FPGA Based Implementation of SDR Waveforms in Digital Receiver Board

V.I. Beatrice Sylviaz and Dr.A. Sumathi

***Abstract*---**In the development of SDR waveforms, the trend goes away from straight programming in dedicated languages like C for DSPs, C++ for GPPs or Verilog and VHDL for FPGAs to rapid prototyping with a model based approach and code generation. The advantages of those developments are reduced time to implement waveforms for SDR platforms and to work with the high-level development environment of developers' choice, independent of the underlying hardware. Software-defined radio (SDR) architectures typically include general-purpose CPUs (GPPs), digital signal processing (DSP) ASSPs and FPGAs that process different waveforms, functions, and algorithms. My Research work aims at developing a Software Defined Radio Waveform Tool in MATLAB for analyzing all these Waveforms and functions. Now the user can select the waveform with minimum peak side lobe as per the application requirement. Then the Digital Waveform Generation Blocks are implemented in Digital Receiver Board using user programmable FPGA Platform. For modeling and code generation MatLab/Simulink with Real-time Workshop, HDL Coder and Xilinx System generator for DSP was used

I. Introduction

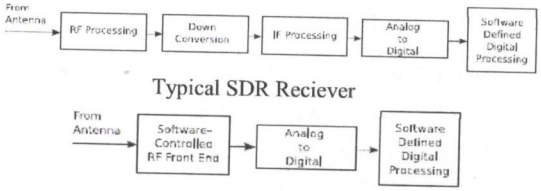
IN the past FPGAs were used as a convenient interconnect layer between chips in a system. In software defined radios (SDRs), FPGAs are being used increasingly as a general-purpose computational fabric implementing signal processing hardware that boosts performance while providing lower cost and lower power. Typical implementations of SDR modems include a general purpose processor (GPP), digital signal processor (DSP), and field programmable gate array (FPGA). However, the FPGA fabric can be used to offload the GPP or DSP with application specific hardware acceleration units. Soft-core microprocessors can have their core extended with custom logic, or separate hardware acceleration co-processors can be added to the system. Furthermore, with general purpose routing resources available in the FPGA, these hardware acceleration units can run in parallel to further enhance the total computational throughput of the system. Today waveforms for SDRs should be portable between different platforms and they should have short development times beside the interoperability requirement. The actual development of a waveform is done in multiple steps. The first step is the simulation in environments like MatLab/Simulink. After simulation the waveform is reimplemented on software architectures like the SCA or GNU Radio, or it is redesigned in platform specific languages like C, C++, Verilog or VHDL.

V.I. Beatrice Sylvia, Research Scholar, Anna University, Chennai.
Dr.A. Sumathi, Professor & Head, Department of ECE, Adhiyamaan College of Engineering, Hosur.

II. Software Defined Radio

The concept behind SDR is that more of the waveform processing can be implemented in reprogrammable digital hardware so a single platform can be used for multiple waveforms. With the proliferation of wireless standards, future wireless devices will need to support multiple air interfaces and modulation formats. SDR technology enables such functionality in wireless devices by using a reconfigurable hardware platform across multiple standards.

SDR is the underlying technology behind the Joint Tactical Radio System (JTRS) initiative to develop software-programmable radios that can enable seamless, real-time communication across the U.S. military services, and with coalition forces and allies. The functionality and expandability of the JTRS is built upon an open architecture framework called the software communications architecture. The JTRS terminals must support dynamic loading of any one of more than 25 specified air interfaces or waveforms that are typically more complex than those used in the civilian sector. To achieve all these requirements in a reasonable form factor requires extensive processing power of different kinds.



Typical SDR Reciever

Advanced SDR Receiver

Fig 1 Block Diagram of SDR Receiver

III. Platform Overview

The Universal Software Radio Peripheral (USRP) is not a Software Radio Platform by itself, but it provides an interface from a host PC to several Radio Frequency (RF) front-ends. The actual baseband processing is calculated on the General Purpose Processor (GPP) running on a PC. Using FPGA resources for hardware acceleration can be done in several ways. However, there are three basic architectures: Custom instructions, custom peripherals as coprocessors, and dynamically reconfigurable application specific processors. These hardware acceleration methods have different features and unique benefits. Understanding how and where to use each of these helps the system architect better use the FPGA

ISBN 978-0-9888421-1-3 © 2013 Published by Coimbatore Institute of Information Technology

resources for offloading the DSP and GPP in a SDR application.

Fig 2 SDR Functions across GPP, DSP and FPGA

The Digital Receiver is the digital counterpart for the microwave components to perform waveform generation and down conversion. Since Digital Receiver can be defined through software it needs a software tool for analyzing the radar waveforms. This software needs to be made user-friendly by developing the GUI (Graphical User Interface), which would allow the operations and the tests to be performed by the user without the prerequisite knowledge of the embedded software.

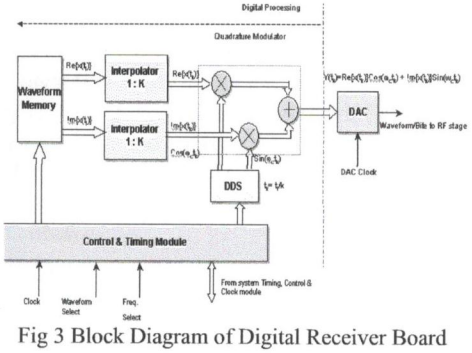


Fig 3 Block Diagram of Digital Receiver Board

The DDS core in this block provides a quadrature (sin and cos) local oscillator signal to the quadrature modulator, were the I and Q data are multiplied by the respective phase of the carrier and summed together, to produce a SDR waveform data stream. All of this occurs in the digital domain, and this digital data stream applied to the DAC to get analog SDR waveform.

IV. Waveform Overview

The implemented waveform should use the whole capacities of the used platforms, therefore the implementation of a high data rate OFDM waveform was chosen. As a template for this, the physical layer of IEEE 802.11g was taken with the following modifications: fixed-data rate, no data scrambling and fixed uni-directional transmission.

Different to broadcast systems, there is no continuous transmission, but data packets. To detect a data frame, the Physical Layer Convergence Protocol (PLCP) is used. It is mainly a preamble for synchronization purpose and consists of the short training sequence (STS) and the long training sequence (LTS). These are periodical sequences, defined by the standard, and are well suited for detection and synchronization with correlation functions. Therefore, an Schmidl and Cox synchronization algorithm will be implemented in the receiver.

The synchronization takes care of frequency, phase and clock mismatch and indicates the starting point for the FFT, needed for the OFDM demodulation. The SIGNAL and DATA fields are OFDM symbols that are differentiated in the data they transmit. The SIGNAL field transmits information about the DATA field as coding scheme, modulation scheme and length of the payload. Therefore, the SIGNAL field is per default BPSK modulated with a Forward Error Correction (FEC) of coding rate 1=2. As mentioned above, in this implementation only one data rate for the DATA field was realized: QPSK modulation with gray constellation ordering.

On the receiver side we perform a timing synchronization based on the correlation algorithms by Schmidl and Cox [15]. The frequency synchronization is split up into two parts: At first, the frequency is corrected to match the OFDM carrier raster. Afterwards the complex data stream is Fourier transformed and a second synchronization is applied to shift the carriers in the right positions. The channel equalization relies on the 4 pilots that survey the channel pointwise. A linear regression determines the channel influence over one complete OFDM symbol. The symbols are demodulated, deinterleaved and passed through a Viterbi decoder.

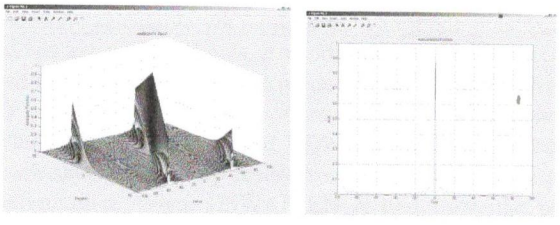


Fig 4 SDR Waveforms with Minimum Peak Sidelobes

A. *Minimum Peak Side Lobe Codes*

A binary code that yield minimum peak side lobes but do not meet the Barker condition is called as Minimum Peak Side lobe Codes (MPS Codes). Results up to N<=69 were reported here. Codes with a peak side lobe of 2 were reported for N<=28. The MPS codes reported for 28<N<48 and N=51 have a side lobe level of 3, and the MPS codes of length N=50 and 52<=N<=69 have side lobe level of 4. It seems that for any peak side lobe level there is a limit of the maximal value of M for which a binary sequence with that side lobe level exists.

i. Merits of MPS Codes:

- Relatively low complexity (no multiplications are needed at the receiver)
- The simulated codes are those that have, the minimum integrated side lobes

ii. Demerits of MPS Codes:

- These codes are known only for limited values of N
- Doppler Intolerance

ISBN 978-0-9888421-1-3 © 2013 Published by Coimbatore Institute of Information Technology