

# Multimedia Terminals: Reception of Real-time Video Streams over DAB/DMB

C. Sanz, P. J. Lobo, F. Pescador, M. J. Garrido, M. C. Rodríguez, A. M. Groba and E. Juárez

**Abstract**—In this paper, a demo platform for reception of real-time video streams over Digital Audio Broadcasting /Digital Multimedia Broadcasting (DAB/DMB) is presented. This platform has been developed as part of the ARTEMI project to validate our research on DAB/DMB receivers and video decoders. The platform consists of a DAB transmitter emulator, a commercial DAB receiver, an Receiver Data Interface (RDI) extractor implemented on FPGA/RISC technology and an MPEG-4 video decoder implemented on DSP technology. We are working currently on adding DMB support to the receiver. With these elements, video streams over DAB/DMB can be received and played on a television set.

**Index Terms**—ARTEMI, DAB, RDI, DMB, Enhanced Packet Mode, IP, MPEG-4.

## I. INTRODUCTION

In the last years, the development of the Information Society Technologies (IST) has been one of the factors with greater impact from the economical and social point of view in the most advanced countries. This development has been characterized, first, by a greater implantation of all kinds of telecommunication networks that currently coexist (cable, satellite and terrestrial); and, second, by the appearance of new services and applications which are based on these networks (digital television, digital radio, Internet access, videoconference, security applications...).

The mean term advance in the development of IST requires to establish a convergence between the applications and services that the different communication networks support. One key element for this convergence is the terminal, which is the medium that people use to access IST. In this context, research on multimedia terminals is being performed in ARTEMI project [1].

In ARTEMI, methodologies are being proposed for the design of complex systems for multimedia applications. Also, a demonstration platform of a multimedia terminal that is able to receive DAB/DMB [2-4] encapsulated video streams is being developed.

Manuscript received May 12, 2006. This work has been supported by the Spanish Ministry of Science and Technology under grant TIC2003-09687-C02-01, and by the Comunidad de Madrid regional government under grant S-0505/TIC/0398.

All authors are with the Grupo de Diseño Electrónico y Microelectrónico (GDEM) of the Universidad Politécnica de Madrid (UPM) (e-mails: {cesar, pjlobo, pescador, matias, mcesar, amgroba, ejuarez}@sec.upm.es).

In this paper, the aforementioned DAB/DMB multimedia terminal platform is described. Section II contains an outline of ARTEMI project for reference. In sections III and IV, the platform DAB/DMB receiver block and the platform video decoder block are explained. In section V, the platform related short term future work is outlined. Finally, Section VI is devoted to the conclusions.

## II. THE ARTEMI PROJECT

### A. Introduction to the ARTEMI project

ARTEMI starts in 2003 as a research project carried out by the Universidad Politécnica de Madrid (UPM) and the Universidad de Las Palmas de Gran Canaria (ULPGC). The project has four main objectives:

- Research on SoC design methodologies having multimedia terminals as a reference application.
- Research on multimedia terminals to receive video streams over DAB/DMB.
- Research on video decoders using FPGA and/or DSP technologies, including comparison of both.
- Super-resolution processing techniques to enhance the image resolution in low bit-rate applications.

As can be seen in Fig. 1, the ARTEMI reference application has eight main functional blocks: in the transmission path, a full-resolution frame sequence is first decimated; the obtained low-resolution sequence is compressed with MPEG-4 and encapsulated over DAB. In the reception path, DAB de-encapsulation and video decoding are performed. A full-resolution sequence is obtained using super-resolution techniques, that is finally displayed.

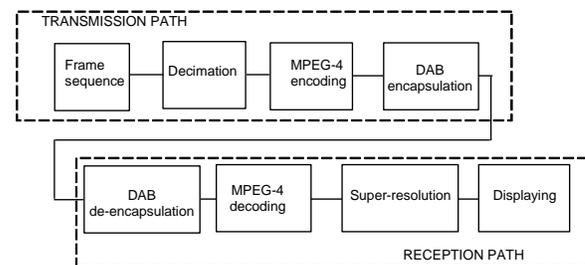


Fig. 1. Functional blocks in the ARTEMI reference application.

### B. Demonstration platform description

The demonstration platform developed in ARTEMI is

mainly focused in the reception path. A block diagram can be seen in Fig. 2. A commercial DAB receiver [5] with an RDI [6] output interface (RDI-O in Fig. 2) is used to tune a DAB ensemble. Alternatively, test RDI frames may be generated and used (RDI-E in Fig. 2). The RDI module allows to select the RDI source and performs the signal conversion, from optical to electrical, if needed. FPGA-RISC and DSP are itself platforms containing resources based on FPGA and DSP technologies. These platforms are used to implement the reception path main functionalities and may communicate with each other through an Ethernet link.

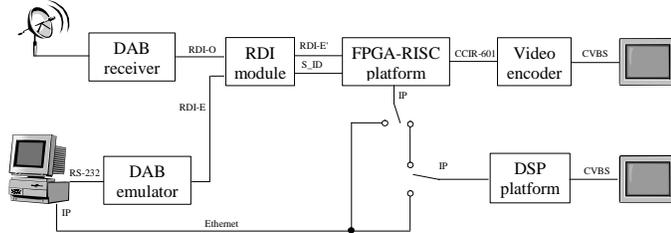


Fig. 2. Block diagram of the prototype of the ARTEMI reference application

### C. The UPM group research in ARTEMI

The research activities of the UPM group inside ARTEMI are focused in two main subjects:

- Research on multimedia terminals to receive video streams over DAB/DMB.
- Research on methodologies for optimization of video decoders using DSP as target technology.

In the next sections, the results up to date in this two working areas are explained and also the planned future work is outlined.

## III. DAB/DMB RECEIVER

In this working area, an extractor for video transmitted over DAB networks is being implemented using the FPGA-RISC platform. A prototype for extraction of IP encapsulated video has been developed.

### A. The DAB ensemble

The DAB ensemble is composed of three channels: the synchronization channel, the Fast Information Channel (FIC) and the Main Service Channel (MSC). Basically, the MSC carries the ensemble data, which is divided into subchannels, and the FIC contains information about the services that comprise the ensemble and which MSC subchannel(s) correspond to each service. Each subchannel can carry information either in stream mode, used for applications that stream data at a constant rate (e.g. audio programmes), or in packet mode, used typically by data services that do not have a constant rate. In packet mode, one subchannel can carry several data flows, each one identified by a packet address.

### B. Video transmission over DAB

The main problem for video transmission over DAB is that

its error protection scheme and network design guidelines are designed for reliable reception of audio data, but the typical Bit Error Ratio (BER) is far from being low enough for video transmission [7]. There are two approaches to overcome this limitation: DMB [3-4] and IP encapsulation using the Enhanced Packet Mode, which is to be standardized in the next revision of [2]. Both approaches provide additional error correction mechanisms.

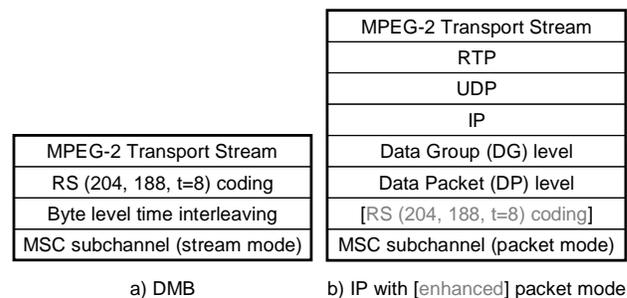
#### 1) DMB characteristics.

DMB (Fig. 3-a) specifies the transmission of H.264 encoded video into an MPEG-2 Transport Stream (MP2TS), which is itself transmitted in an MSC subchannel in stream mode. The MP2TS packets are protected with a Reed Solomon RS (204, 188, t=8) shortened code and byte-level time interleaving [3].

#### 2) Video over IP using the Enhanced Packet Mode.

This approach (Fig. 3-b) uses the standard methods for streaming video over IP networks using RTP [8-9]. IP datagrams are encapsulated into the DAB ensemble as packet mode NPAD (Non Programme Associated Data) [10]. Each datagram is contained in a Data Group, which is then sliced in fixed length Data Packets, which in turn are carried in a MSC subchannel. One subchannel can contain several IP datagram flows, each one identified by its Data Packet address.

The Enhanced Packet Mode adds further error protection (shown in grey in Fig. 3-b) to the standard DAB packet mode by applying the same RS (204, 188, t=8) code used in DMB to the Data Packets. The way in that they are arranged in the coding process adds also a time interleaving effect.



a) DMB

b) IP with [enhanced] packet mode

Fig. 3 Video transmission mechanisms in DAB

### C. The DAB/DMB receiver

Fig. 4 shows a functional block diagram of the DAB/DMB receiver, which is implemented using a general purpose RISC processor and a specialized hardware coprocessor. It can forward the received data either to an USB link, to an Ethernet link, or both. The blocks shown in grey (the RS decoders and the Ethernet port) are not yet implemented, so it works currently as an IP datagram extractor.

The receiver works as follows: initially, the microprocessor gets the identifier of the DAB service which carries the IP flow of interest. The hardware coprocessor is initially configured to forward the FIC data to the microprocessor, which parses it and gets the parameters of the MSC (subchannel identifier and packet address) that are necessary

to get the actual data. The microprocessor, then, configures the hardware filter with those parameters. The hardware coprocessor extracts the Data Groups from the MSC and also performs a SLIP encapsulation, in order to demand the less possible processing power to the microprocessor. The microprocessor's only function in this second (and definitive) phase is to forward the encapsulated IP datagrams to the personal computer via the serial link.

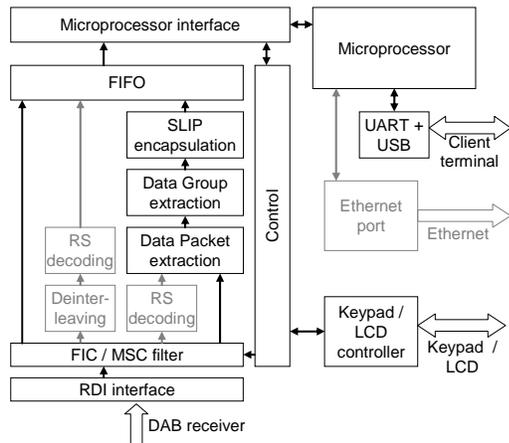


Fig. 4 Functional block diagram of the DAB/DMB receiver

#### D. The FPGA-RISC platform

The FPGA-RISC platform mentioned in II.B is shown in Fig. 5, and a block diagram is shown in Fig. 6. The platform is based on the EPXA10 Development Board [11] from Altera, which is itself based on an Excalibur EPXA10 FPGA [12] from the same manufacturer. The FPGA includes an embedded ARM-based processor. An additional daughter board has been developed to support the interfaces that were absent in the development board: the RDI optical transceivers, a keypad, a 12x2 LCD display and a RS-232 to USB converter. These are the elements in the RDI module mentioned in II.B.

The system gets the DAB ensemble from the RDI output of a DAB receiver. The optical transceivers convert the RDI optical signal into an electrical one, with the appropriate levels to be fed to the FPGA. The FPGA performs the IP datagram extraction and outputs them via the Ethernet port, the USB port, or both. The keypad and the LCD display are used for configuration and progress information.

#### E. Tests and results

In order to test the system, we have developed an RDI generator that is able to produce an RDI stream. The generator is composed of a software application, that runs on a Windows based PC, and a hardware subsystem (the DAB emulator mentioned in II.B), which has been implemented in an FPGA based prototyping board [13].

The application creates a DAB ensemble which contains only data services. There can be any number of those services, and each of them gets the data from its own source, either a file or a network stream. All the ensemble parameters are

configurable. The application creates the service information data (to be transmitted into the FIC) and sends it to the hardware subsystem, where it is stored. From there on, the application arranges all the incoming data flows into the MSC and transmits it continuously to the hardware subsystem. More details about this framework can be found in [14].

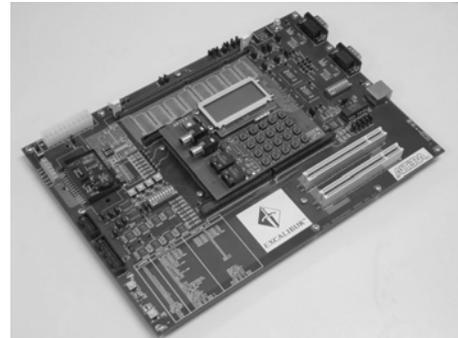


Fig. 5 DAB/DMB receiver platform.

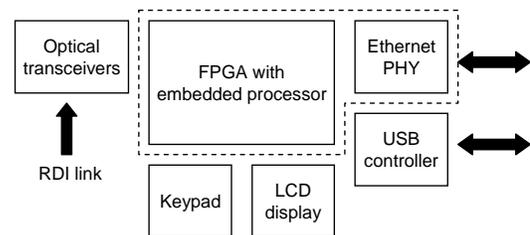


Fig. 6 Block diagram of the DAB receiver platform.

We have used VideoLan Client [15] to stream MPEG-4 encoded video in CIF resolution (352x288) at 25 frames per second with a bit rate of 256 kbps. The stream was sent to the client (a notebook, as the Ethernet port of the receiver is still in development, running also VideoLan Client) through the RDI generator and the IP datagram extractor. The subjective image quality was good when playing the video at its native format, which is well suited for PDA-size receivers. The testbench is shown in Fig. 7. The RDI generator is composed by the desktop PC and the prototyping board atop it. The DAB/DMB receiver is located on the shelf, to the right of the notebook.



Fig. 7 Testbench of the DAB/DMB receiver

#### IV. VIDEO DECODER

In this working area, an optimization methodology for video decoding using DSPs as a target technology has been developed. An MPEG-4 video decoder has been implemented using this methodology. The decoder has been tested in real-time using a DSP platform designed to support the DSP research activities in ARTEMI.

##### A. The DSP platform

The DSP platform, mentioned before in II-B, can be seen in Fig. 8. A block diagram is shown in Fig. 9. The platform is based on a TMS320DM641 DSP [16] from Texas Instruments, running at 600 MHz. The DSP interfaces with an Ethernet port, two external memories, a video encoder, an audio digital to analog converter<sup>1</sup> (DAC), and a JTAG emulator using a minimum amount of glue hardware. The DSP reads from the Ethernet port an MP2TS encapsulated over IP/UDP that contains the video stream. The DSP outputs the video data to the encoder in ITU-R BT.601 format. The video encoder generates composite video (CVBS) and S-video (Y/C) to interface with a standard TV set.

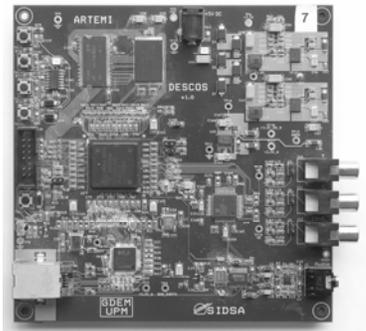


Fig. 8. The DSP platform

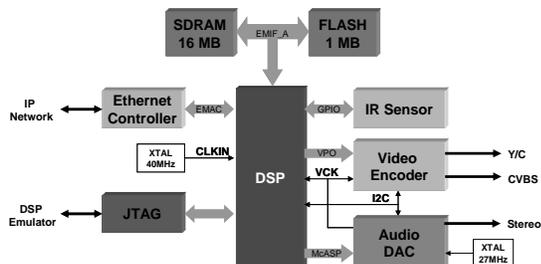


Fig. 9. Block diagram of the DSP platform

##### B. The MPEG-4 decoding algorithm

The MPEG-4 video coding standard [17] has 19 profiles and 18 levels. The profiles define sub-sets of the video coded syntax while the levels define limits on the values of the stream syntax elements.

In ARTEMI, we are interested in two profiles used to encode rectangular natural video sequences: Simple Profile (SP) and Advanced Simple Profile (ASP). The SP supports I and P frames with progressive 4:2:0 format and  $\frac{1}{2}$  pel motion

<sup>1</sup> Though ARTEMI does not consider working with audio, this feature has been added for completeness, as the DSP platform may be useful in other projects.

compensation. The ASP supports also interleaving format, B frames,  $\frac{1}{4}$  pel Motion Compensation (MC) and Global Motion Compensation (GMC). The different levels support spatial resolutions, in pels, from 176×144 to 352×288 (SP) or 720×576 (ASP).

The MPEG-4 SP/ASP decoding algorithm is outlined in this subsection for reference. As can be seen in Fig. 10, after the picture header parsing, the algorithm enters in a loop to decode each macroblock. Several buffers are used in the decoding process: one to store the video stream (Elementary Stream), a macroblock buffer (Block) used in the decoding of coefficients and Inverse Discrete Cosine Transform (IDCT) computing, several intermediate macroblock buffers used in the motion compensation process (intermediate buffers for MC), a buffer to store the reconstructed macroblock (Reconstructed MB) and, finally, a triple buffer to store the Y, U, V components of the reconstructed image (Ref\_frames). In the figure, a grey scale code has been used to link the different processes (i.e. Decode Motion Vectors, IDCT...) with the buffers used in each one of them.

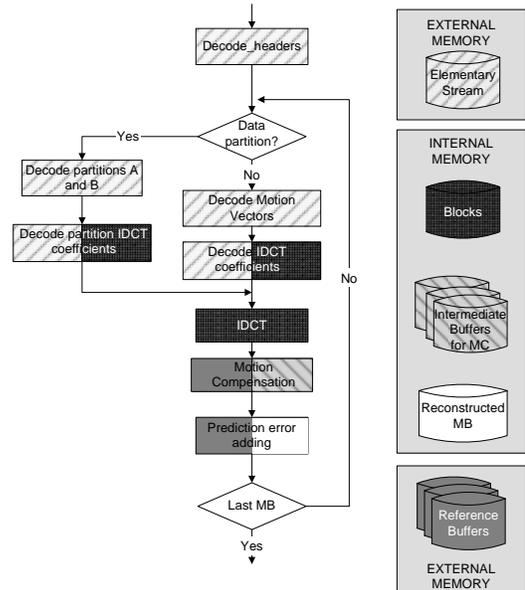


Fig. 10. Block diagram of the decoding algorithm

##### C. The MPEG-4 video decoder optimization process

The starting point of the optimization process was an MPEG-4 SP/ASP standard compliant raw-C code implementing the algorithm outlined in the former subsection. It was first fully tested in a PC and then ported to the DSP.

Initially, this first version of the video decoder was located entirely in external memory and spent 50 to 300 million clock cycles per frame to decode MPEG-4 ASP @ L5 streams (D1 format). With this performance, only 2 to 12 frames/sec could be decoded using a DSP with a 600 MHz system clock.

From this starting point, the optimization process was developed in four steps:

- **First step: compiler optimizations.** Compiler was configured to optimize the execution speed. Moreover several framework tools and specific directives for the

precompiler have been applied. Using these resources, a speedup of about 60% was achieved.

- **Second step: memory management optimization.** Internal DSP memory was configured to split into cache and code/data memory. After several iterations 2<sup>nd</sup> level cache was dimensioned. The Elementary Stream buffer and the Reference Buffers (see Fig. 10) were located in external memory because of their size. The other buffers (Fig. 10) and most functions were fitted in internal memory. With these assignments an additional speedup of about 50% was achieved.
- **Third step: function optimization and assembly coding.** An optimized function from the DSP vendor was used to implement the IDCT. Moreover, critical parts of the algorithm ( $\frac{1}{2}$  pixel interpolation arithmetic and prediction error addition and saturation) were re-encoded in the DSP assembly code. An additional speedup of about 20% was achieved.
- **Fourth step: Explicit DMA transfers.** In this step, DMA transfers are executed in parallel with CPU processing. Explicit DMA transfers are used in the motion compensation to read the reference blocks from external memory and in the reconstruction process to write the decoded frames also to the external memory. Special care was taken with both, data alignment and balance among DMA queues [18].

Before fourth step, the decoder performed the operations A to F in a serial way, as can be seen in Fig. 11-a. In fourth step, the decoding of a macroblock (operations A, B, and C in Fig. 11-b) was scheduled to be executed in parallel with the DMA transfer of the former reconstructed macroblock to external memory (operation F). Also, the IDCT computing (operation C) was scheduled in parallel with the reference macroblock transfer (operation D).

In this step, an additional speedup of about 40% was achieved.

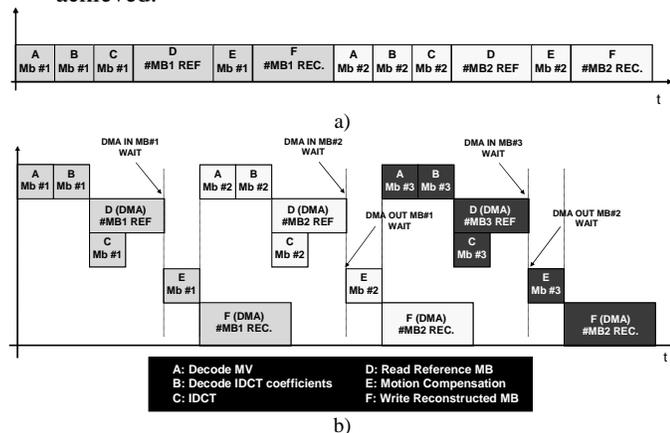


Fig. 11. Parallelization of video decoding process: a) serial, implemented in optimization steps before fourth b) parallel, implemented in the fourth step

After all the optimizations, a speedup of about 90% with respect to the initial code was achieved. More detailed performance data are given in the next section.

#### D. Functional and performance tests

A set of functional and performance tests have been implemented for the optimized decoder. The results of these tests guarantee the conformance with MPEG-4 SP and ASP. Also, different MPEG-4 SP/ASP streams have been decoded in real-time using the DSP platform.

##### 1) Functional tests

In order to verify the functionality of the optimized decoder, conformance tests have been implemented using the set of standard sequences defined in the MPEG-4 standard [19] as inputs. These sequences were decoded with the optimized decoder running on a PC using the Code Composer Studio framework debugger [20] and the results were inspected with a viewer. All sequences were decoded correctly.

Furthermore, for some of these sequences (shown in Table 1), the optimized decoder was compared with the ISO reference decoder [21].

TABLE 1  
SOME SEQUENCES USED IN THE CONFORMANCE TESTS.

Sequence	Profile & Level	Size	Type of frames	MC	Type of frames	Other tools
GE-16	S@L2	352x288	I+P	$\frac{1}{2}$ pel	Progr.	-
A1GE-10	AS@L2	352x288	I+P	$\frac{1}{4}$ pel	Progr.	-
A1GE-11	AS@L1	352x288	I+S-GMC	$\frac{1}{4}$ pel	Progr.	GMC
A1GE-12	AS@L1	352x288	I+B	$\frac{1}{4}$ pel	Progr.	-
er-3	S@L3	352x288	I+P	$\frac{1}{2}$ pel	Progr.	Data part. + RVLC

The sequences shown in Table 1 were first decoded using the ISO reference decoder running on a PC. Then, the same sequences were decoded using our optimized decoder running also on a PC using the Code Composer framework debugger. The PSNR was computed to compare the outputs of both decoders (using the output frames from the ISO decoder as a reference). In all sequences, mean PSNR values above 50 db were obtained for Y, C<sub>r</sub> and C<sub>b</sub>.

##### 2) Performance tests

The standard sequences shown in Table 1 have been also used to test the decoder performance. The average number of clock cycles needed to decode a frame was measured after each one of the optimization phases described in section IV-C. The Code Composer Studio framework profiling tool running on a PC was used to perform the measures. The results are given in Fig. 12.

The same tests have been performed running the optimized decoder on the DSP platform instead of a PC, with the same results. With a 600 MHz system clock, real-time can be achieved for all sequences using less than a 25% of the DSP computational power.

Finally, real-time tests have been performed using a PC server to stream actual DVD movies to the DSP platform, with excellent results. In these cases, VideoLan Client [15] was used to transcode the DVD movies from MPEG-2 to MPEG-4 “on the fly” (in D1 format @ 25 frames/sec, with I, P & B frames and  $\frac{1}{4}$  pel motion compensation). The same tests have

been performed using the Foreman sequence encapsulated over MP2TS and streamed by the PC server. All the streamed video sequences were decoded by the DSP platform in real-time and displayed on a TV set (see Fig. 13).

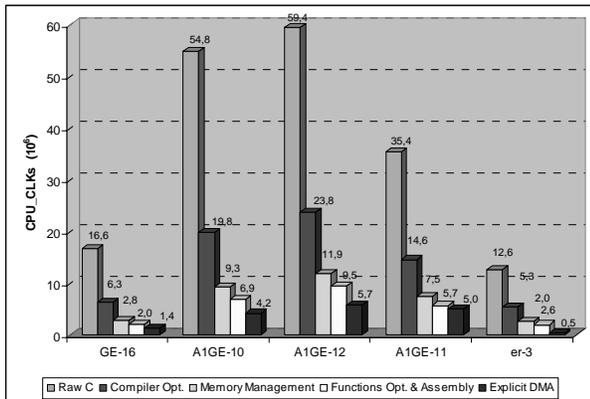


Fig. 12. Average number of clock cycles ( $\times 10^6$ ) needed to decode a frame for the sequences shown in Table 1.



Fig. 13. Testbench used for real-time tests

## V. FUTURE WORK

Our near term future work is focused on the completion of some tasks related with the DAB/DMB receiver as well as the implementation of system level tests.

### A. Work related to the DAB/DMB receiver

The DAB/DMB receiver will be completed in order to enable the reception of video data properly. Both options mentioned in III.B –DMB and video over IP using the Enhanced Packet Mode– will be implemented, adding the RS decoder and deinterleaver blocks (see Fig. 4) to the datapath. These new features will be validated by performing the tests described before in III.E.

### B. Work related to the system level tests

Using the DSP platform described in Fig. 2, a complete set of tests will be performed to validate both, the DAB/DMB receiver and the MPEG-4 video decoder at system level. An Ethernet interface (shown in Fig. 4) will be implemented in the FPGA-RISC platform so as the video data obtained by the DAB/DMB receiver may be forwarded to the DSP platform.

## VI. CONCLUSION

In this paper, a demo platform for reception of real-time video over DAB/DMB implemented in the context of ARTEMI project has been presented. The platform consists of a DAB/DMB receiver block and an MPEG-4 decoding block. Both blocks have been tested separately with excellent results. In the near term, these blocks will work together in the reception of video streams supporting DMB (a recently published standard) and the video over IP using the Enhanced Packet Mode (a future standard).

## ACKNOWLEDGMENT

The authors would like to thank M<sup>a</sup> Cruz Tejedor, David Samper, Javier Iglesias and Rafael Antoniello, all from UPM, for their contributions to this work.

## REFERENCES

- [1] ARTEMI project. <http://www.iuma.ulpgc.es/artemi>.
- [2] ETSI EN 300 401 v1.3.3: "Radio broadcasting systems; Digital Audio Broadcasting (DAB) to mobile, portable and fixed receivers". May 2001.
- [3] ETSI TS 102 427 v1.1.1: "Digital Audio Broadcasting (DAB); Data Broadcasting; MPEG2 TS streaming". July 2005.
- [4] ETSI TS 102 428 v1.1.1: "Digital Audio Broadcasting (DAB); DMB video service; User Application Specification". June 2005.
- [5] DRX 701 ES DAB receiver from Pure Digital . <http://www.videologic.com/Products/Product.asp?Product=VL-60640>.
- [6] EN 50255: "Digital Audio Broadcasting system; Specification of the Receiver Data Interface (RDI)". Dec. 1997.
- [7] B. Sostawa, J. Speidel, "Investigations on bit error performance for video over DAB". IEEE Trans. on Broadcasting., Vol.44, Dec 1998, pp. 445-448.
- [8] RFC 3550: "RTP: A Transport Protocol for Real-Time Applications". Jul. 2003.
- [9] RFC 2250: "RTP Payload Format for MPEG1/MPEG2 Video". Jan. 1998.
- [10] ETSI ES 201 735 v1.1.1: "Digital Audio Broadcasting (DAB); Internet Protocol (IP) datagram tunnelling". Sep. 2000.
- [11] Altera Corporation. "EPXA10 Development Board. Hardware Reference Manual v1.1". Apr. 2002.
- [12] Altera Corporation. "Excalibur Devices. Hardware Reference Manual v3.1". Nov. 2002.
- [13] C. Sanz et al. "Advanced Tools for Digital Electronics Teaching". 4th European Workshop on Microelectronics Education, EWME 2002. Microelectronics Education, pp 189-192, Ed. Marcombo.
- [14] P. J. Lobo et al. "The Prototyping Methodology of a Data Receiver for Digital Audio Broadcasting (DAB) Networks". Accepted for publication in the 17th IEEE International Workshop on Rapid System Prototyping, RSP 2006.
- [15] VideoLAN Client v.0.8.4. <http://www.videolan.org>
- [16] TMS320DM641/TMS320DM640 Video/Imaging Fixed-Point DSPs. Data Manual. (SPRS222D-Apr. 2005). <http://focus.ti.com/lit/ds/symlink/tms320dm641.pdf>
- [17] ISO/IEC 14496-2. "Information technology - Coding of audio-visual objects - Part 2: Visual". 2004.
- [18] TMS320C6000 DSP Enhanced Direct Memory (EDMA) Controller. Reference Guide. (SPRU234-March 2005). <http://focus.ti.com/lit/ug/spru234b/spru234b.pdf>
- [19] ISO/IEC 14496-4. "Information technology - Coding of audio-visual objects. Part 4: Conformance testing". First Edition: Dec 2000. Amendment 1: Jun 2003.
- [20] Texas Instruments Incorporated. "Code Composer Studio v3.1 IDE Getting Started Guide" (SPRA795A-April 2003). May 2004.
- [21] ISO/IEC 14496-5. "Information technology - Coding of audio-visual objects - Part 5: Reference software for MPEG-4".