SODIO
A Software Radio Platform for advanced HF communications

Department of Communications
Enginyeria i Arquitectura La Salle
Ramon Llull University.
Pau Bergadà, Carles Vilella, Marc Deumal, Joan Luís Pijoan.

1. Abstract

Software Defined Radio (SDR) is happening to be the key to find a common global standard that could solve the problem of current multi-standard wireless communications.
The attractions of a SDR are not only flexibility and ease of adoption, but also that almost any aspect of a program that implements radio functions can easily be changed.
In this paper we present a high performance and high flexibility digital software radio platform expressly designed to be used in ionospheric radio links in the HF band. The hardware is based on two Field Programmable Gate Arrays (FPGAs) and several telecommunications specific ASIC devices that can process I/O rates up to 65 MSPS. Configurable modulation, baud rate and carrier frequency constitute the first stage towards a fully digital transceiver skilled in inospheric links.

2. Introduction

A SDR consists of a joint of different digital signal processing functions hosted in a processor to perform real-time communications. A SDR architecture can be defined as a set of functions, components and design rules which main goal is to create a communications system able to dynamically modify any aspect of the signal processing in order to properly adapt to changes on the environment conditions, user requirements, traffic constraints, changes on the communications standards and infrastructure constraints [1],[2],[3].

Thanks to Moore's law, the increased performance of the enabling technologies as A/D and D/A converters, ASICs especially designed for a digital signal processing purpose, programmable logic as FPGAs, DSPs chips and embedded computing has made SDR a real alternative solution to the classical implementation of a radio communications link.
The idea of reconfigurable devices appeared in the early 90's due to the fast technology evolution and the need to develop a device able to adopt the new performance, maintaining the old functionality. Software radio technology can find a solution to the problem of lack of interoperability at the physical layer by developing an architecture that can be reprogrammed to interoperate with different systems utilizing a variety of physical layer technologies.
The flexibility is one of the most important characteristic a SDR platform must support. Flexibility is needed to properly adapt to the inherent diversity of a radio channel communications link. Moreover, compatibility with different devices, efficient bandwidth use and new functionality and standards adoption must also be accomplished [4]. A multimode transceiver is able to make local and global roaming a reality, furthermore there are also cost reduces when introducing new technology and enables multi standard connection with the flexibility to change the application program, such as modem, equalizer, channel codec, synchronization and so on. Flexibility means not only adaptive algorithm where parameters can be modified according to changing constraints, but also applications where the functionality of the signal processing can be altered.
When a SDR system is implemented, a platform which enables separation of software from hardware needs to be chosen. Software needs to be autonomous from hardware in which it runs to track the fast evolution of technology, enabling migration to more powerful systems. Such a kind of problem, suffered in [5], caused a lack of portability of the system to other processing platforms . If a more generic approach had been chosen more powerful coding and modulation techniques would have been allowed.

3. Hardware strategies

A SDR platform main's purpose is to implement a wireless transceiver with the flexibility to make dynamic optimizations discussed above. The ultimate goal in radio receiver is to digitize the RF signal directly at the output of the receiver antenna moving the analog/digital boundary as close to the antenna as possible, as shown in Figure 1 and hence implement all receiver functions in either digital hardware or software processing.

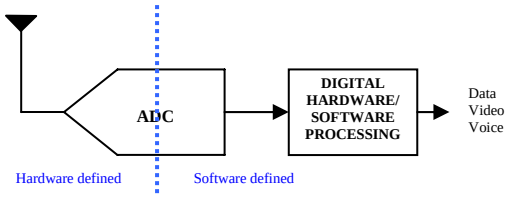

Figure 1. The ideal software radio

Within an ideal SDR architecture all the radio functions from the antenna to the application interface are performed by a high-level software language using generic computing and signal processing hardware. As the aim of SDR transceiver is to process signals to a wide range of frequency bands and channel bandwidth, some improvements have to be introduced in this functional block diagram. The architecture in Figure 2 describes a SDR platform split into a hardware defined subsystem and software defined subsystem. The hardware defined subsystem performs wideband capability, designed to replace many narrow analog receive or transmit frequency conversion chains. In others words, the hardware defined subsystem adds concentration capacity among multiple radio channels, sharing the front end radio stage. The main function of the wideband front end consists of shifting a whole segment of spectrum to a desired IF frequency, instead to shifting individual carriers to baseband, and deliver it to the A/D converter.

An alternative concept to the Heterodyne receiver for the wideband analog front end is presented in [6]. It describes a SDR terminal in which a direct conversion architecture is used. It focus its attention on receiver hardware implementation with system control to avoid signal saturation, nonlinear distortion and DC offset on analog stage to achieve BER specification.

ADC performance is a very important issue when developing digital radio receivers. Several digitizing techniques are used to deal with the wide variety of time signals. One of them is the Nyquist Sampling which digitizes at Nyquist rate and commonly presents large demand on the anti-aliasing filter. Oversampling assures antialiasing filters with more gradual transition and therefore with lower demand. Quadrature sampling reduces the sample rate by a factor of two at the expense of using two ADCs instead of one. Band Pass Sampling method can be used to downconvert a band pass signal at a RF or IF to a bandpass signal at a lower IF. Bandpass sampling allows sampling rates to be much lower than those required for Nyquist Sampling. This means that lower sampling rates and therefore higher performance ADC can be used [7]. As input signals to radio receivers are usually bandpass signals, bandpass sampling is a commonly used technique to digitize directly at RF or IF.

Others ADC specification as stated resolution, signal to noise ratio (SNR), spurious free dynamic range (SFDR), noise power ratio (NPR), power dissipation and intermodulation distortion (IMD) have to be taken into account to properly develop digital radio receivers. A well known tradeoff exists between high sampling rates and high resolution when developing digital radio receivers using digitations at RF or IF. However, as ADC performance continues to improve, digitations at increasingly higher frequencies, maintaining the same level of resolution, will be possible.

A performance characterization, including effective resolution, power consumption, SNR, SFDR and NPR for a wide range of different manufacturer ADCs is made in [8]. The most important factors that affect ADC performance as quantization noise, thermal noise, aperture jitter and comparison ambiguity are shown.

The so-called software defined subsystem, depicted in Figure 2 can be implemented either by digital hardware or software. Many solutions had been proposed to cope with minimization of power consumption, price and die size and maximization of flexibility, efficiency and performance. As none of them find a global solution for all requirements in a SDR system, a balance among all them has to be adopted in order to deliver the best performance answer to applications requirements.

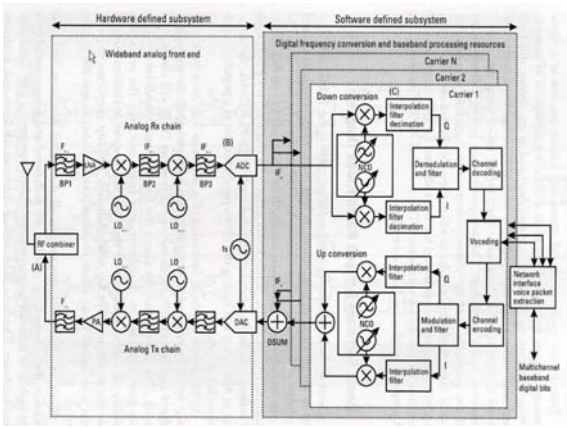

Figure 2. Basic SDR architecture with wideband RF front end.

At one end of the flexibility spectrum we can find the general purpose processor (GPPs), including digital signal processing (DSP) that achieve a very accurate solution to flexibility demands at the cost of lower efficiency, moreover large amount of die area is dedicated to overhead the branch prediction caused by

the fact of being able to cover such a wide range of applications. DSP are developed to implement any arbitrary computation but they are not well suited in cases where big amounts of parallelism can be exploited and deliver poor performance.
In IF processing, requirements are on the order of 500 MIPS/MFLPOS to upwards of 10GFLOPS and in baseband processing on the order of 10 to 100 MIPS/MFLOPS. DSP must be fast enough to perform isochronous operations on this rapid flow of data. If the system is to operate in real time, which is usually the requirement, then the data must be able to get in and out the DSP, which can cause I/O bottleneck problems [15].

A SDR solution based on GPPs takes profit of processor characteristics, such as virtual memory and multithreading to implement a very accurate option to radio requirements. Many processor limitations, such memory and bus time access, make this option a non viable one , nevertheless. So, an hybrid solution between GPP and DSP, turns out to be the more efficient one [9] .

At the other end of the flexibility spectrum application specific integrated circuits (ASICs) can be found. ASICs achieve the highest possible performance at the lowest silicon, area and power cost because they are explicitly tailored for specific application. They can be useful when working with very well defined and widespread applications.
Between GPPs and ASICs are the field programmable gate arrays (FPGAs) which are flexible hardware that can be structured to fit in the application requirements, exploiting the concurrency in the computation. A FPGA consists of an array of gates that can be reprogrammed allowing the silicon resources in single device to be time shared among several functions. FPGA gates are structured in an array of logic blocks and a interconnection architecture to connect all of them. Every logic block contains flip-flops and look up tables (LUTs). These last ones are usually realized by using static RAM. LUTs can implement any Boolean function and work as little distributed RAM as well. By interconnecting LUTs and flip-flops not only combinatorial functions but also sequential functions can be realized [10], [11]. In addition FPGAs have dedicated blocks to work as block RAMs and new generation FPGAs also have specific hardware, such as multipliers, specially tailored for digital signal processing functions.
Comparing FPGAs to ASICs, they are quite more flexible at a cost of lower efficiency. Although FPGAs merge the flexibility of software programmable processor with the advantages of ASICs, never reach the power, clock rate or die size that could be achieved in a custom chip designed for a particular task.

An optimal SDR should be able to switch its hardware implementation to the needs of the current system, minimizing power consumption, silicon area, system latency and maximizing throughput to reach performance similar to an ASIC.
Configurable computing machine (CCM) can provide real-time paging of algorithms on hardware. CCMs have static hardware to implement communications-oriented algorithms such as multiplications and filtering, which result in efficient radio designs [12].
A reconfigurable architecture which follows the ideas of CCM is the field programmable function array (FPFA). FPFAs have appeared to implement DSP tasks at a lower power consumption, reaching high performance levels by exploiting its inhere parallelism [13],[14]. They are commonly produced by an embedded GPP subsystem plus and embedded FPGA plus and embedded IO subsystem. With this structure , FPFA combines the advantage of implicit parallelism of an FPGA with the arithmetical richness of a microprocessor. However, the great flexibility results in a huge quantity of control signals and tedious programming process.

4. Wideband SDR transceiver.

The main objective of this project is to develop a digital radio transceiver defined by software in order to switch channels, change modulations schemes, codification and channel equalization to properly adapt to the ionospheric channel, which is an extremely changing and noisy channel.
A reconfigurable platform upgraded by loading new software routines instead of replacing expensive hardware has been chosen. It's main purpose is to deal with HF (3-30 MHz) communications, therefore a simple antialiasing BPF from 3 to 30 MHz plus a Low Noise Amplifier (LNA) is designed as analog front-end (Figure 3).

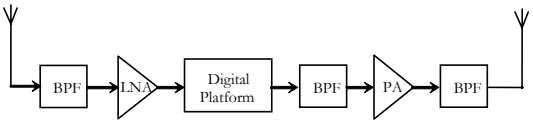

Figure 3. Transceiver global scheme

This platform can also be used for wireless applications with bandwith lower than 30 MHz, as sampling rate is 60 MHz, to prevent from aliasing. In this case undersamplig techniques must be utilized in order to recover the RF band pass signal, as well as an analog front end with an heterodyne stage.

The design presented in this paper stands out for its high performance, high degree of reconfigurability, wide range of possible interconnections with other devices and platforms and the capacity to be fully upgraded just changing the software, so it seems to fulfill the requirements of a so-called SDR.
As depicted in Figure 4 the transceiver is divided in two parts : the receiver chain and the transmitter chain. In each of them there is a FPGA (Virtex XCV400 from Xilinx) which behaves as the core of the subsystem. The FPGA enables interconnection among all devices in and out the platform and computes digital signal processing, as well.

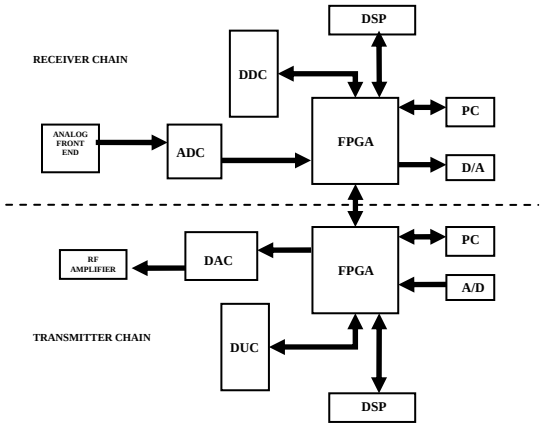

Figure 4. Wideband SDR transceiver architecture.

As FPGA assures global interconnection, high grade of flexibility can be guaranteed.
The most restricting constraints in digital transceivers are dynamic range and frequency rates, so A/D converters must be chosen carefully. In this case we have opted for AD9764 from Analog devices, which is an A/D converter with sampling rate up to 65 MSPS with 14 bits of resolution. In Table 1 a measure of performance and SFDR is made. The measure is performed with two carriers. One of them at 24 MHz at full scale range (7,5 dBm) and the second one at 10 MHz with power from -50dBm to -65dBm. On the first column we can observe the power relation between carriers at the input of the A/D converter, on the second one the same measure at the output of the A/D converter and on the last one the SFDR between the 24 MHz carrier and its highest harmonic.
We can conclude that the A/D converter in presence of an interference signal at full scale range has a good performance beneath 70 dBc.

dBc (analogical)	dBc (digital)	SFDR (digital)
57,5	59,4	58,7
63,5	64,6	58,7
69,5	71,3	58,1
72,5	76,2	58,4
75,5	79,2	58,1

Table 1. Measures of performance and SFDR of the AD6644.

An other important part on the receiver chain is the Digital Down Converter (DDC) from Analog Devices (AD6620), which has decimating filters and Numerically Controlled Oscillators (NCO) in order to downconvert any HF signal to baseband, with a maximum sampling rate of 65 MSPS.
On the transmitter chain the main function is made by the Digital Up Converter (DUC) which is analogous to the DDC and by the D/A AD9764 from Analog Devices which supports a sampling rate of 125 MSPS with 14 bits resolution.
In Figure 5 a photograph of the ultimate prototype is depicted.

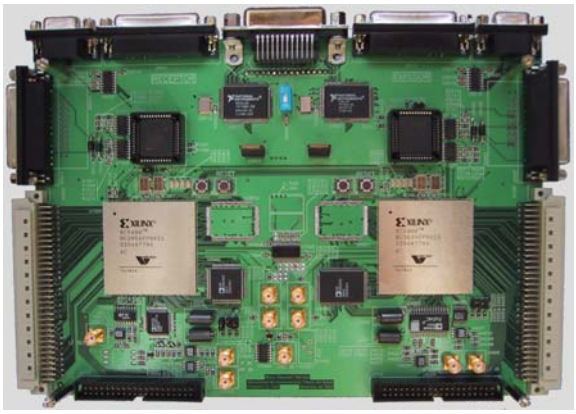

Figure 5. SODIO Platform

5. Preliminary software

The first steps has been done in the transmitter chain. A software module interfaces the SDR platform with a PC in order to enable the data and configuration flow. A connection with a PC through parallel port is performed, however serial and GPIB connections could be done as well. The FIFO synchronizes the data flow from the PC and delivers it to the MODULATOR, which changes the bits into symbols. This modulated base band data feeds the UPCONVERTER through its interface. That ASIC interpolates, filters whit raised cosine and upconverts the data into the selected carrier frequency. The sampling rate at its output is 60 MSPS. That flow of

data is send to the A/D converter through the TX FPGA (Figure 6).

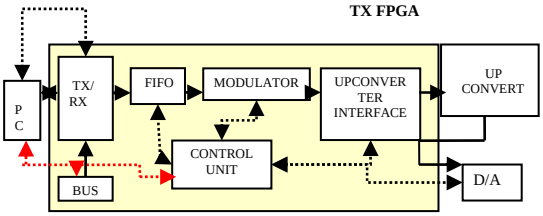

Figure 6. Transmitter chain

The configuration flow is send directly to the CONTROL UNIT which main function is to reconfigure the whole system in order to supply the best performance to carry through the requirements. That module has previously stored in memory several configurations and now has to chosen among all them the one that fulfills the requirements.

6. Preliminary results

We present two preliminary results examples of the above application. The figures have been taken with HP89440 vector analyzer from Agilent.

The first one (Figure 7) is a 8 MHz carrier modulated with a 8PSK modulation. The bit rate is 50 Kbps and the symbol filter is a Root Raised Cosine of 128 coefficients, $\alpha = 0.22$.
The second one (Figure 8) matches with a 8 MHz carrier modulated with a 16QAM modulation. The bit rate is 58Kbps, with equal symbol filter.

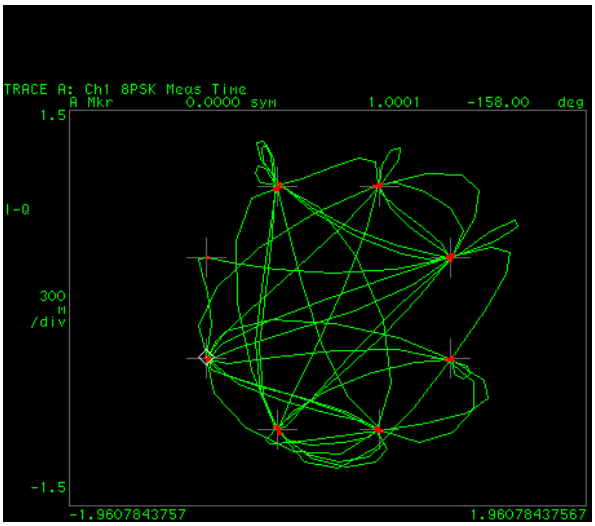

Figure 7. 8-PSK

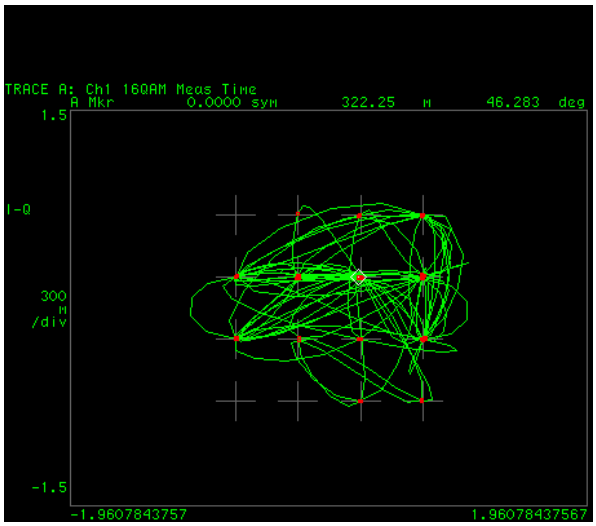

Figure 8. 16-QAM

7. Future works

Once the hardware has been tested and basic software has been implemented, several works have to be made. Firstly, a platform like Figure 3 must be developed to fit in a real ionosferic communications scenario. Equalization, synchronization and demodulation modules must be implemented in the receiver chain.
Moreover we want to interconnect SODIO platform with a DSP board in order achieve a better performance and efficiency in baseband processing.

8. Bibliography

[1] Joe Mitola, “The software radio architecture”, IEEE Communication Magazine, pp.26-28, May 1995.
[2] J. Mitola, “Software Radios. Survey, Critical Evaluation and Future Directions”, IEEE AES Systems Magazine, April 1993.
[3] Walter W. H. W. et al., “Software-Defined Radios: Facets of a developing Technology”, IEEE Personal Communications, pp. 38-44, April 1999.
[4] Vanu G. Bose, “The impact of Software Radio on wireless networking”, in Mobile Computing and Communications Review, January 1999.
[5] Lackey and Upmall, “Speakeasy: The Military Software Radio”, IEEE Communications Magazine, May 1995.
[6] Hiroshi Tsurumi and Yasuo Suzuki, “Broadband RF Stage Architecture for Software-Defined Radio in Hanheld Terminal Applications”, IEEE Communications Magazine, pp. 90-95, February 1999.
[7] Jeffery A. Wepman, “Analog-to-Digital converters and Their Applications in Radio Receivers”, IEEE Communications Magazine, pp. 39-44, May 1995.
[8] Robert H. Walden, “Performance Trends for Analog-to-Digital converters”, IEEE Communications Magazine, pp. 96-101, February 1999.
[9] Philip Mackenzy et al., “Selecting appropriate hardware for software radio systems”.

[10] Mark Cummings et al., “FPGA in the Software Radio”, IEEE Communications Magazine, pp. 108-112, February 1999.
[11] John Villasenor and Brad Hutchings, “The flexibility of configurable computing”, IEEE Signal Processing Magazine, pp.67-84, September 1998.
[12] Srikathyayani Srikanteswara et al., “An overview of configurable computing machines for software radio handsets”, IEEE Communications Magazine, pp. 134-141, July 2003.
[13] Paul M. Heysters et al., “Reconfigurable architecture for handheld devices”, in Proceedings of the 3D Progress Workshop on embedded Systems, October 2002.
[14] H. Bouma, Design and implementation of an FPFA, M.Sc. Thesis, July, 2001.
[15] Rupert Baines, “The DSP bottleneck”, IEEE Communications Magazine, pp. 46-54, May 1995.