A Software-based Instrument for Testing and Monitoring Multi-processing Communications Devices

P. Savvopoulos, A. Kotsopoulos, M. Varsamou, N. Papandreou† and Th. Antonakopoulos
Department of Electrical and Computers Engineering
University of Patras, Rio-Patras, 26500, Greece
Email: {psavvop, mtvars, antonako}@upatras.gr, {akotsop, npapandr}@ece.upatras.gr

***Abstract*—This paper presents a software-based instrument for testing and monitoring the behavior of complex communications systems that employ a number of interconnected DSPs. The instrument is based on a dynamic architecture that incorporates a set of distributed software probes for signal capturing and a central data acquisition unit for acquiring and post-processing of data. A high level application that is executed on a host computing machine enables command and control of the instrument's mode of operation and performs the necessary post-processing for the analysis and visualization of the measurements data. The application of this approach in the design and development of a DVB-S2 receiver using a software-radio platform is presented.**

I. Introduction

Modern communications systems rely on new advances in various technical fields, such as adaptive modulation, iterative signal processing and error control coding [1] for providing new services. As a result, the hardware implementation of such systems involves complex and demanding techniques in terms of processing power and speed. Due to the embedded multi-layer and multi-domain functionality, i.e. from physical to network layer and from complex signal constellations to binary user data, system-on-chip solutions are realized in the form of a multi-processing and multi-tasking environment, where the various processing stages are implemented as concatenated and parallel software circuits that interact with multiple hardware accelerator modules. Therefore, system prototyping appears to be a quite challenging task, since the data flow from different hierarchical levels, e.g. physical-layer signaling, multi-space constellation mappings and N-tuple codewords, need to be monitored, associated and often visualized, which requires a more sophisticated testing and validation approach.

When discrete logic is used, verification is performed by using oscilloscopes and logic analyzers, while for FPGA implementations, hardware embedded probes are inserted by using additional discrete logic and memory, such as Chipscope [2]. In systems that employ multiple DSPs, a similar approach is needed in order to use software probes in the various software circuits. In this case, the various measurements and parameters collected at the various system processing stages, have to be associated with a common time-base in order to support signal validation and overall functional verification procedures. According to this approach, a flexible methodology is necessary for organizing such signal measurements and for enabling faster detection of out-of-specs conditions through automated and more effective verification mechanisms.

In this paper, we present a software-based instrument for testing and monitoring such multi-processing systems. The instrument is based on a versatile architecture that incorporates a set of interconnected software circuits acting also as system probes and is developed on a software-radio platform suitable for the design and implementation of complex communications systems. Additionally, a high-speed local bus or a high-speed network is used to transfer the collected data to a high-level software tool, for analysis and post-processing. In this work, we have used the high-level MATLAB/Simulink environment for providing the user interface and implementing the data collection and post-processing functionality.

The rest of this paper is organized as follows. Section II presents the details of the instrument's architecture and describes its functionality. Section III demonstrates the application of the presented methodology in the design and implementation of a DVB-S2 receiver using a software-radio platform, while Section IV provides a measurement and verification procedure example of the instrument in the above satellite receiver.

II. The Instrument's Architecture

The architecture of the presented instrument is shown in Fig. 1. It consists of various software probes attached to different processing stages of the device under development, the memory modules that provide the necessary storage of the collected data, the data acquisition and packet generation unit that coordinates the collection of the measurements from the various probes, the instrument's basic controller and finally the interface to the high-level analysis environment.

The software probes are integrated with the various software circuits (SCs) of the system under development. The SCs are based on multiple concurrent threads and may be implemented in different processors. Each SC has several inputs and outputs that are fed by and drive other subsystems. The software probes are implemented using dedicated threads that are embodied in the SCs of interest and are responsible for driving the measurement data to the memory modules using a

†N. Papandreou is currently with IBM Zurich Research Laboratory

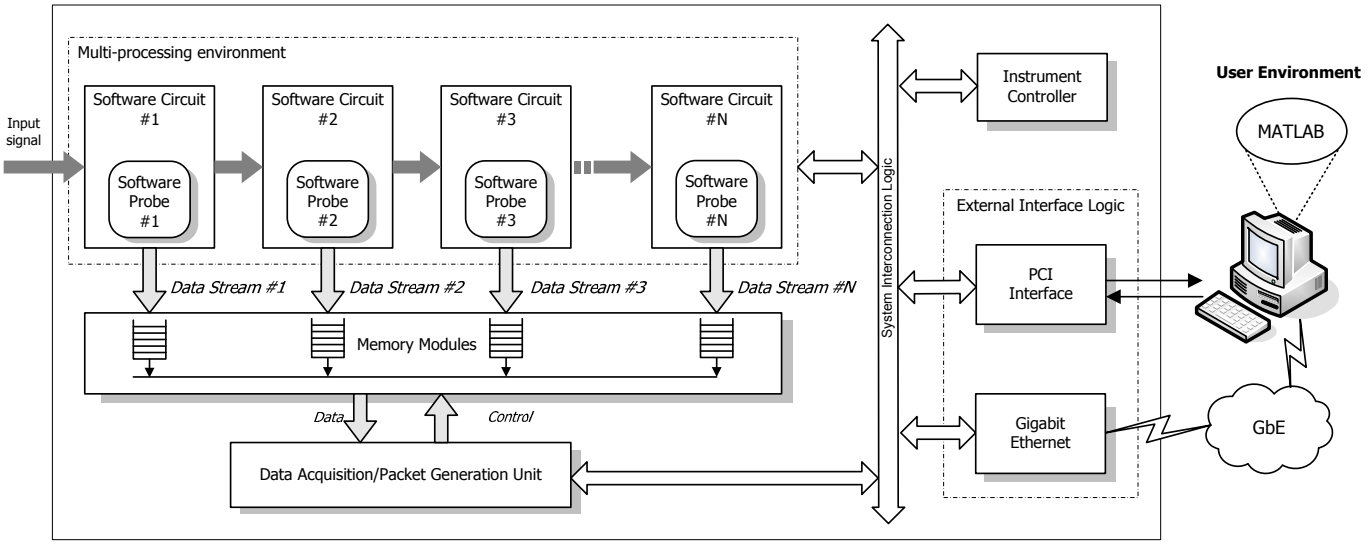


Fig. 1. The instrument's architecture.

flexible structure of software/hardware FIFOs. The scheduling of the various threads enables the transparent exchange of data between the SCs and the memory modules in the overall system data flow.

All data inserted into the FIFOs are further processed by the Data Acquisition and Packet Generation Unit (DAQ/PGU), the main task of which is to provide time-stamps, to coordinate the measurement processes and to transfer the data to the external interface. In particular, the DAQ/PGU associates each software probe with a specific data stream from the instrument to the external interface. For each data stream the DAQ/PGU converts the retrieved data to a common format and generates a packet which is associated with a header and an appropriate time stamp. When packets from multiple streams are available, they are multiplexed in order to generate the final measurements packet to be sent to the external interface unit.

The presented instrument provides two flexible interfaces to the user. A PCI interface for interconnection with a host PC and a Gigabit Ethernet (GbE) interface for networking environments. In the current version, a MATLAB-based high level application is used for controlling the instrument and for the visualization of the measurements and debugging information collected from the various SCs. In particular, the user application is responsible for demultiplexing the different data streams and proper correlation of the measurement data based on their timing information while it enables the user to send commands to the instrument where they are processed and acknowledged by the instrument's controller. Several types of commands are supported by this measurement environment, including:

- enable or disable each measurement data stream
- configuration of each stream sampling rate
- configuration of each stream data format
- definition of the measurement data carrying packet length

The high-level application performs the post-processing of the received measurements for proper presentation and visualization. Various statistics can also been extracted from the received data streams giving a more detailed perspective of the system's behavior.

III. The Instrument in a DVB-S2 Receiver Implementation

In this section, we present an application example of the presented instrument integrated in a DVB-S2 receiver prototype. The receiver consists of an FPGA circuit responsible for processing the analog IF input signal and for generating the respective baseband signals, and three DSP cores implementing the baseband signal processing from the I-Q signal components up to the processing of user IP packets and transmitting them over a gigabit ethernet (GbE) LAN. All the receiver circuits have been developed on a software radio platform, which also incorporates additional reprogrammable logic for logic demanding functions (line ECC), for the interconnection of the four hardware modules [3] and for acquiring the data at various stages of the receiver's processing flow. The integration of the instrument in the architecture of the DVB-S2 receiver prototype is shown in Fig. 2.

A. The DVB-S2 Receiver Architecture

The input IF signal feeds the receiver's high resolution analog to digital converter (ADC). The ADC digitizes the signal and drives the subsequent digital signal processing units. The first unit is the digital down-converter (DDC) [4] that is implemented in a FPGA and converts the IF signal into the respective baseband one. The DDC numerical control oscillator (NCO) frequency is controlled by the decisions of a coarse carrier offset control loop in the Carrier Frequency Recovery unit.

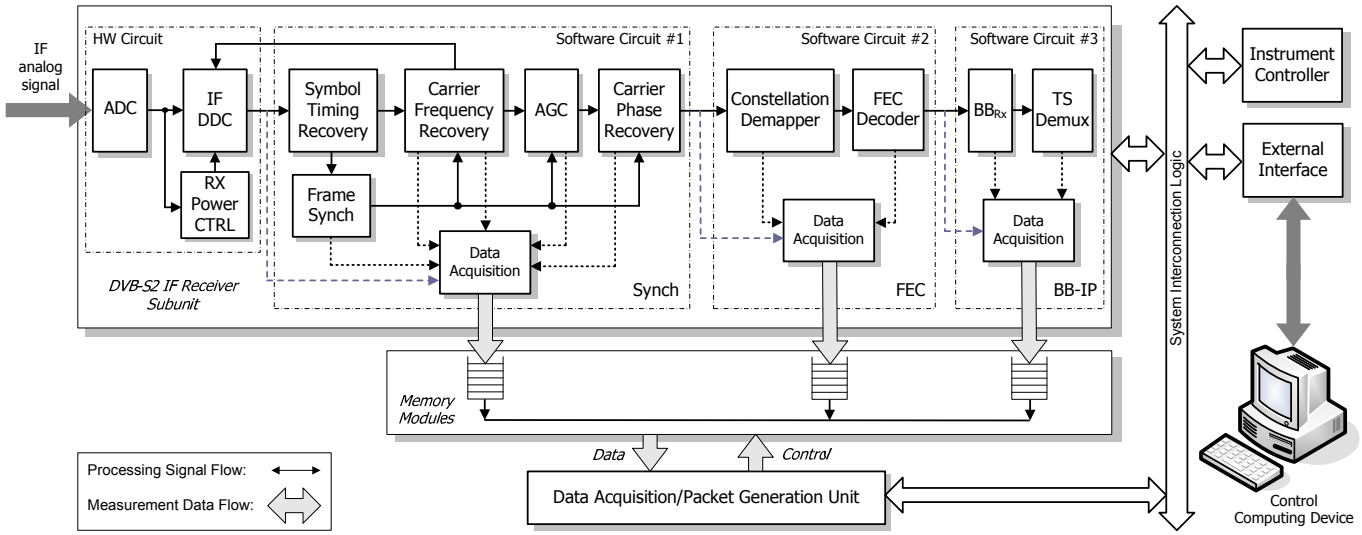


Fig. 2. Instrument's integration in a DVB-S2 IF receiver.

The baseband processing stages have been implemented as various software processing circuits (SCs) and allocated to three DSPs. In particular, the symbol rate error is compensated by using a mechanism that tracks the symbol rate fluctuations and generates the synchronized samples of the incoming symbols. Symbol timing recovery (STR) is implemented as a second order feedback loop utilizing a Farrow structured interpolator along with a non-data-aided (NDA) timing error detector (TED) [5]. As soon as STR converges, the boundaries of the DVB-S2 physical layer (PL) frames are detected by a frame synchronizer. This is done by searching the physical layer header based on the maximization of an appropriate output correlator [6]. Based on the correct framing alignment, it is possible to demultiplex (through proper descrambling) the pilot symbols from the incoming frame in order to drive the following pilot-aided carrier and phase synchronization units.

Carrier frequency recovery is based on pilot symbols that are regularly repeated in the transmitted PL frame. The recovery is performed in two sequential steps. The first step incorporates a '*coarse*' frequency recovery (CFR) 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 [7]. After the convergence of the CFR loop, where the frequency offset error is in the order of a few hundreds of kHz, a second stage of '*fine*' frequency recovery (FFR) is performed. FFR deploys a feed-forward estimation algorithm derived from an alteration of the L&R technique [8].

After carrier recovery, the resulting residual frequency offset error is compensated by two phase recovery mechanisms, i.e. '*coarse*' (Coarse Phase Recovery-CPR) and '*fine*' (FPR) respectively. The first is targeted for low order modulation transmissions (QPSK or 8-PSK) and is based on a pilot-assisted maximum-likelihood (ML) feed-forward estimator which computes the average phase of each pilot field [7]. The second mechanism operates additionally to the first one when the 16 or 32-APSK modulation scheme is used and consists of a closed loop based on the NDA phase error detector of Q-th power [9]. Between these two procedures, a digital automatic gain control (DAGC) takes place that is based on a data-aided vector tracker mechanism (DA-VT) [10], which utilizes the known pilot symbols for calculating the amplitude normalization multiplication factor.

Finally, the retrieved symbol stream is forwarded to the signal constellation demapper and the respective bit frames are further processed by the forward error correction (FEC) module that incorporates LDPC and BCH decoding with interleaving. The decoded frames are then processed by the respective payload processing circuits in order to extract the IP datagrams and to forward them into the GbE network.

B. The Software Radio Platform

The DVB-S2 receiver prototype along with the presented software-based instrument are implemented on a powerful hardware prototyping platform based on multiple DSP and FPGA devices. The hardware accelerator module shown in Fig. 2 digitizes and processes the incoming IF signal using a 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 temporary storage purposes. The first and second SCs are executed on two powerful C6416T DSPs running at 1GHz and two Virtex-II Pro (XC2VP30) FPGA with an SDRAM of 256MB. The third SC runs on a C6713 DSP, a Virtex-II FPGA, while a NetSilicon ARM-chip with an integrated MAC controller is used for network protocol processing.

C. Mapping of the Instrument Procedures into SCs

The first SC is dedicated to the multi-domain synchronization of the demodulation process which includes the symbol rate, frequency and phase recovery mechanisms along

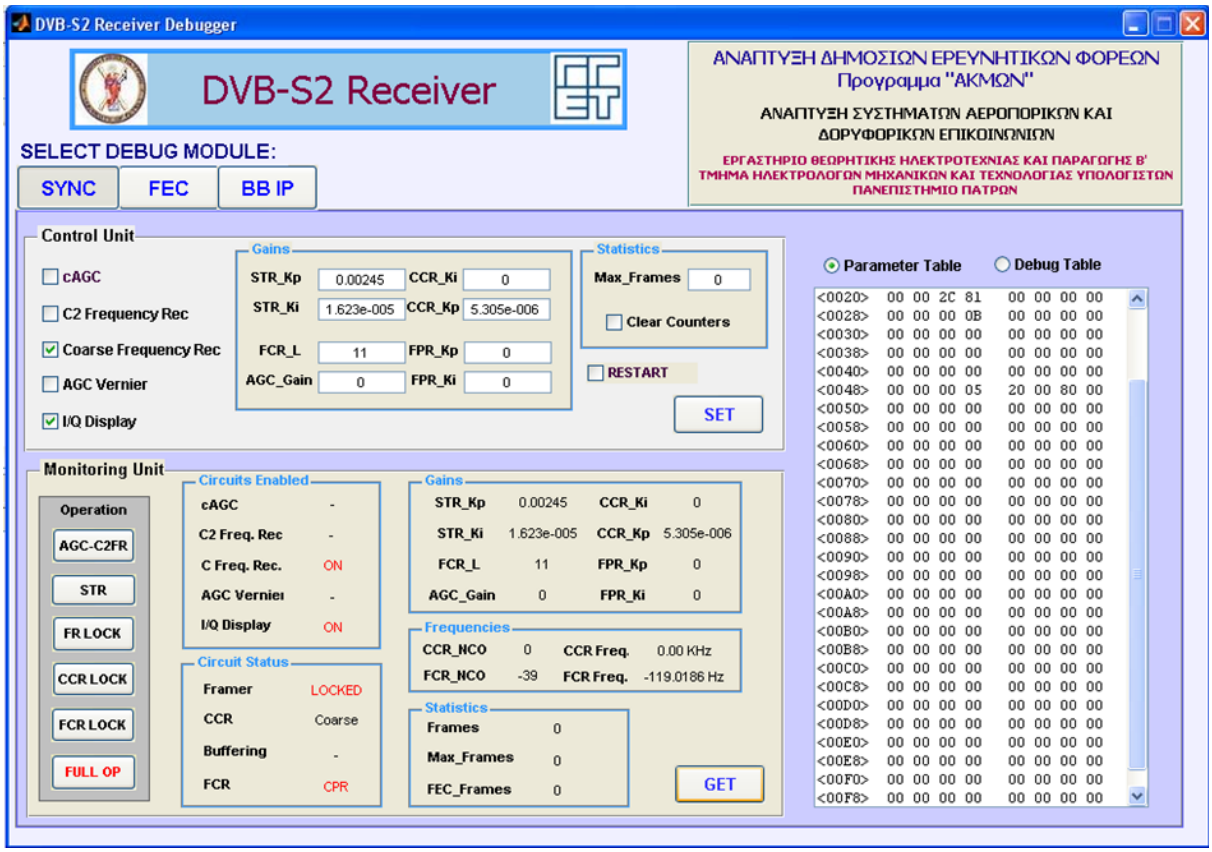


Fig. 3. The receiver's synchronization tab of the MATLAB-based graphical user interface.

with frame synchronization and amplitude gain control. These mechanisms are implemented as parallel DSP threads. The threads implementing symbol rate recovery, frame synchronization and coarse/fine carrier recovery operate on the DDC output sample stream generating data blocks of PL frames. The threads responsible for the phase recovery and amplitude gain control are executed periodically on the complete frames derived by the previous set of threads. The total processing load of the SC's threads is properly divided into small time slots by the SC task scheduler.

The second SC is responsible for demapping, interleaving and decoding (LDPC/BCH) the received frames of complex symbols which operations are performed on several threads deploying specific hardware accelerator modules. Finally, the last SC implements the interface of the receiver with the external environment that extracts the IP datagrams from successive frames in order to forward them to the GbE Ethernet network. The GbE network interface is also used for the connection of the high-level application with the instrument platform for control, post-processing and visualization of retrieved measurements. The procedure above exploits several DSP threads along with dedicated hardware logic. Furthermore, the last SC circuit is also responsible for receiving and distributing the control commands of the high-level application that perform either receiver configuration or management of the performed measurements. All software circuits incorporate a dedicated thread that is used to coordinate their operation.

As shown in Fig. 2, an additional concurrent software thread exists in each software circuit (Data Acquisition) that manipulates the various outputs of the signal processing units along with specific internal parameters. Its main task is to transfer the required data from the software circuit into dedicated FIFOs, implemented into the distributed memory resources of the platform. In addition, these threads are also responsible for converting the data of different parameters and signals into a common format and adding the necessary timing information. The parameters and signals monitored in each SC can be either enabled or disabled through a set of control commands initiated by the high-level application.

Finally, the data that have been selected for acquisition are stored into FIFOs which are read from the Data Acquisition and Packet Generation Unit (DA/PGU) in order to generate the necessary packets. DA/PGU logic exploits hardware resources on each SC while the transmission of the generated packets is totally performed by the last SC (BB-IP in Fig. 2).

At the computing device, a MATLAB based application is used to control the various receiver parameters, while it also collects information about the receiver's status, giving a full overview of the state of each software processing module. More specifically, the user is able to:

- activate/deactivate different circuits
- monitor their status
- change or acquire their configuration parameters
- collect statistics of the synchronization process (i.e. ratio of the undetected frame boundaries)
- measure performance metrics (i.e. EVM and BERT) and
- collect statistics at the network protocol processing stages

As an example, the GUI with the receiver synchronization specific tab is given in Fig. 3. Moreover, the high-level application is able to receive the measurement data packets and by exploiting the visualization and data processing capabilities of the MATLAB environment, to determine the dependence of various parameters and to demonstrate them. Such characteristics enforce the easy and effective comprehension of acquired measurement data and related post-processing results.

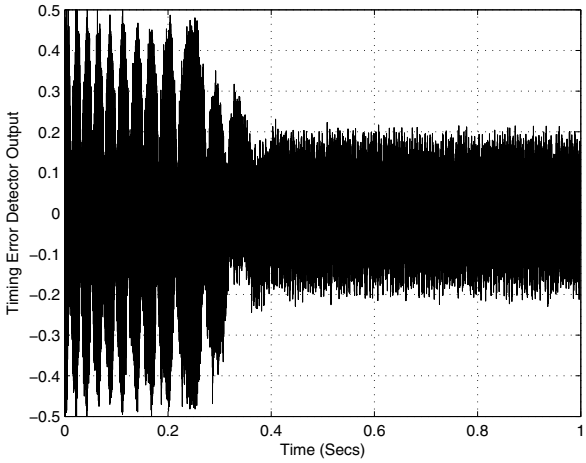
IV. DVB-S2 Receiver Measurements and Validation Example

In this section, we demonstrate an example of using the highlighted instrument, which is integrated into the DVB-S2 receiver software-radio implementation, that depicts the instrument's contribution to complex communications systems.

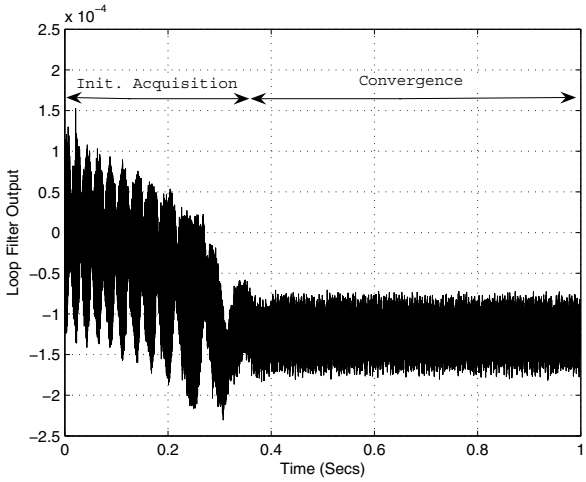
Symbol timing recovery (STR) comprises a significant mechanism that strongly affects the overall receiver performance and efficiency while it can be hardly comprehended in terms of its functionality and influence to the input sample stream, since it is usually realized as a closed loop. The presented instrument was utilized for proper evaluation, verification and also illustration of the STR closed loop operation. In the rest of this subsection, it is described the STR verification that was performed using the given instrumentation environment:

1) Initially, the Data Acquisition thread was enabled.
2) Then the internal parameters of STR that have to be monitored were determined, including its input and output. These parameters are the timing error detector output, the loop filter output and the control word of the interpolator which compensates the estimated symbol rate error.
3) Then the size of the generated packets was determined by a specific command (from the high-level application) and the receiver was activated.
4) Packets were transmitted to the control computing device through the GbE interface until an internal STR lock detector has identified that the STR has converged.
5) Finally, the high-level application creates the plots of the different parameters using a common time scale, while post-processing calculates several performance metrics such as: duration of initial acquisition, standard deviation of estimation error and mean normalized estimated error.

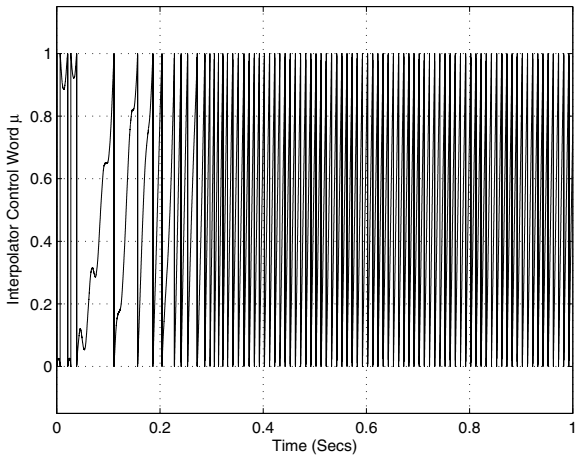
An example of the three STR internal parameters is given in Fig. 4.



(a)



(b)



(c)

Fig. 4. The three STR loop internal parameters monitored during STR operation ((a): Timing Error Detector Output, (b) Loop Filter Output, (c): Interpolator Control Word). The calculated performance metrics related with this measurement are: *Lock Time*: 335.37 msec, *Estimation Error Standard Deviation*: 2.2425e-5 and *Mean Norm. Estimated Error*: 2.493725e-4.

V. Conclusions

This paper presented the architecture of a software-based instrument that can be used for testing, debugging and verification of complex communications systems that are realized in a multiple DSP environment. Due to the complexity of such systems a flexible methodology for efficient and rapid prototyping is needed. The highlighted instrument fulfills such a demand since it provides the means for effective validation and testing as it incorporates a versatile software measurement acquisition environment along with a control computing device that performs post-processing and visualization using a high-level MATLAB application. An extensive application example of the software-based instrument into a software radio implementation of a DVB-S2 IF receiver has also been thoroughly analyzed.

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) in the framework of project 'AKMON'.

References

[1] A. Morello and V. Mignone, "DVB-S2: The second generation standard for satellite broad-band services," *Proc. IEEE*, vol. 94, no. 1, pp. 210–227, Jan. 2006.

[2] XILINX Inc., "Chipscope Pro, Software and Cores User Guide," ver. 10.1, March 2008.

[3] P. Savvopoulos, N. Papandreou and T. Antonakopoulos, "The Architecture of a Software Radio DVB-S2 Receiver," *In Proc. 13th Ka and Broadband Communications Conference*, Turin, Italy, Sep. 24-26, 2007.

[4] P. Savvopoulos and T. Antonakopoulos, "An IF digital down-converter for software radio DVB-S2 receivers," *In Proc. 14th IEEE International Conference on Electronics, Circuits and Systems - ICECS'07*, Marrakech, Morocco, Dec. 11-14, 2007.

[5] F. M. Gardner, "A BPSK/QPSK Timing-Error Detector for Sampled Receivers," *IEEE Trans. Commun.*, vol. COM-34, pp. 423–429, May 1986.

[6] Y. J. F.-W. Sun and L.-N. Lee, "Frame synchronization and pilot structure for second generation DVB via satellites," *International Journal of Satellite Communications and Networking 2004*, vol. 22(3), pp. 319–339, 2004.

[7] U. Mengali and A. N. D'Andrea, *Synchronization Techniques for Digital Receivers*. New York: Plenum, 1997.

[8] 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.

[9] E. Casini, R. DeGaudenzi, and A. Ginesi, "DVB-S2 modem algorithms design and performance over typical satellite channels," *International Journal of Satellite Communications and Networking 2004*, vol. 22(3), pp. 281–318, 2004.

[10] R. De Gaudenzi and M. Luise, "Analysis and Design of an All-Digital Demodulator for rellis Coded 16-QAM Transmission over a Non-Linear Satellite Channel," *IEEE Trans. Commun.*, vol. 43, no. 2/3/4, pp. 659–668, Feb/Mar/Apr 1995.