INTERACTIVE DSP EDUCATIONAL PLATFORM FOR REAL-TIME SUBBAND AUDIO CODING*

Davide Quaglia[§], *Alfonso Montuori*[†], *Eros Pasero*[†], *Juan Carlos De Martin*[‡]

[§]Dipartimento di Automatica e Informatica/ [†]Dipartimento di Elettronica/ [‡]IRITI-CNR
Politecnico di Torino
C.so Duca degli Abruzzi 24 — I-10129 Torino, Italy
E-mail: `[davide.quaglia|pasero|demartin]@polito.it`, `montuori@lep.polito.it`

ABSTRACT

We present an interactive educational DSP platform for real-time subband audio coding. The tool integrates a DSP course on advanced audio coding and focuses on wavelets, subband coding, and signal quantization. The package consists of a Windows control application running on a PC and real-time DSP code implemented on a Texas Instruments DSP Starter Kit (DSK), which performs the actual audio encoding and decoding process. The tool is interactive since students can customize the codec by changing some encoding parameters through the PC interface and then verify the effect on the coding process. The DSP codec accepts audio samples in real-time via the DSK audio analog input port and sends the reconstructed samples to the output port. The graphical user interface allows to choose the type of filter-bank and scalar quantization; the availability of the full source code and of a DSK makes possible to experience first-hand the main aspects of real-world DSP development, e.g., real-time programming, arithmetic precision and algorithmic delay. The software is available at `http://multimedia.polito.it/dspwavelab`.

1. INTRODUCTION

New digital audio applications are changing the way we experience entertainment and broadcast services. Digital audio, for a long time limited to the compact disc arena, is finding myriad of new ways to play a role in our life, from MP3 portable players to Digital Audio Broadcasting radio receivers. The key enabling technology for these new exciting products is audio compression, which makes possible to store and transmit the otherwise unwieldy audio signal with levels of sound fidelity unthinkable until just a few years ago.

Audio coding is thus becoming an important topic in EE curricula, being at the same time a precious know-how as well as an attractive topic for engineering students. Audio coding is a good subject for educators, too, since it permits to introduce and discuss important DSP topics such as sub-band decomposition, scalar and vector quantization, sample-based coding schemes. Among different ways to perform audio coding, in fact, frequency-domain analysis through subband decomposition is one of the most effective. Wavelet theory gives a more formal approach to subband decomposition showing the relationship between filter banks and wavelet bases [1][2][3]. The design of a subband coder involves the choice of a filter bank and of a coefficient encoder (e.g., PCM, APCM, ADPCM). In particular, a filter bank is characterized by its main structure (uniform-band vs. octave-band filter bank), the type and the order of the filters, their implementation and the number of bands.

Most of the existing teaching tools [4][5][6] cover basic DSP concepts. We wanted to offer a teaching aid at the same time more advanced as far as topics were concerned, and closer to real-time DSP programming, to enhance the value of the overall educational experience. The use of a DSP board for this purpose was a natural choice, which has already proved to be effective in previous experiences [7][8].

We present an audio coder/decoder based on the wavelet transform for both research and teaching purposes. The software consists of an audio codec and of a host application. The former runs in real-time on the Texas Instruments TMS320C6211 DSP Starter Kit (DSK) board, while the latter is a Windows application running on the host PC to which the DSP board is connected. The codec accepts audio samples from the analog input of the board, performs subband coding and decoding and, finally, sends the decoded samples to the analog output. Students can, therefore, directly evaluate the distortion and the delay introduced by the codec, as well as issues regarding real-time computation. The host application is used to download the codec on the DSP board and to set the coding parameters. Through these parameters, it is possible to change the type of wavelet, the algorithm used to code the wavelet coefficients and the cod-

*This work was partially supported by Texas Instruments Inc. which provided DSP equipment under the TI University Program.

ing performance for each subband.

The paper is organized as follows. Section 2 deals with the hardware architecture of the system. Section 3 describes the architecture of the application running on the DSK with details on subband decomposition and quantization of wavelet coefficients. In Section 4 the host application is presented. Section 5 reports some results. Finally, conclusions are presented in Section 6.

2. HARDWARE ARCHITECTURE

Figure 1 shows the layout of the system. It consists of the DSK board connected to a PC through its parallel interface. The *codec* is the software application that runs on the board and encodes audio samples coming from the analog input (e.g., a tape player). The decoded audio samples are sent to the analog output (e.g., a loudspeaker or a headphone). The PC runs the *host application*, which initializes the DSK board, downloads the codec, and sends encoding parameters to it. One of the main advantages of the DSK board is that any PC with a parallel port can be used to program and control it; a DSK board connected with a notebook computer can thus become a portable kit for demonstrations and lessons.

The DSK board is equipped with the 32 bit - 150 MHz - fixed point 'C6211 DSP and with 4 MB External SDRAM. A parallel port controller is the gateway between the Host Port Interface (HPI) of the DSP and the PC. The analog input port consists of an analog-to-digital converter; it supports the output of a normal player and stereo electret microphones with voltage bias; sampling rate is 8 kHz and audio samples are quantized on 16 bits. The loudspeaker driver has a programmable gain amplifier. Analog/digital converters are connected to the DSP chip through its Multichannel Buffered Serial Port (McBSP).

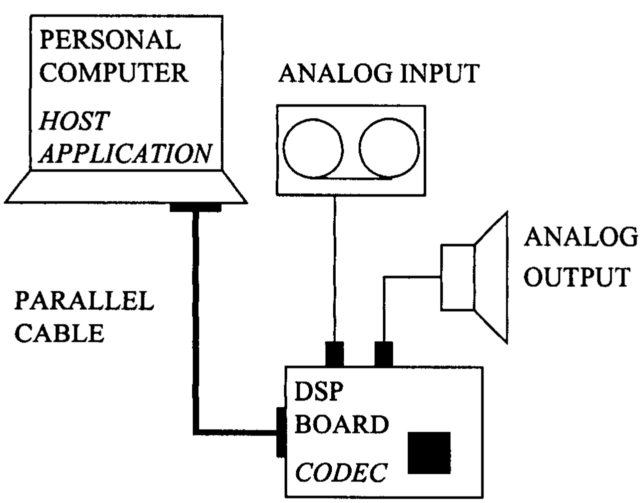

Fig. 1. Layout of the system.

3. SOFTWARE ARCHITECTURE OF THE CODEC

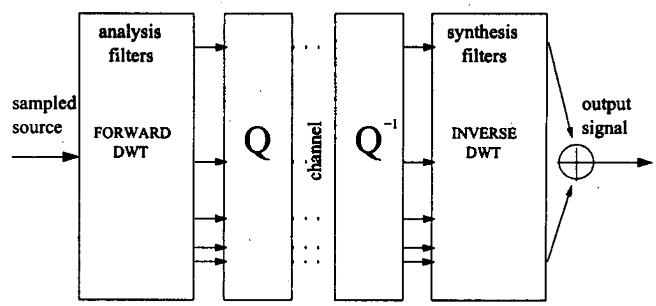

Fig. 2. Block diagram of the audio codec.

The software architecture of the codec consists of three modules. The module *CnfDSP* handles the communications with the host program through the Host Port Interface (HPI). In fact, the host program sends parameters to the codec using the HPI. The module *Codec_edma* initializes and uses the on-chip DMA controller to transfer blocks of 64 samples from the analog input port to memory and from memory to the analog output port. This module contains the interrupt service routine that is called when a new block of samples is available in memory for processing. Finally, the module *Audio* contains the actual subband encoder/decoder that is called by the interrupt service routine. A block of 64 samples (called *frame*) is taken by the encoder and produced by the decoder; to compensate random delays, a circular buffer of 18 frames is placed before the encoder and after the decoder.

Figure 2 shows the block diagram of the subband encoder and decoder. The frame is transformed into a time-frequency representation through the discrete-time wavelet transform (DWT). The audio samples are first transformed by the analysis section into a time-frequency representation; the coefficients of the time-frequency representation are then quantized and this is the actual compression step. The encoder is directly connected to the decoder. At the decoder, de-quantized data are transformed into the real-world time-domain representation by the synthesis filters. The quantization process introduces an irreversible data loss which is perceived as a distortion in the reconstructed audio signal after the synthesis section.

A major advantage of the frequency-domain approach is that it can directly incorporate knowledge about the human auditory system. From an audio coding perspective, the most important property of the ear is the noise-masking effect in which the ear masks noise components (i.e., quantization noise) that are close in frequency to signal components. In the case of speech, energy concentrates into different bands for voiced and unvoiced sounds. This uneven and time-varying distribution of the signal energy provides motivation for using an adaptive subband coding in which *i)* a different number of bits is used to encode each frequency component; *ii)* for a given band, the number of bits changes on a frame-by-frame basis. The main advantage of the wavelet transform over the Fourier transform is its ability to represent the signal on both time and frequency domains within Heisenberg's uncertainty limits. It has been shown that wavelets can approximate time-varying non-stationary signals in a better way than the Fourier transform [1]. The coding process entails obtaining a frame of 64 speech samples, which are transformed in a time-frequency representation by the analysis filter bank. We used an m-level tree-structured filter bank which leads to the decomposition of the input signal into $m+1$ sub-sequences each having a halved resolution with respect to the previous one. This subdivision, called *dyadic subdivision*, is accomplished by recursively dividing each sequence into a sub-sequence containing its approximated version and a sub-sequence with the residual details. The resulting signal representation is not redundant since each sub-sequence is long half of the original one (sub-sampling occurs below each filter).

The synthesis bank reconstructs the audio signal; starting from the lowest resolution, the low-pass component of the signal is added to the detail component in order to obtain the low-pass component with double resolution. This operation is repeated till the full resolution signal is reconstructed. The coefficients are oversampled before being filtered and added each other.

3.1. Coding of the wavelet coefficients

Three different approaches have been adopted to code the wavelet coefficients. All these techniques are based on a scalar uniform quantization and the number of bits per coefficient is decided for each subband and changes over time to satisfy a user-defined parameter.

In the *constant SNR* coding the basic idea is to vary the number of bits per coefficient in each subband in order to keep the audio quality constant. In this scenario, the user specifies a SNR value for each subband. The SNR is related to the number of bits per coefficient and to the actual energy of the signal according to the following equation:

$$\begin{aligned} \frac{\frac{1}{N(i)}\sum_{n=1}^{N(i)} s_n^2}{\sigma^2_{\Delta(i)}} &= SNR(i) \qquad (1) \\ \textit{where} \quad \sigma^2_{\Delta(i)} &= \frac{\Delta_{max}(i)^2}{3}, \\ \textit{and} \quad \Delta_{max}(i) &= \frac{D(i)}{2^{b(i)}}. \end{aligned}$$

In (1), for the i-th subband, $N(i)$ is the number of coefficients, s_n is the n-th coefficient, $\sigma^2_{\Delta(i)}$ is the variance of the quantization error (in the case of uniform distribution), $\Delta_{max}(i)$ is the maximum quantization error, $SNR(i)$ is the imposed SNR, $D(i)$ is actual range of the unquantized coefficients, and $b(i)$ is the number of bits per coefficient. The computed value of $b(i)$ changes at any block to guarantee the desired SNR; therefore, $D(i)$ and $b(i)$ have to be transmitted to the decoder (forward-adaptive approach).

In the *constant bitrate (CBR)* coding the user chooses the number of bits per coefficient for each subband, and the type of scalar quantizer, mid-riser or mid-tread.

4. HOST APPLICATION

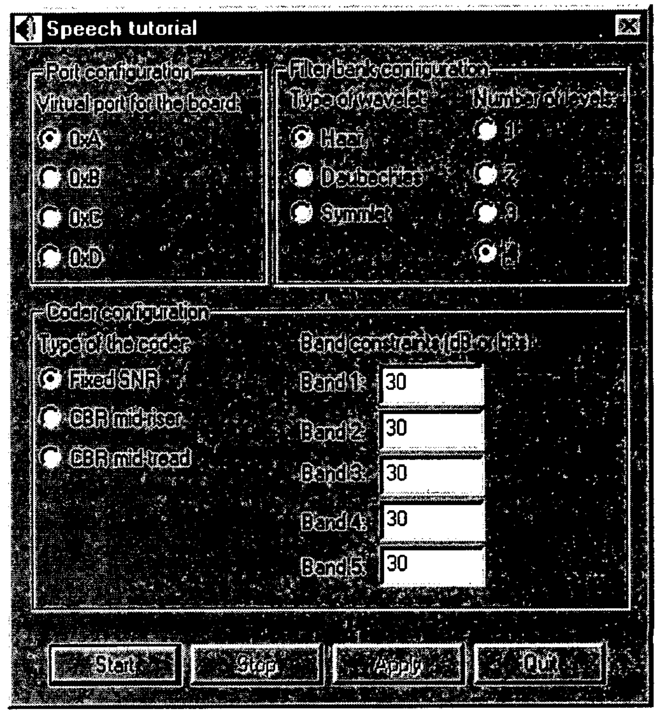

Fig. 3. Graphical user interface of the host application.

The host application graphical user interface consists of a window, shown in Figure 3. The user sets a number of parameters grouped into three frames. Through "Port configuration," the user can set the virtual port to which the DSK is connected; 0x0a is the default choice (details on this topic can be found in the DSK documentation [9]). Through "Filter bank configuration," the user can choose the type of wavelet to be used for the subband decomposition and the depth m of the analysis and synthesis trees. Three kinds of wavelet are available, Haar being the default choice. A thorough explanation of wavelet theory and subband decomposition can be found in, e.g., [1][2][3]. The depth of the analysis/synthesis tree affects the number of subbands: with a tree of m levels, $m+1$ subbands are created. The default value for this option is four. "Coder configuration" sets the algorithm used to code the wavelet coefficients in each subband; all the algorithms perform a uniform scalar quantization of the wavelet coefficients. The first algorithm maintains a constant signal-to-noise ratio (SNR) in each subband; for each subband the SNR value (in dB) is specified by the user. The other two algorithms (CBR mid-riser and CBR mid-tread) use a constant bitrate scheme; for each subband the user specifies the number of bits per coefficient.

By pressing the Start button, the codec is downloaded on the DSP board and the host application transmits the chosen parameter values to it. Finally, the button labeled "Quit" closes the host application.

5. RESULTS

This Section shows how the application can be used in a DSP course. The main aspects involved in the actual implementation on a DSP system are real-time programming, algorithmic delay, and arithmetic precision. In a real-time application there are time constraints that must be satisfied; in the case of our codec all the operations (analysis, quantization, de-quantization and synthesis) have to be performed before a new block of samples is provided by the analog-to-digital converter. In particular, with 8 kHz sampling rate, for blocks of 64 samples, the available time-slot is 8 ms. Students can try to reduce this interval using smaller block sizes or increase the computational load of the processor and observe what happens when real-time constraints are not satisfied.

The block size, as well as the order of the filters and the number of levels of the filter tree, affect the algorithmic delay of the system. This parameter is very important in full-duplex communications e.g., phone conversations. On the other hand, larger blocks improve compression since more data are available for frequency analysis. Students can investigate this trade-off between delay and performance by modifying the codec.

The TMS320C6211 is a fixed-point DSP; in our work, we use floating-point operations emulated by a standard software library, at the cost of additional CPU load. If other scenarios, a fixed-point implementation might become necessary. In this case, special attention should devoted during filter design, to avoid either loss of precision or overflow during convolution.

6. CONCLUSIONS

An interactive educational real-time subband audio coding platform has been implemented on the Texas Instruments TMS320C6211 DSP Starter Kit. The codec is controlled by an application running on the host PC, which lets the users customize most aspects of the coding process. Users can apply an audio signal to the analog input of the board and hear the resulting output in real-time using a loudspeaker. The application can be used in DSP courses, with focus on wavelets, subband coding, and signal quantization. The tool also allows students to strengthen their understanding of the main aspects of real-world DSP development, e.g., real-time programming, arithmetic precision and algorithmic delay.

7. REFERENCES

[1] I. Daubechies, "Orthonormal bases of compactly supported wavelets," *Communications on Pure and Applied Mathematics*, vol. 4, pp. 909–996, Nov. 1988.

[2] S. Mallat, *A wavelet tour of signal processing*, Academic Press, San Diego, 1998.

[3] M. Vetterli and J. Kovacevic, *Wavelets and Subband Coding*, Prentice Hall, Englewood Cliffs, NJ, 1995.

[4] Digital Signal Processing Group, *DSP Teaching Tools*, Rice University, http://www-dsp.rice.edu.

[5] Y. Cheneval, L. Balmelli, P. Prandoni, J. Kovacevic, and M. Vetterli, "Interactive DSP education using Java," *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, vol. 3, pp. 1905–1908, 1998.

[6] D. Donoho, M. R. Duncan, X. Huo, and O. Levi, *Wavelab 802 for Matlab 5*, Statistics Department, Stanford University, http://www-stat.stanford.edu/~ wavelab.

[7] M. G. Morrow and T. B. Welch, "WinDSK: a Windows-based DSP demonstration and debugging program," *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, pp. 3510–3513, 2000.

[8] M. G. Morrow, T. B. Welch, C. H. G. Wright, and G. W. P. York, "Demonstration platform for real-time beamforming," *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, vol. 5, pp. 2693–2696, 2001.

[9] *TMS320C6211 Starter Kit User's Guide*, Texas Instruments Inc., 2000.