THE ARCHITECTURE OF A SOFTWARE RADIO DVB-S2 RECEIVER

Panayiotis Savvopoulos, University of Patras, Dept. of Electrical and Computer Engineering, 26500 Patras, Greece, Phone: +30-2610996483, e-mail: psavvop@upatras.gr

Nikolaos Papandreou, University of Patras, Dept. of Electrical and Computer Engineering, 26500 Patras, Greece, Phone: +30-2610996485, e-mail: npapandr@ece.upatras.gr

Theodore Antonakopoulos, University of Patras, Dept. of Electrical and Computer Engineering, 26500 Patras, Greece, Phone: +30-2610996487, e-mail: antonako@upatras.gr

Abstract

This paper presents the architecture of a DVB-S2 receiver prototyping environment that is based on software defined radio techniques. DVB-S2 comprises the newest European standard for broadband satellite communications. The software defined radio approach enables the adaptation of the receiver to different operational modes e.g. CCM, VCM, ACM, and the support of multiple services. The prototyping methodology as well as the mapping approach of the various DVB-S2 receiver functional modules are described.

1. Introduction

Software defined radio (SDR) is an enabling technology for future commercial, multimode and reconfigurable communication terminals. DVB-S2 comprises the newest European standard for broadband satellite communications services that manages to apply new achievements in the field of modulation and coding techniques so as to meet the requirements of today's needs for satellite broadcasting and unicasting communications [1]. DVB-S2 technology has been specified around the key concept of achieving high transmission performance, while providing system and functional flexibility and keeping the complexity of the terminal receivers at acceptable levels. These functional characteristics are based on the versatility of the DVB-S2 physical layer with frame-by-frame adaptability according to channel conditions. DVB-S2 supports three modes of operation, Constant, Variable and Adaptive Coding and Modulation (CCM, VCM and ACM), with differentiated levels of signal robustness according to predefined criteria. It accommodates the widely used MPEG transport stream as well as generic streams of constant or variable packet lengths. DVB-S2 also supports two modes of broadcasting services, the Backwards Compatible mode, where the signal can be partially decoded by legacy DVB-S receivers, and the Non-Backwards Compatible mode, that takes advantage of the DVB-S2 broadcasting features.

The adaptability of a DVB-S2 terminal to the different modes of operation and to future advances in services and/or technology is made possible by using reconfigurable hardware, such as digital signal processors (DSP) and field programmable gate arrays (FPGA). The functional reconfigurability is also combined with SDR techniques for supporting different operational modes and multiple services. The improved terminal performance over a wide range of transmission techniques relies on the receiver’s tendency to be wideband in nature and as linear as possible, so as to minimise the distortion of the incoming signal. The above feature is by no means a straightforward task, with a lot of research being performed in this area [2].

In this work, we present the architecture of a SDR DVB-S2 receiver and its various processing modules, starting from the IF incoming signal up to the recovery of the output IP data packets. Several aspects are also presented regarding the implementation approach of the receiver subunits. The receiver implementation is targeted for a powerful and flexible reconfigurable hardware prototyping platform that consists of multiple DSPs and reprogrammable circuits.

The paper is organized as follows. Section 2 describes the DVB-S2 receiver architecture and provides details of the various receiver modules. Section 3 presents the prototyping SDR platform and describes the used implementation methodology. Finally, Section 4 describes several mapping perspectives regarding the various functional modules and their prototyping into the software radio oriented platform along with respective implementation information.

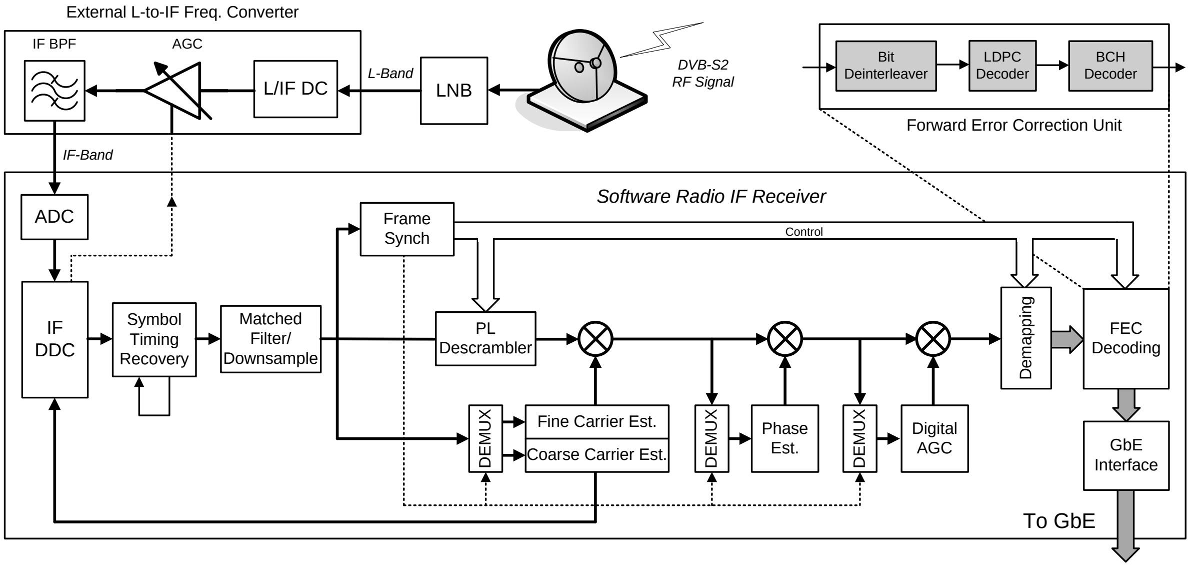

Figure 1: DVB-S2 Terminal Receiver Architecture

2. DVB-S2 Receiver Architecture

A typical architecture of a complete DVB-S2 terminal that is based on a software defined radio IF receiver is presented in Figure 1. The basic components of the presented receiver are: the outdoor unit (ODU), the external L-to-IF band frequency converter and the SDR IF receiver. The ODU consists of a parabolic dish antenna and the Low Noise Block (LNB) which converts the received RF signal into the respective L-Band (950-2150 MHz). Then the signal is further down-converted to the IF-Band (70 MHz) by an external analog down-converter, which includes output power control and fixed bandpass filtering operations. Incoming signal adjustment is achieved by an automatic gain controller (AGC), implemented in the digital down-converter (DDC) of the IF digital receiver. The AGC is driven by a power measurement circuit, implemented in the DDC, in order to properly adjust the received signal to the dynamic range of the used ADC. At the output of the L-to-IF down-converter, the ADC digitizes the incoming signal and feeds the various processing modules of the digital receiver. This work is focused on the architecture of the IF digital receiver that implements the appropriate synchronization, demodulation and decoding functions.

2.1. Digital Down-Converter

The digital down-converter (DDC) is the first digital processing module of the SDR receiver that processes and converts the IF signal into the respective baseband signal, by using two cascaded stages of frequency conversion. The first stage mixes the input signal samples with a quadrature oscillator of a fixed frequency equal to the nominal $F_s/4$ (F_s stands for ADC sampling frequency). The resulting in-phase (I) and quadrature (Q) components are then low pass filtered in order to remove signal images of higher frequencies. Then, both signal components drive a programmable frequency converter that utilizes a numerical control oscillator (NCO) and a complex multiplier, both running at $F_s/2$. The NCO generates the variable frequency complex carrier samples used for the mixing of the respective input signal samples in the complex multiplier structure. The NCO carrier frequency is controlled by the decisions of the coarse carrier frequency recovery (CFR) unit described below. After the second frequency conversion stage, a sample rate conversion takes place that aims of producing the necessary rational rate change factor so that the sampling rate at the DDC output is adjusted to an integer multiple of the nominal symbol rate. In our implementation, this factor is either 2 or 4, and is related to the acceptable sample rate at the input of the next baseband processing unit, which is the symbol timing recovery (STR).

2.2. Symbol Timing Recovery

The first synchronization function performed inside the IF receiver is the symbol timing recovery (STR). The functionality of this module is to track the symbol rate fluctuations introduced during the signal transmission and the generation of up-sampled synchronized samples to the incoming symbols. The rate is 2 or 4 times the nominal symbol rate and is equal to DDC output sampling rate. STR is implemented as a second order feedback loop utilizing a Farrow structured cubic interpolator along with the non-data-aided (NDA) timing error detector (TED) proposed by Gardner [3], which provides the loop with the required error signal. Gardner TED is capable of operating under random symbols and unknown carrier frequency offset error without the need of precise carrier synchronization. At the output of the STR interpolator, the synchronized stream of output samples is matched filtered and finally downsampled leading to a rate of one sample per symbol that feeds the rest of the DVB-S2 receiver.

2.3. Frame Synchronization

After symbol timing recovery has reached steady state, the next function to be performed is frame synchronization. This is done by searching the physical layer header using an appropriate correlator that operates on a symbol-by-symbol basis. Using differential detection, accurate frame synchronization is possible even in the presence of strong carrier frequency errors [4], [5]. An appropriate finite state machine controls the frame synchronization unit during acquisition and normal operation. Based on the correct framing alignment, it is possible to demultiplex (DEMUX) the pilot symbols from the incoming frame in order to drive the various pilot-aided carrier and phase synchronization subunits.

2.4. Carrier Frequency Recovery

Carrier frequency recovery (CFR) is invoked directly after frame synchronization has detected the start of the transmitted frames and thus the locations of the respective pilot fields. CFR is performed in two sequential steps, which both use the known pilot symbols, regularly repeated in the form of fields during payload transmission of DVB-S2 frames. The first step of carrier synchronization consists of a coarse carrier recovery mechanism which compensates large frequency offset errors up to several MHz and is implemented as a second order feedback loop based on a delay-and-multiply (DM) frequency error detector [6]. The compensation of this loop is applied through the NCO that is implemented in the DDC. The second step completes the carrier synchronization process, since it performs fine carrier recovery that is initiated after the convergence of the first carrier recovery loop, where the frequency offset error is in the order of a few hundreds of kHz. Fine recovery deploys a feedforward estimation algorithm, derived from an alteration of the L&R technique [7], and an integrator look-up table for the respective frequency offset removal.

2.5. Phase Recovery

Phase recovery needs to cope with residual frequency offset resulting from both carrier recovery procedures. In case of low order modulation transmissions (QPSK or 8PSK), a pilot-assisted maximum-likelihood (ML) feedforward estimator is used for computing the average phase of each pilot field [6]. Accordingly, phase compensation is based on the interpolation between the estimations of two consecutive pilot fields, for determining the evolution of the phase during the transmission of intermediate data symbols. When high order modulations of 16 or 32 APSK are used, an additional phase synchronizer is needed which consists of a closed loop based on the NDA phase error detector of Q-th power (Q=3 for 16APSK and Q=4 for 32APSK) [4], which is initiated after the ML feedforward estimator has reached its steady state. This NDA feedback loop is located after the digital amplitude control unit, described in the next subsection.

2.6. Digital Automatic Gain Control

Digital automatic gain control (DAGC) takes place after the first level of phase recovery and is based on a pilot-assisted or data-aided vector tracker mechanism (DA-VT) [8], which utilizes the known pilot symbols for the determination of the amplitude normalization multiplication factor. Therefore the DAGC is activated during the transmission of pilot fields and it is frozen during data symbol transmissions when the proper amplitude adjustments are also made.

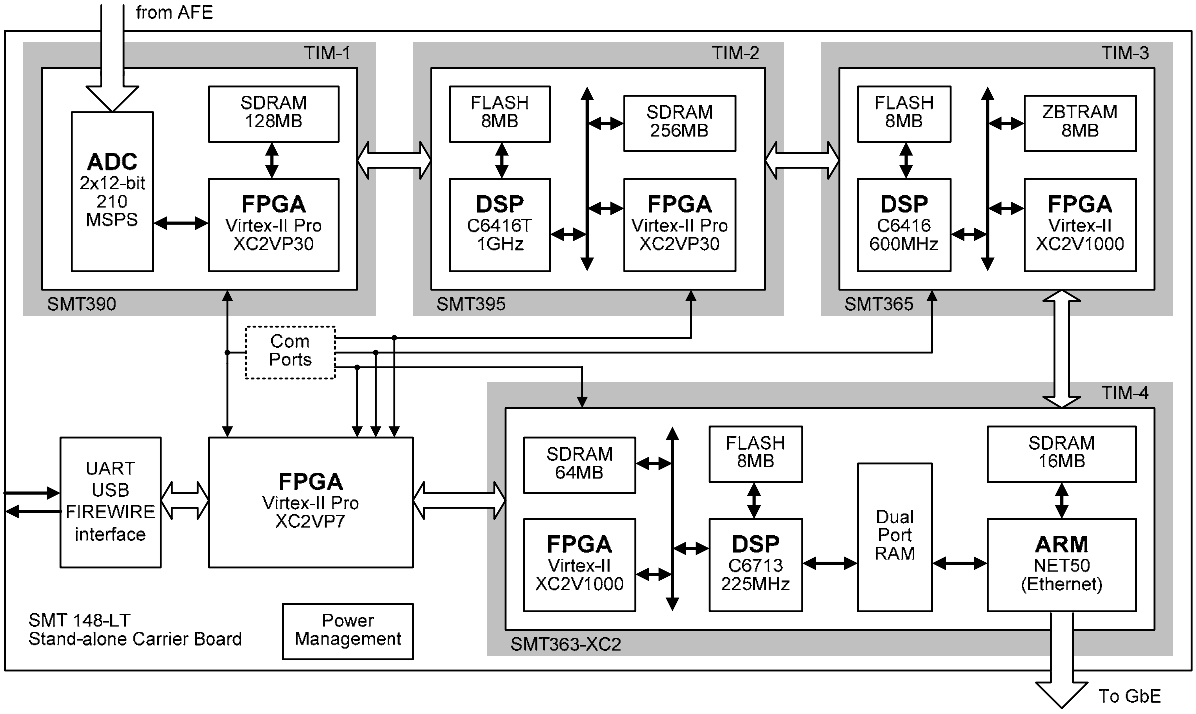

Figure 2: The Software Defined Radio Prototyping Platform.

2.7. Demapper and FEC Decoder

After the various synchronization parameters have been compensated, the input complex symbols (I, Q symbol sequence) of the incoming frame are demapped into the data bit sequence based on the used modulation efficiency. The DVB-S2 standard supports the following modulation formats: QPSK, 8PSK, 16APSK and 32APSK. The resulting bit sequence is grouped into complete coded data frames (blocks) with the control signals of the frame synchronization subunit. The DVB-S2 standard utilizes a robust forward error correction (FEC) scheme. In particular, the FEC decoding is carried out by the concatenation of an LDPC inner decoder and a BCH outer decoder. A block deinterleaving function is also preceded for the 8PSK, 16APSK and 32APSK modulation formats. Various rates for the inner code are also defined based on the channel conditions [1, Table 12]. The length of the FEC coded block is either 64800 (normal frame) or 16200 (short frame) bits according to the type of the used application.

2.8. IP and data protocols.

DVB-S2 supports two major transport modes, the MPEG Transport stream and the DVB-S2 Generic stream. In the first case, the MPEG Transport stream, three different data encapsulation/decapsulation procedures are supported, namely data ping, multiprotocol and ultra lightweight. These protocols are also used by the DVB-S, while for the DVB-S2 Generic stream only the Generic Streaming Encapsulation is supported. The LDPC/BCH decoded frames are used for extracting the MPEG frames and the IP datagrams are recovered before they are transmitted to the Ethernet network.

3. Software Radio Platform

The implementation of the SDR DVB-S2 receiver is based on a versatile and powerful hardware prototyping platform based on multiple DSPs and FPGA circuits. The prototyping platform, which is shown in Figure 2, is the stand alone carrier board SMT 148-LT by Sundance [9] that consists of four Texas Instruments Modules (TIM). This stand alone carrier board provides the physical connections between the TIMs and includes a Virtex-II Pro (XC2VP7) device that incorporates a PowerPC processor for management and control purposes.

TIM-1 is used to digitize and process the incoming IF signals by a high-speed 12-bit ADC (up to 210 MSps) and a Virtex-II Pro (XC2VP30) FPGA with a PowerPC processor. An SDRAM of 128MB is also available for storage purposes. TIM-2 combines a powerful C6416T DSP running at 1GHz along with a Virtex-II Pro (XC2VP30) FPGA. Due to the high performance DSP, the TIM-2 module is used for implementing time critical tasks of the DVB-S2 receiver architecture, such as symbol timing and frame synchronization. TIM-3 also incorporates a C6416 DSP running at 600MHz and a powerful Virtex-II FPGA of 1 million gates suitable for the implementation of demanding designs, such as the LDPC and BCH decoder circuit modules. TIM-4 combines a C6713 DSP running at 225 MHz, a Virtex-II FPGA and a NetSilicon ARM-chip with integration of a MAC controller for connection to an Ethernet network. The above hardware platform provides the flexibility to adapt the functionality of the custom logic to various modes of operation and also serves as a standalone system in order to demonstrate the implementation of the DVB-S2 receiver.

For the development of the various receiver functional modules, a carrier board with PCI bus is used, which provides data exchange between a host application and a master TIM. Through the above interface the custom implementation modules of the various TIMs can interact with and be controlled by customly design host application. The described setup enables the systematic development of the DVB-S2 receiver using a methodical prototyping process that leads to fast debugging and easy system optimization.

3.1. The Prototyping Methodology

The prototyping methodology is based on a top-down process that determines how the receiver's architecture is mapped into hardware/software modules in the prototyping platform starting from a high-level simulation model that describes the system functional requirements.

Initially, the system specifications are defined based on the functionality and complexity of the various modules along with the design requirements and restrictions imposed by the hardware platform. This step also involves the system's functional decomposition into circuit components and protocol units, as well as the definition of the signal and data interfaces between each component and its environment according to the architecture of the prototyping platform. Based on this information, an analytical model of the DVB-S2 receiver was developed using the MATLAB environment. Using the Simulink and Stateflow tools, we are able to build a multi-domain high-level model that combines the complete dataflow using signal processing algorithms and communication protocol entities based on FSM modules. We have build a complete model of the DVB-S2 receiver in order to evaluate well-known as well as custom algorithms. The functionality of the various modules is verified and optimized by simulations using fixed-point arithmetic in order to account for the target hardware implementation. Thus, an initial but accurate estimation of the implementation's complexity was performed. The high-level simulation environment was used also for generating a set of testing scenarios.

After the model design was completed, the next step was to map the model components into hardware/software modules of the prototyping platform. The progressive substitution of the model components with circuit/microcode modules was realized using automated code generation tools and custom implementation designs. The MATLAB environment provides special libraries for automated code generation of basic functions, i.e. filters, NCO operations etc., supporting selected families of TI, Inc. and Xilinx, Inc. devices. In the simulation model, the translated model blocks are replaced by special functions that provide communication between the simulation workspace and the circuit modules in the prototyping platform. This is realized via a PCI-based custom interface that supports data exchange and synchronization between the host application and the hardware platform. A custom interface logic has been developed for the effective exchange of variable length data blocks between the hardware platform and a host application. The interconnection is managed through the utilization of dedicated high-speed channels and communication ports [10] over the PCI bus interface, and is totally based on a command oriented and a handshaking mechanism supported by both communicating ends.

4. Functional Mapping

According to the above functionality and the available resources depicted in Figure 2, the mapping of the various receiver modules was performed as described below. The DDC is implemented in the FPGA of TIM-1, while the STR module along with the matched filter, the frame synchronizer and the coarse carrier recovery circuit in the FPGA device of TIM-2. Phase recovery (coarse and fine) and digital amplitude control is implemented in the DSP of TIM-2, while the demapper which retrieves and organizes the blocks of the coded bits is implemented in the DSP of TIM-3. Deinterleaving and

LDPC/BCH decoding is implemented in the FPGA of TIM-3. The last TIM module is used for all protocol processing and the decapsulation functions, along with the required TCP/IP protocol stack.

In order to estimate the maximum operational clock of each processing stage, we use the maximum symbol rate supported by the previously described receiver architecture, for various modes of operation. Table 1 presents the maximum symbol rates and the respective rates. These values correspond to a bandpass bandwidth of 36 MHz, a carrier frequency of 70 MHz and a maximum A/D converter sampling rate of 210 MSps. Apart from symbol rates, the rates at the DDC and matched filtered downsampler output are also given.

Table 1: Maximum Supported Symbol Rates by the Software Radio DVB-S2 Receiver

Roll-Off Factor	**Maximum Symbol Rate (MBaud)**	**Sampling Rate (MSps)**		
		ADC Output/ DDC Input	**DDC Output/ STR Input**	**MF Downsampler Output**
0.20	30	180	120	30
0.25	28.5	199.5	114	28.5
0.35	26	208	104	26

5. Conclusions

In this work the architecture of a software defined radio DVB-S2 receiver has been presented. The various functional modules have been presented along with the architecture of a versatile prototyping platform. The presented environment allows the adaptation of the receiver's functionality to different operational modes and the support of multiple services.

6. Acknowledgment

This work was supported by the European Regional Development Fund (ERDF) and the Greek Ministry of Development – General Secretariat for Research and Technology (GSRT).

7. References

[1] ETSI EN 302 307 V1.1.2, "Digital Video Broadcasting (DVB); second generation framing structure, channel coding and modulation systems for broadcasting, interactive services, news gathering and other broadband satellite applications," June 2006.

[2] E. Buracchini, "The Software Radio Concept", *IEEE Communications Magazine*, vol. 38, no. 9, pp. 138-143, Sep. 2000.

[3] F. M. Gardner, "A BPSK/QPSK timing-error detector for sampled receivers", *IEEE Trans. Commun.*, vol. 5, pp. 423-685, May 1986.

[4] E. Casini, R. De Gaudenzi, and A. Ginesi, "DVB-S2 modem algorithms design and performance over typical satellite channels," *Int. J. Satell. Commun. Network.*, vol. 22, pp. 281-318, 2004.

[5] F.-W. Sun, Y. Jiang, and L.-N. Lee, "Frame synchronization and pilot structure for second generation via satellites," *Int. J. Satell. Commun. Network.*, vol. 22, pp. 319-339, 2004.

[6] U. Mengali and A. D' Andrea, "Synchronization Techniques for Digital Receivers," *Plenum Press*, 1997.

[7] M. Luise and R. Reggiannini, "Carrier frequency recovery in all digital modems for burst mode transmissions," *IEEE Trans. Commun.,* vol. 43, no. 2/3/4, pp. 1169-1178, Feb/Mar/Apr 1995.

[8] R. De Gaudenzi and M. Luise, "Analysis and design of an all-digital demodulator for trellis coded 16-QAM transmission over a nonlinear satellite channel," *IEEE Trans. Commun.,* vol. 43, no. 2/3/4, pp. 659-668, Feb/Mar/Apr 1995.

[9] Sundance Multiprocessor Technology Ltd., *SMT 148 User Manual, Version 1.4*, Aug. 2006.

[10] Diamond User Guide, Sundance Edition V3.0, June 2005.