object Request Broker Architecture: Core Specification", accessible at: http://www.omg.org/docs/formal/04-03-12.pdf

[4] Small Form Factor (SFF) SDR development platform from Lyrtech Incorporated. Product description can be found at: http://www.lyrtech.com/index.php?act=view&pv=SFF%20SDR%20development%20platforms

[5] IEEE Std 802.11™-2007 (Revision of IEEE Std 802.11-1999), "Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications", accessible at: http://standards.ieee.org/getieee802/download/802.11-2007.pdf

[6] IEEE Std 802.16e-2005, "Part 16: Air Interface for Fixed Broadband Wireless Access Systems - Physical and Medium Access Control Layers for Combined Fixed and Mobile Operation in Licensed Bands," web-site: http://wirelessman.org/pubs/80216e.html.

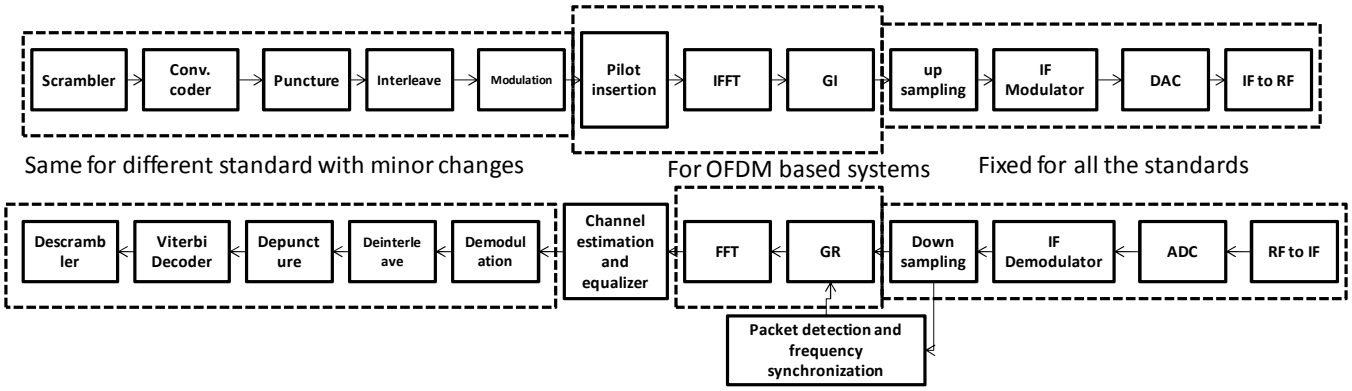

Figure 8. WLAN block diagram

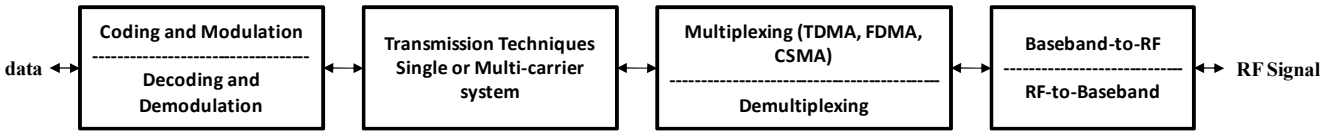

Figure 9. General block diagram