Computer Communications 128 (2018) 106–125

Contents lists available at ScienceDirect

Computer Communications

journal homepage: www.elsevier.com/locate/comcom

Software-defined Radios: Architecture, state-of-the-art, and challenges

Rami Akeela, Behnam Dezfouli*

Internet of Things Research Lab, Department of Computer Engineering, Santa Clara University, USA

ARTICLE INFO

Keywords:
SDR
Wireless communication
Programmability
Co-design
LTE
WiFi
IoT

ABSTRACT

Software-defined Radio (SDR) is a programmable transceiver with the capability of operating various wireless communication protocols without the need to change or update the hardware. Progress in the SDR field has led to the escalation of protocol development and a wide spectrum of applications, with a greater emphasis on programmability, flexibility, portability, and energy efficiency in cellular, WiFi, and M2M communication. Consequently, SDR has earned a lot of attention and is of great significance to both academia and industry. SDR designers intend to simplify the realization of communication protocols while enabling researchers to experiment with prototypes on deployed networks. This paper is a survey of the state-of-the-art SDR platforms in the context of wireless communication protocols. We offer an overview of SDR architecture and its basic components, and then discuss the significant design trends and development tools. In addition, we highlight key contrasts between SDR architectures with regards to energy, computing power, and area, based on a set of metrics. We also review existing SDR platforms and present an analytical comparison as a guide to developers. Finally, we recognize a few of the related research topics and summarize potential solutions.

1. Introduction

Advances in wireless technologies have altered consumers' communication habits. Wireless technologies are an essential part of users' daily lives, and their impact will become even greater in the future. In a technical report, the World Wireless Research Forum (WWRF) has predicted that for 7 billion people, 7 trillion wireless devices will be deployed by 2020 [1]. When these devices are connected to the Internet to form an Internet of Things (IoT) network, the first challenge is to adjust the basic connectivity and networking layers to handle the large number of end points. There is an increasing number of wireless protocols that have been developed, such as ZigBee, Bluetooth Low Energy (BLE), Long Term Evolution (LTE), and new WiFi protocols, that have been developed to meet the demanding requirements of various domains such as 5G, IoT, and cyber-physical systems [2–4]. Wireless standards, in general, are adapting quickly in order to accommodate different user needs and hardware specifications [5,6]. To meet these specifications, a transceiver needs to be designed with the ability to handle several protocols, including the existing ones and those being developed. In order to accomplish this task, one needs to recognize the protocols' need for a *flexible, re-configurable*, and *programmable* framework.

Both consumer enterprise and military frameworks have a need for programmable platforms. Due to the rapid and consistent advancement of wireless protocols, programmability is of central significance to designers in the industry. Hardware needs to be able to keep up with both the evolution of technology and the changing user demands. For example, the authors in [7] proposed a platform called OpenRadio for programming both Physical (PHY) and Medium Access Control (MAC) layers while offering a high level of abstraction. Rather than including yet another piece of equipment to deal with a new standard or recurrence band, the equipment of a formerly introduced platform is able to adjust to the features of another standard. In a military scenario, for example, the needs of these platforms can change in light of the highly unpredictable conditions that arise during a mission. While these needs might not have been envisioned when designed initially, they led to the development and utilization of new protocols.

Software-defined Radio (SDR) is a technology for radio communication. This technology is based on software-defined wireless protocols, as opposed to hardware-based solutions. This translates to supporting various features and functionalities, such as updating and upgrading through reprogramming, without the need to replace the hardware on which they are implemented. This opens the door to the possibility of realizing multi-band and multi-functional wireless devices.

The driving factors for the high demand of SDR include network interoperability, readiness to adapt to future updates and new protocols, and most importantly, lower hardware and development costs. In

* Corresponding author.
E-mail addresses: rakeela@scu.edu (R. Akeela), bdezfouli@scu.edu (B. Dezfouli).

https://doi.org/10.1016/j.comcom.2018.07.012
Received 24 March 2018; Received in revised form 27 June 2018; Accepted 5 July 2018

a report [8], the SDR market is projected to be worth more than $29 billion by the year 2021. Global Industry Analysts, Inc. [9] highlights some of the market trends for SDR as follows: (i) increasing interest from the military sector in building communication systems and large-scale deployment in developing countries, (ii) growing demand for public safety and disaster preparedness applications, and (iii) building virtualized base stations (BSs). SDRs are also ideal for developing future space communications [10–12], Global Navigation Satellite System (GNSS) sensors [13], Vehicle-to-Vehicle (V2V) communication [14–16], and IoT applications [17,18], where relatively small and low-power SDRs can be utilized.

The SDR industry flourished due to the Joint Tactical Radio System (JTRS) program, which was responsible for producing SDRs for the military. In turn, this led to the creation of an entire world of new technologies, Software Communications Architecture (SCA), and Electronic Design Automation (EDA) tools that facilitate the development of SDRs [19]. The newly abundant resources made it relatively feasible to fuel the effort to develop more SDRs, not only for the military, but also for civil applications. The first commercial SDR, named Anywave [20], was a dual-mode base station that supported both Global System for Mobile communication (GSM) and Code Division Multiple access (CDMA) concurrently and ran on GPPs. Another technological advancement with a huge impact on the SDR industry was the development and release of Radio Frequency Integrated Circuit (RFIC), which supports most frequency bands in the MHz to GHz range.

Researchers have been studying SDRs for several years and are striving to find better means of implementing them in order to optimize their processing and energy efficiency. SDRs are implemented using various types of hardware platforms, such as General Purpose Processors (GPPs), Graphics Processing Units (GPUs), Digital Signal Processors (DSPs), and Field Programmable Gate Arrays (FPGAs). Each of these platforms is associated with its own set of challenges. Some of these challenges are: utilizing the computational power of the selected hardware platform, keeping the power consumption at a minimum, ease of design process, and cost of tools and equipment. Both the research community and industry have developed SDRs that are based on the aforementioned hardware platforms. A few examples include USRP [21], Sora [22], Atomix [23], Airblue [24], and Wireless Open Access Research Platform (WARP) [25]. Each SDR is unique with regards to the design methodology, development tools, performance, and end application.

In this paper, we first present an overview of the SDR architecture, as well as the analog and digital divides of the system and interconnection of components. Then, we introduce the criteria that defines how the different hardware platforms are classified. We thoroughly examine the architecture and design approaches employed by these hardware platforms and present their strengths and weaknesses in the context of SDR implementation. Furthermore, we provide an analytical comparison of hardware platforms as a guide for design decision making. Moreover, we discuss the use of development tools and present a summary to give a streamlined explanation of their functionalities and the platforms they support. Afterwards, we review the SDR platforms developed by both industry and academia, analyze them, and compare them using the criteria that was discussed earlier. Finally, we identify the current challenges and open research topics that are related to future SDR development.

This paper is organized as follows: Section 2 provides a description of SDR architecture and the classification process that is used to summarize the various design approaches adopted. Section 3 provides a comprehensive study of all the hardware platforms and associated design methodologies that are used to build SDR platforms. Section 4 lists some of the corresponding development tools and platforms. Section 5 presents an analysis and comparison of the commercially and academically developed SDR platforms. Section 6 highlights research questions and future trends. Section 7 presents an analysis of the existing literature on SDR surveys. We conclude the paper in Section 8. A list of key abbreviations used in this paper can be found in Table 1.

Table 1
Key abbreviations.

ADC	Analog-to-Digital Converter
ASIC	Application-Specific Integrated Circuit
BS	Base Station
CUDA	Compute Unified Device Architecture
DAC	Digital-to-Analog Converter
DSP	Digital Signal Processor
FFT	Fast Fourier Transform
FLOPS	Floating Point Operations Per Second
FPGA	Field Programmable Gate Array
GPP	General Purpose Processor
GPU	Graphics Processing Unit
HLS	High Level Synthesis
NFV	Network Function Virtualization
RTL	Register-Transfer Level
SDR	Software-Defined Radio
SDN	Software-Defined Network
SNR	Signal-to-Noise Ratio
SoC	System on Chip
USRP	Universal Software Radio Peripheral

2. Concepts and architecture

In this section, we examine the general architecture of SDRs, their main components, and their processing requirements. As explained in the previous section, SDRs play a vital role in wireless standard development due to their flexibility and ease of programmability. This is due to the fact that most digital signal processing and digital front end, which includes channel selection, modulation and demodulation, takes place in the digital domain. This is usually performed in the software running on processors, such as GPPs and DSPs. However, it can also run on programmable hardware, i.e., FPGAs.

In general, from the transmitter's point of view, first a baseband waveform needs to be produced and then an Intermediate Frequency (IF) waveform. A RF waveform will be generated and then sent through the antenna. From the receiver's point of view, this RF signal is sampled, demodulated, and then decoded. To provide more details to the process, we study the receiving end of the system as follows.

The RF signal from the antenna is amplified with a tuned RF stage, which amplifies a range of the frequency band. This amplified RF signal is then converted to an analog IF signal. The Analog-to-Digital Converter (ADC) digitizes this IF signal into digital samples. Then, it is fed into a mixer stage. The mixer, which is an electrical circuit that takes in two signals and yields a new frequency, has another input coming from a local oscillator with a frequency that is set by the tuning control. The mixer then translates the input signal to a baseband. The next stage is a Finite Impulse Response (FIR) filter that permits only one signal. The FIR is a combination of multiply-add units and shift registers. This filter limits the signal bandwidth and acts as a decimating low-pass filter. The digital down-converter includes a large number of multipliers, adders, and shift-registers in the hardware in order to accomplish the aforementioned tasks. Next, the signal processing stage performs tasks such as demodulation and decoding. This stage is typically handled by a dedicated hardware like an Application Specific Integrated Circuit (ASIC) or other programmable alternatives like FPGA or DSP [26].

As shown in Fig. 1(a) and (b), at a high level, a typical SDR transceiver consists of the following components: Signal Processing, Digital Front End, Analog RF Front End, and an antenna.

2.1. Antenna

SDR platforms usually employ several antennas to cover a wide range of frequency bands [27]. Antennas are often referred to as