Soft-DVB: A Fully-Software GNURadio-based ETSI DVB-T Modulator

Vincenzo Pellegrini, Giacomo Bacci and Marco Luise
University of Pisa - Dip. Ingegneria dell'Informazione - Via Caruso - 56122 Pisa, Italy
Email: vincenzo.pellegrini@studenti.ing.unipi.it; giacomo.bacci@iet.unipi.it; marco.luise@iet.unipi.it

***Abstract*— Typical architectures for digital video broadcasting-terrestrial (DVB-T) modulators are based on task-specific hardware to adapt the baseband audio/video signal to the radio frequency terrestrial channel. This paper describes the architecture of a fully-software, real-time modulator for DVB-T signals named Soft-DVB, which makes use of a software section developed from scratch by the authors and integrated into the GNURadio framework. Implementation results are shown and possible application scenarios are outlined. The excellent performance achieved by Soft-DVB proves the feasibility of a fully-software architecture on a standard PC platform as a viable and profitable solution for next-generation DVB-T (DVB-T2) modulators.**

***Index Terms*— DVB-T, OFDM, software defined radio, broadcasting, baseband processing, GNURadio**

I. Introduction

DIGITAL VIDEO BROADCASTING (DVB), in its terrestrial version (DVB-T), is currently the most widely deployed system for delivering standard and high definition video content to digital TV users worldwide. Although born as an European initiative, with the standardization process led by the European Telecommunications Standards Institute (ETSI) [1], DVB-T is today already deployed or adopted in more than 70 countries [2]. Such broad set of nations include most countries from Europe, Africa, the Middle East, and Oceania, and some Asian countries, including India and Singapore. As a consequence, the worldwide application, along with the extension to handheld devices, makes ETSI DVB-T the mainstream broadcasting system for both free-to-air (FTA) transmissions and conditional access content.

The main reasons for the almost worldwide application of ETSI DVB-T lie in the high spectral efficiency and in the robust multipath resistance due to the orthogonal frequency division multiplexing (OFDM) technique. As of 2008, state-of-the art techniques for modulating the ETSI DVB-T signal heavily rely upon task-specific hardware (HW) to implement the coded OFDM (COFDM) modulation required by the standard [1]. This architecture, which applies both to broadcasting-oriented full HW modulators and to DVB-T testbeds equipped with peripheral component interconnect (PCI) boards, is widely adopted due to good performance and low cost.

However, an HW approach shows many drawbacks, that can be mitigated by resorting to a fully software (SW) solution. Just to mention a few, major benefits of an SW architecture include:

i) affordable camp equipment for emergency broadcast;
ii) easy upgrade to DVB-T standard evolution;
iii) support for high-complexity cell-based content-distribution networks;
iv) potential for using DVB-T as the transport layer technology for Internet protocol (IP)-based networks [3] in rural communities to address digital-divide impairments.

We also emphasize that migrating to SW technology at the transmitter side does not call for existing user-side equipment to be modified.

This paper describes a proof of concept, low-cost GNURadio-based modulator for COFDM DVB-T signals, named Soft-DVB. The baseband-to-radio frequency (RF) front-end is provided by the universal software radio peripheral (USRP), a device produced by the Ettus Research LLC [4]. The SW section is entirely developed by the authors, integrating structured C++ code into the GNURadio hybrid C++/Python framework [5].

The remainder of the paper is structured as follows. Sect. II describes the main features of the standard ETSI DVB-T modulator, whereas Sect. III provides an introduction to the relevant architectural choices of Soft-DVB. Sect. IV contains some details about the high-level solutions and the functional blocks, including the optimizations needed to reduce the computational load and to reach real-time operation. The implementation results are reported in Sect. V, whereas perspectives and conclusions are outlined in Sects. VI and VII, respectively.

II. The Standard ETSI DVB-T Modulator

The ETSI DVB-T system allows one transport stream (TS) to be transmitted over the air. (Optionally, two TSs can be combined through the use of hierarchical modulation techniques.) Each TS is designed to carry multiple audio/video and/or data channels. Distribution over single frequency networks (SFNs) is also supported.

In this section we will briefly analyze the structure of the transmitter device that adapts the baseband motion picture expert group-2 (MPEG2) signal to the typical multipath radio frequency channel, as described in [1].

Fig. 1 shows the functional block diagram of a DVB-T modulator. Each function block is shortly described as follows:

- the *multiplex adaptation for energy dispersal (MAED)* removes time domain correlation from byte-wise MPEG2

This research was supported in part by the Mediterranean Broadband Infrastructure (MBI).

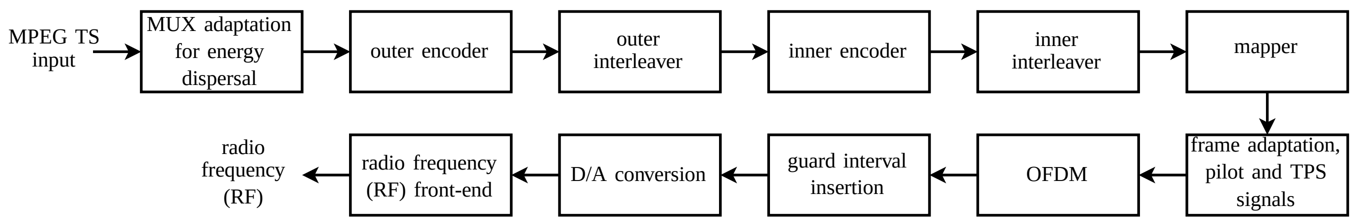


Fig. 1. Functional block scheme for the standard ETSI DVB-T transmitter [1].

input stream by performing a bit-level XOR with a pseudo-random binary sequence (PRBS). The proper PRBS is generated via a feedback shift register (FSR) by a generator polynomial;

- the *outer encoder* performs Reed-Solomon (RS) encoding at byte level and in systematic form. The chosen code is a RS(204,188), obtained by shortening a RS(255,239). The main purpose of the RS code is to mitigate the impact of the error bursts produced at the receiver side by the Viterbi algorithm;
- the *outer interleaver* aims to minimize correlation between the residual errors at decoding time;
- the *inner encoder* performs convolutional encoding and puncturing to achieve quasi-error-free (QEF) transmission.
- the *inner interleaver* provides support for hierarchical modulation and fairness in assigning the payload bits to the OFDM carriers, i.e., it prevents certain bits from the original TS from being constantly assigned to the same set of OFDM carriers with unsatisfactory signal-to-noise ratio (SNR);
- the *mapper* modulates the encoded bits onto QPSK, 16-QAM, and 64-QAM constellations. Both non-hierarchical and hierarchical modulations are supported, thus allowing two different TSs with different error performance to be transmitted simultaneously;
- *frame adaptation (FA)* provides insertion of all needed reference signals: pilot carriers, used for synchronization and channel estimation, and carriers conveying information upon the transmission parameters being implemented. This operation is called transmission parameter signaling (TPS);
- the *OFDM modulation* block adds virtual carriers and performs an inverse fast Fourier transform (IFFT) with 2048 (2k) or 8192 (8k) subcarriers.;
- *guard interval insertion* inserts the guard interval (cyclic prefix) required by the duration of the channel impulse response.
- the *digital-to-analog (D/A) conversion* interpolates the signal, thus producing an analog signal;
- the *RF front-end* shifts the basesband I/Q analog signal to the proper carrier frequency of the desired TV channel.

III. Soft-DVB Architecture

As stated in the introduction, Soft-DVB represents a proof-of-concept DVB-T modulator designed following the software-defined radio (SDR) paradigm. In an SDR architecture, most (if not all) signal processing is performed digitally via SW routines. By inspecting Fig. 1, it is possible to identify the last two blocks as the only strictly HW components required. Hence, in the spirit of designing a fully-SW DVB-T modulator, the architecture selected for Soft-DVB only consists of the front-end as the HW section, whereas all the other functional blocks are implemented through SW modules.

Amidst all the possible solutions to realize Soft-DVB, the GNURadio paradigm [5] has been selected for the SW section, whereas the USRP produced by Ettus Research LLC [4] has been chosen for the HW section. The remainder of this section reports the main motivations and implications of both choices.

A. The SW section: The GNURadio paradigm

The main motivations that led us to choose the GNURadio framework lie in the efficiency and high flexibility provided by this solution. GNURadio is an open source project aimed at developing SDR terminals [5]. In particular, this project provides a collection of software modules to perform the main operations for signal processing. It is interesting to note that the GNURadio paradigm currently gathers a very active worldwide community of developers building their projects around this platform.

The GNURadio framework is split in two wide areas. The first one makes use of the C++ programming language, while the latter one exploits the flexibility provided by the Python scripting language. Although C++ and Python are both objected-oriented languages, the first one is compiled, whereas the latter one is interpreted. Due to the different performance, C++ is devoted to all the operations concerning signal processing within a block, whereas Python is used to instantiate and connect different blocks together. The use of Python is particularly attractive, since it makes the whole project flexible and easily reconfigurable.

The entire Soft-DVB baseband signal processing is performed via SW within the host computer. Consistently with the GNURadio general architecture, we make use of the Python scripting language to build up the GNURadio "flowgraph" of Soft-DVB, i.e., the functional block diagram in Fig. 1. As we will show in Sect. IV, the need for high efficiency in a real-time context is met by writing each signal processing block in C++. This approach in fact provides high performance, and also allows us to have the full control over the computational efficiency of the modulator.

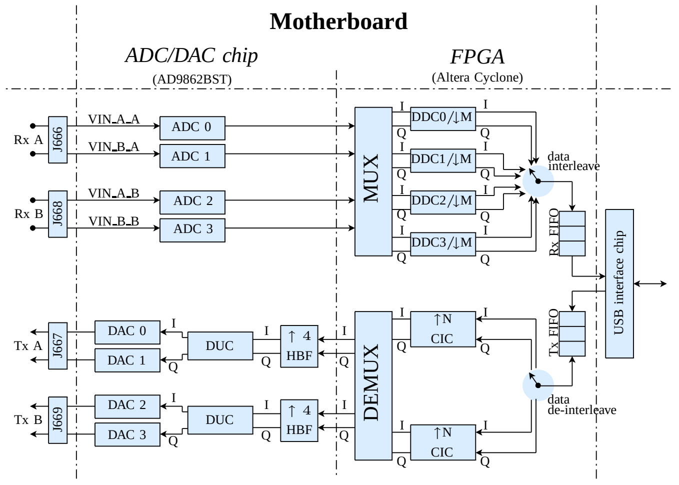


Fig. 2. Functional block scheme of the USRP motherboard [5].

B. The HW section: The USRP device

The USRP is a device designed within the GNURadio project [5] and produced by Ettus Research [4]. As a consequence, it perfectly fits the GNURadio framework considered for the SW section. Furthermore, the USRP has been selected for Soft-DVB due to the good efficiency of the device and because of its high quality/pricing ratio.

In the following, a short description of the main operations performed by the USRP is given. The output of the transmission chain depicted in Fig. 1 is the stream of I/Q samples computed by the host computer. The interface between the SW and the HW section is represented by the universal serial bus (USB), which conveys the output samples to the HW section, represented by the USRP.

As already stated in the introduction to this section, the USRP implements the last two blocks in the functional scheme of Fig. 1, namely D/A conversion and RF front-end. It is composed of a *motherboard* (for the D/A conversion) and a *daughterboard* (for the RF upconversion). The functional scheme for the motherboard is sketched in Fig. 2. Once the samples are delivered to the URSP, they are interpolated by means of the field programmable gate array (FPGA) section, then they are upconverted by the digital up converters (DUCs), and finally they are sent to the two digital-to-analog converters (DACs) to produce a signal in the analog domain.

The analog signal is then upconverted and amplified by the daughterboard. During our test activity, and still at the time of writing this paper, no front-end has been made available from Ettus Reserach LLC that can cover the DVB-T VHF/UHF spectrum without requiring modifications to the HW. A daughterboard covering such needs is currently under development. Therefore, given the contingent lack of proper HW solutions, we use the USRP in combination with the "BasicTX" daughterboard, which is not meant to operate above $250\,\mathrm{MHz}$ [6]. Even though this solution is heavily suboptimal, we use the fourth spectral replica of the DAC output, applying a preliminary shift of $f_{DUC} = 42$ MHz via the DUC (Fig. 3). The DACs operate at $f_{DAC} = 128$ MHz,

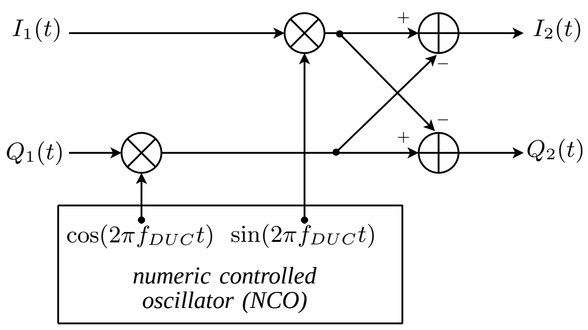

$I_1(t)$
$Q_1(t)$
$I_2(t)$
$Q_2(t)$
+
−
+
−
$\cos(2\pi f_{DUC}t)$ $\sin(2\pi f_{DUC}t)$
numeric controlled oscillator (NCO)

Fig. 3. DUC functional block scheme [5].

thus yielding an RF central frequency

$$f_{RF} = 4f_{DAC} + f_{DUC} = 554\ \mathrm{MHz}, \tag{1}$$

which corresponds to channel 31 UHF. As a consequence, we will use this frequency for our DVB-T test transmissions (see Sect. V for further details).

Another limitation in using the USRP is given by the speed of the interface with the host computer. In fact, its performance is currently limited by the USB 2.0 bus transfer rate. For commercial chipsets, the maximum reliable transfer speed over this bus is[1] $R_{max} = 32\,$Mbyte/s. When sampling an 8-MHz channel using a sampling frequency $f_s = 8\,$Msample/s, the most accurate sample representation is

$$N_{bit} = \frac{R_{max}}{2f_s} = 2\,\text{byte/sample} = 16\,\text{bit/sample}. \tag{2}$$

In order to represent a 16-MHz bandwidth, the sample resolution can be reduced from 16 to 8 bits. However, this yields a loss in quantization SNR equal to $48\,\mathrm{dB}$, which is definitely not acceptable for any RF transmission. Thus, according to [1], an 8-MHz band forces us to implement a 7-MHz DVB-T channel. Nonetheless, it is interesting to note that the next version of USRP, which is currently under development, will remove the USB bottleneck by using gigabit ethernet. This feature will allow Soft-DVB to broadcast a full 8-MHz channel just by slightly upscaling the sampling frequency, without any change to the code.

IV. Soft-DVB Implementation

A. High-level description

To achieve real-time operation of the transmitter, all the signal processing blocks have been developed from scratch by the authors to have full control over the computational efficiency of the modulator. The only exception is represented by the OFDM block, in which the fast Fourier transform (FFT) is performed by making use of the algorithm developed in [7]. As shown in Table I, each digital signal processing (DSP) block reported in Fig. 1 is directly mapped into one of the Soft-DVB C++ DSP blocks.

DSP blocks are assembled and configured via a Python script named Soft-DVB.py, which also controls the USRP

[1]Many bargain chipsets actually provide a maximum throughput far lower than this limit.

TABLE I
MAPPING OF DVB-T BLOCKS INTO SOFT-DVB C++ BLOCKS.

DVB-T block	*Soft-DVB C++ block*
MAED	dvb.maed_bb
outer encoder	dvb.ocoder_bb
outer interleaver	dvb.ointerleaver_bb
inner encoder	dvb.icoder_bb
inner interleaver	dvb.iinterleaver_bb
mapper	dvb.mapper_bc
frame adaptation	dvb.insert_refsig_cc
OFDM	gr.fft_vcc, dvb.insert_vica_cc
guard interval insertion	dvb.insert_gi_cc

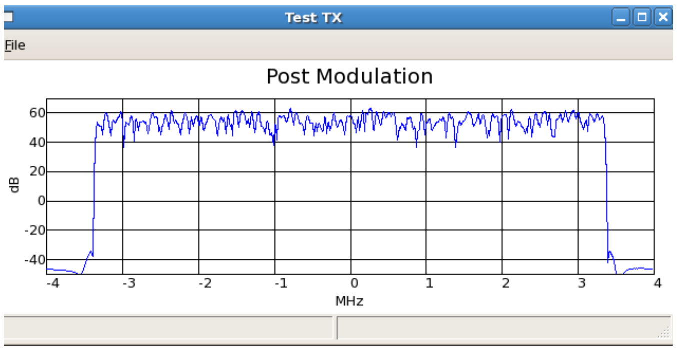


Fig. 4. Spectrum of the digital signal at the output of the host computer.

interpolation rate and the RF channel center frequency. The Soft-DVB version referred to in this paper implements either a 7 or an $8\text{-}\,\mathrm{MHz}$ DVB-T channel with non-hierarchical 16-QAM mapping, convolutional coding rate $2/3$ and OFDM guard interval $1/4$. As stated in [1], this yields a useful bitrate for our transmission of $11.612\,\mathrm{Mb/s}$. All other transmission parameters combinations are easily achievable with minor changes to the code. As an example of the flexibility provided by GNURadio hybrid Python/C++ framework, a different guard interval can be set just by sending a single parameter to the top-level Pyhton script. This script will automatically assemble a new flow-graph containing the block suitable for the chosen guard interval, without further modifying the transmission chain.

The implementation results described in this paper are obtained on a single core PC and without using any parallel computing strategy. In the future releases of Soft-DVB, we will take advantage of the cell processor-aimed parallel computation tools planned for next GNURadio releases (for further details, see Sect. VI).

B. *Reaching real time performance*

The first working draft of Soft-DVB was obtained after about two thirds of the development process, without taking computational performance into account. This draft version could correctly generate a standard DVB-T signal in about 6.8 times the amount of time required by real-time transmission.

The critical resource to achieve real time performance is the computational power of the host computer, whereas Soft-DVB is definitely *not* a memory-demanding application. Therefore, the most effective optimizations are those that trade off central processor unit (CPU) time for memory occupancy. A typical example is the Galois finite field (GF) multiplier involved in RS encoding, where the computation is really heavy but the set of possible result is limited — just 256 possible values in a $GF(2^8)$. Using a precalculated array impacts very profitably in terms of performance gain. The same principle is applied to functions such as PRBS generation, TPS data and redundancy calculation, constellation mapping, multiple permutations in the interleaver, and also when slicing bytes into the single bits for blocks working at bit level. Furthermore, converting bit-level operations into byte-level actions proves to have a dramatic impact on computational performance.

Further benefits are achieved by improving the way DSP blocks exchange data. For instance, preferring vectorized data structures to scalar structures reduces the number of input/output connections of single blocks. The vectorized data structures are assembled and disassembled within each blocks, without relying on external tasks, although they are present (and generally useful) in GNURadio. Additional reduction in the computational load is achieved by reducing the number of processing blocks to to be exactly those described in the ETSI DVB-T standard (as depicted in Fig. 1), therefore maintaining the flexible structure described in Sect. IV-A.

V. EXPERIMENTAL RESULTS

Real-time functioning of Soft-DVB has been validated at the Laboratory of DSP for Communications of the University of Pisa. As described in Sect. III, the signal is calculated by the host computer. The spectrum of the output digital signal is presented in Fig. 4. As can easily be verified, its shape is consistent with typical spectra of OFDM transmission schemes.

At the end of the SW section, the signal is sent via USB2.0 to the USRP and carried by the BasicTX front-end to the receiver via a $50\text{-}\Omega$ coaxial cable. Experimental tests have been performed using two different receivers: i) an off-the-shelf digital TV set-top-box, namely the Access Media® STBL3012; and ii) a typical USB-pen DVB-T receiver, namely the Pinnacle® PCTV USB Stick 70e. We have used a TS obtained by recording a transmission by RAI (the Italian public broadcaster). The output of the Access Media STB L3012 is shown in Figure 5. Both receivers can easily and quickly acquire the MPEG TS synchronization byte and report a BER below 10^{-9}.

GNURadio "realtime scheduling" prevents Soft-DVB from being interfered by other processes. Hence, Soft-DVB can regularly feed the USRP with a continuous sample stream, ensuring real-time operation in the presence of concurrent applications. Typical performance has been evaluated through several transmission cycles with minimum duration of 30 minutes. In its current version, Soft-DVB runs flawlessly real-time on a $3.0\text{-}\,\mathrm{GHz}$ Intel® Pentium IV processor, draining 83% of its computational power. Fig. 6 reports the CPU percentage used by Soft-DVB, as measured by GNOME on a Fedora6-equipped PC.

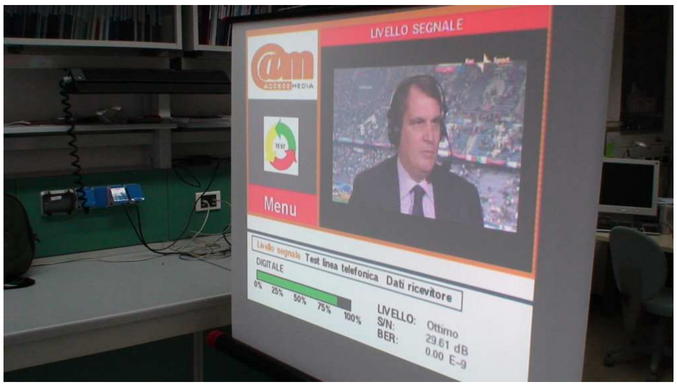


Fig. 5. Output of the Access Media STB1230 test receiver fed by the Soft-DVB transmitted signal.

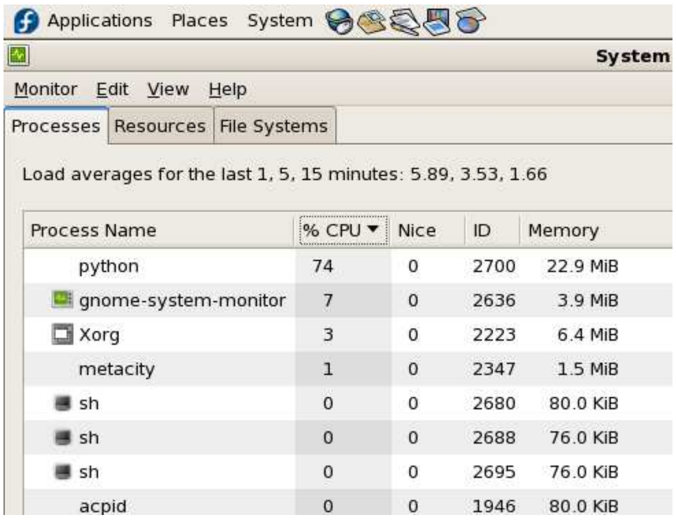
Fig. 6 shows another interesting performance metric, namely the memory usage, which confirms the statements provided in Sect. IV-B. Although Soft-DVB is a CPU-hungry application, due to the presence of several computation-intensive blocks, the memory demand is moderate. As can be seen in our tests, the whole transmission chain (from the input TS to the output signal) requires a memory amount lower than 24 MiB.

Further encouraging results have been obtained testing Soft-DVB on an entry-level general purpose laptop, namely an HP® Compaq nx6310. The current version of Soft-DVB runs real-time on this HW, equipped with a 1.46- GHz Intel® Celeron M CPU. As a conclusion, DVB-T modulation is likely to be feasible even on very low-end, extremely cheap machines through marginal optimization effort. Similarly, running many Soft-DVB instances on a heavily multicore CPU is expected to be highly profitable. Further details on this solution are discussed in Section VI.

VI. Perspectives

In the last few months, developers within the GNURadio project have been working on a new parallel architecture of this framework, providing software radio programmers with the tools needed to access the huge raw computational power of the Cell BE processor [8]. This architecture is expected to be part of GNURadio releases planned for the near future. On the Cell BE platform, the accessible computing power will be approximately $3 \cdot 10^{10}$ floating point operations per second (flops), which is approximately 30 times the computing power of the machine used to test the current version of Soft-DVB. Thus, assuming the number of flops as a proper metric for the system computational capability, some hypotheses about the potentiality of a Cell BE-based Soft-DVB system can be drawn.

Considering the current Soft-DVB computational needs to be fully parallelized and distributed among the various processing cores of the Cell BE, a bunch of USRPs is supposed to allow about 30 11.612-Mb/s DVB-T multiplexes to be broadcast. This solution yields a total throughput of 348.36 Mb/s, which is suitable for broadcasting about 70 standard definition MPEG2 channels. Higher throughputs are achievable by upscaling the sampling frequency to that required for a 8- MHz channel and using different constellations, coding rates, guard intervals. In terms of commercial applications, a large-sized HW transmission facility can be replaced by a high computing power platform (such as the Cell BE) running multiple Soft-DVB instances at full DVB-T channel capacity.



Load averages for the last 1, 5, 15 minutes: 5.89, 3.53, 1.66

Process Name	% CPU ▾	Nice	ID	Memory
python	74	0	2700	22.9 MiB
gnome-system-monitor	7	0	2636	3.9 MiB
Xorg	3	0	2223	6.4 MiB
metacity	1	0	2347	1.5 MiB
sh	0	0	2680	80.0 KiB
sh	0	0	2688	76.0 KiB
sh	0	0	2695	76.0 KiB
acpid	0	0	1946	80.0 KiB

Fig. 6. System status measured on a Fedora6 PC while one Soft-DVB instance is running. The “python” entry displays the resources consumed by Soft-DVB.

Furthermore, the strong cost reduction yielded by Soft-DVB fully SW architecture makes it possible to increase the number of DVB-T modulators that a provider could deploy. This possibility may suggest to switch to a cellular-based video distribution network, in which a large number of general purpose PCs implement many low power DVB-T transmitters, each delivering different video streams within different cells. This would make a true video on demand (VOD) offer possible over the existing DVB-T infrastructure.

Another appealing feature of possible evolutions of Soft-DVB, equipped with future releases of the USRP (e.g., the upcoming USRP2) and a multithreaded version of the SW section, is represented by transmitting two adjacent VHF/UHF channels with only one host PC and one front-end. This solution, leading to a 16- MHz bandwidth, allows for simultaneous transmission of two different DVB-T services, e.g., DVB-T and its upcoming evolution, DVB-T2. This architecture is thus suitable to provide next-generation broadcasting while not causing notable impairments to existing DVB-T users.

A further application of Soft-DVB is represented by the possibility of running single instances of Soft-DVB on low-end machines. This solution allows for broadcasting two standard definition TV channels in a “camp transmission” scenario, suitable for emergency civil protection purposes, and to provide services to limited land areas during abroad missions of armies or non-governmental organizations (NGOs).

VII. Conclusions

This paper described the architecture of a fully-software, low-cost real-time DVB-T modulator, named Soft-DVB, developed using the GNURadio hybrid C++/Python framework.

The promising results achieved by Soft-DVB point out that a full software solution for ESTI DVB-T signal generation is not just feasible, but also shows high flexibility, good performance and ease of implementation.

Soft-DVB represents a competitive solution to provide both small and large transmissive capability, with ease of upgrade to new forthcoming extensions of the DVB-T such as DVB-T2. The software approach, along with the cost reduction itself, allows for completely novel architectures, which directly result in a wide range of possibilities opening up for multimedia content distribution networks.

References

[1] *Digital Video Broadcasting (DVB); Framing Structure, channel coding and modulation for digital terrestrial television*, ETSI Std. EN 300 744 V1.5.1, Nov. 2004.

[2] DVB Project, “Official DVB-T deployment data,” 2007. [Online]. Available: http://www.dvb.org/dvb-deployment-data.xls

[3] *Digital Video Broadcasting (DVB); DVB specification for data broadcasting*, ETSI Std. EN 301 192 V1.4.1, June 2004.

[4] Ettus Research LLC website. [Online]. Available: http://www.ettus.com

[5] GNURadio website. [Online]. Available: http://gnuradio.org/trac

[6] M. Ettus, “TX and RX daughterboards for the USRP radio system,” 2006. [Online]. Available: http://www.ettus.com/downloads/miscdboards-v3b.pdf

[7] M. Frigo and S. G. Johnson, “FFTW: An adaptive software architecture for the FFT,” in *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, Seattle, WA, May 1998, pp. 1381–1384.

[8] S. Williams, J. Shalf, L. Oliker, P. Husbands, S. Kamil, and K. Yelick, “The potential of the cell processor for scientific computing,” in *Proc. ACM Conf. on Computing Frontiers*, Ischia, Italy, May 2006, pp. 9–20.

Vincenzo Pellegrini received the B.E. degree in telecommunications engineering from the University of Pisa, Pisa, Italy, in 2006.

He is currently attending his M.E. degree studies at the same university. Since the beginning of his B.E. degree thesis, he is actively working with the “DSP for Communications Lab” (DSPCOLA) of the Dept. of Information Engineering, University of Pisa, under the supervision of Prof. Marco Luise.

Giacomo Bacci received the B.E. and the M.E. degrees in telecommunications engineering from the University of Pisa, Pisa, Italy, in 2002 and 2004, respectively. He is currently working toward the Ph.D. degree in information engineering at the same university.

Since 2005, he has been with the Department of Information Engineering at the University of Pisa. In 2006, he was a visiting student research collaborator at the Department of Electrical Engineering at Princeton University, Princeton, NJ. His research interests are in the areas of digital communications, signal processing and estimation theory. His current research topics focus on power control for multiple-access wireless networks and time delay estimation for satellite positioning systems and wireless communications.

Marco Luise is a Full Professor of Telecommunications at the University of Pisa, Italy. After receiving his M.E. and Ph.D. degrees in electronic engineering from the University of Pisa, he was a Research Fellow of the European Space Agency (ESA) at ESTEC Noordwijk, The Netherlands, and a Researcher of the Italian National Research Council (CNR), at the CSMDR Pisa. Prof. Luise co-chaired four editions of the Tyrrhenian International Workshop on Digital Communications, was the General Chairman of the URSI Symposium ISSSE’98, and the General Chairman of EUSIPCO 2006 in Florence.

Prof. Luise is a Senior Member of the IEEE and served as an Editor for Synchronization of the IEEE TRANSACTIONS ON COMMUNICATIONS, and as an Editor for Communications Theory of the European Transactions on Telecommunications. He is the co-Editor-in-Chief of the recently founded International Journal of Navigation and Observation, and acts as the General Secretary of the Italian Association GTTI, Gruppo Telecomunicazioni Teoria dell’Informazione. He has authored more than 150 publications on international journals and contributions to major international conferences, and holds a few international patents. His main research interests lie in the area of wireless communications, with particular emphasis on CDMA/multicarrier signals and satellite communications and positioning.