Bernhard Grill, Stefan Geyersberger, Johannes Hilpert, and Bodo Teichmann
Fraunhofer Gesellschaft, Institut fuer Integrierte Schaltungen
Erlangen, Germany

Presented at
the 109th Convention
2000 September 22-25
Los Angeles, California, USA

AN AUDIO ENGINEERING SOCIETY PREPRINT

Implementation of MPEG-4 Audio Components on various Platforms

Bernhard Grill, Stefan Geyersberger, Johannes Hilpert, Bodo Teichmann

Fraunhofer Gesellschaft, Institut fuer integrierte Schaltungen, Erlangen, Germany

Abstract

The specification of MPEG-4 Audio, the next generation audio coding technology, has been finalized by ISO/IEC. This paper investigates implementation options of the new standard on various platforms. Special attention is given to the new Low-Delay AAC-Coder variant and the MPEG-4 Scalable Profile, and to several transport options for MPEG-4 Audio content.

1 Introduction

The specification of MPEG-4 Audio, the next generation in the meanwhile long line of MPEG standards, has been frozen in December 1999. Now the integration into new products starts. Compared to MPEG-1 and MPEG-2, which focused on the low bitrate coding of high quality audio and visual content, MPEG-4 covers a much broader range. As far as audio coding is concerned, this means the integration of speech coding and methods for synthetic material and new functionalities.

The broad range of MPEG-4 Audio leads to questions about implementation requirements and cost. The paper will describe the features and demands of various MPEG-4 Audio encoders and decoders, which have been implemented by the authors.

Implementation platforms include PCs running UNIX/Linux or Windows and code for various DSPs.

2 MPEG-4 Audio

2.1 History and Versions

In 1991 and 1994 world wide standards for perceptual audio coding were normalized in ISO/IEC SC29 IS 11172-3 (MPEG-1 Audio) [1] and IS 13818-3 (MPEG-2 Audio) [2]. These

systems were capable to deliver high sound quality at rather low bit rates. As far as bit rate efficiency is concerned, they may now be considered as the first generation of perceptual audio coding. A second generation of MPEG audio, Advanced Audio Coding (AAC), was standardized in 1997 [3]). Formally AAC is an extension to MPEG-2, but actually it is a true second generation coder, which is not backwards compatible to the previous MPEG standards. Meanwhile the MPEG audio coder family has been further expanded with the availability of MPEG-4 Audio [4]. MPEG-4 Audio is quite different, compared to its predecessors. Although high quality audio coding (called General Audio Coding in MPEG-4) is still a core functionality, as MPEG-2 AAC is an integral component, it additionally contains a lot of new functionalities. These include bit rate scalability modes, which allows the partial decoding of useful subsets of the bitstream. Moreover, speech coding algorithms as well as methods for synthetic sound generation are included. This makes MPEG-4 Audio the first standard covering the complete range of today's low bit rate audio coding applications. Therefore it is hoped that MPEG-4 Audio will allow for a much improved inter-operability between applications using low bit rate coding.

MPEG-4 has been standardized in two steps. MPEG-4 "Version 1" [4], which includes the scalable audio coding modes as well as the speech coding and synthetic coding techniques, has been technically frozen in December 1998. "Version 2" [5], formally an amendment to Version 1, reached the same status in December 1999. MPEG-4 Version 2 does not contain improved versions of functionalities already covered in Version 1. It only covers extensions which add capabilities not yet available at the time version 1 was finalized. Currently a re-write is under way which combines both versions into a single document. In the future, therefore, there will be only MPEG-4 Audio, without any version tag and the discrimination between Version 1 and Version 2 will become obsolete.

2.2 New Concepts: Object orientation of MPEG-4

The main focus of MPEG-1 and MPEG-2 has been the transmission and storage of traditional video and audio sequences, or in other words, on digitizing the existing analog video and audio broadcasting and storage techniques. MPEG-4 is based on a different concept. Basically an MPEG-4 decoder is a machine which composes an audio-visual scene from a number of audio or visual objects [6]. These may be either coded natural audio or visual sequences or artifically created objects. A two- or three-dimensional compositor unit places these objects into a virtual environment. All these objects exist only for a specified period in time.

Of course MPEG-4 can still support MPEG-1/2 style applications. E.g. TV-broadcast in MPEG-4 terminology means the transmission of one audio and one video object with unlimited duration. In this case the compositor is simply a synchronization unit for the two objects.

2.3 MPEG-4 Object Types

The terminology "Object Type" is used in MPEG-4 to distinguish between the different coding methods in which an MPEG-4 object can be represented with. The object type directly determines - without further parsing the bitstream - the MPEG-4 tool subset required to decode a specific object. The MPEG-4 profile definition (see section 2.5 below) is based on the object types, as each profile supports a certain list of object types. In a more traditional view the object types may be seen simply as the top level entry in the MPEG-4 Audio bitstream syntax. MPEG-4 Audio contains the following Object Types, which can be grouped in three classes (Natural General Audio, Natural Speech, Synthetic Material) :

General Audio Object Types The MPEG-4 variant of MPEG-2 AAC [3] is the most important coder type for general audio signals. In MPEG-4 the object type mechanism replaces the Profiles of the original MPEG-2 AAC standard to distinguish between the different MPEG-4 AAC flavors. Figure 1 gives an overview of the MPEG-2/4 AAC coder family. The three MPEG-2 AAC profiles are directly mapped to three MPEG-4 object types (AAC Main, LowComplexity(LC), SSR). These are almost identical to the predecessor versions. However, in MPEG-4 the Perceptual Noise Substitution (PNS [7, 8]) tool has been added for improved performance at low bitrates. Moreover, an alternative frame length of 960 samples in addition to the 1024 of MPEG-2 AAC is available in the MPEG-4 variant. All MPEG-2 AAC bitstream can be decoded by the corresponding MPEG-4 versions.

Furthermore, MPEG-4 contains the AAC LTP object type, which is a superset of AAC LC. AAC LTP adds a Long Term Predictor (LTP)[9] to the LC-variant. The LTP is a lower complexity replacement of the predictor of the AAC Main object type. Another AAC object type is called AAC Scalable, which uses a modified top level bitstream syntax and decoding process to support the bitrate scalability functionality. The intermediate and lower levels of the bitstream syntax and the decoder functions are identical to to the AAC LTP object type.

ER AAC LC, ER AAC LTP, and ER AAC Scalable are the Error Resilient (ER) versions of the above described object types, which have been added in Version 2. The decoding process is basically identical to the non-ER versions, however, the bitstream syntax has been rearranged to support unequal error protection and the decoder needs to be capable to handle this syntax. There is no ER version of AAC main, as this object type is not recommended for applications in error prone environments.

ER AAC LowDelay (see also section 6) signals an AAC version which has been scaled down from a block length of 1024 or 960 to a length of 512 or 480, respectively. With some other modifications this has resulted in a codec with an algorithmic delay of around 20 ms, which is low enough for two way communication.

For very low bitrates in the range from 6 to 16 kbit/s MPEG-4 Audio includes the TwinVQ [10] object type. MPEG-4 TwinVQ uses many of the AAC modules, like MDCT-filterbank and TNS and, therefore, can be seen as an alternative quantization and coding module in the AAC coder [11]. Like the AAC object types, TwinVQ has a more error resilient Version 2 counterpart, called ER TwinVQ.

Two more object types for the coding of general audio signals have been added in Version 2. The first one, ER BSAC, represents the Bit Sliced Arithmetic Coding Technology (BSAC) [12] and the second, ER HILN, signals material coded with the Harmonic and Individual Line plus Noise (HILN) Algorithm [13]. Both object types are not included in the implementations described in this paper and, therefore, are not covered here.

Speech Coder Object Types There are two basic speech coder technologies in MPEG-4, signaled by the object types HVXC and CELP. These abbreviations stand for Harmonic Vector Excitation Coding (HVXC [14]) and Code Excited Linear Prediction Coding (CELP [15, 16, 17]). Both Object types share some common modules, like the LPC-analysis/synthesis, but use different excitation modules. While HVXC aims at very low bitrate narrow band speech transmission in the range from 2 to 4 kbit/s the MPEG-4 CELP object type provides narrow band and wide band speech coding from around 4 to 24 kbit/s. HVXC clearly is intended for speech signals only. The CELP coder, in general, is more stable for non-clean-speech input material. For both, more error resilient object type versions have been added in Version 2. These are ER HVXC and ER HVXC. It can be speculated that in the long run all coder/decoder will support the ER version, as the additional for these overhead is relatively small.

Synthetic Object Types The object types signaling synthetically generated audio material: TTSI, Main Synthetic, Wavetable Synthesis, General MIDI and Algorithmic Synthesis are not covered by the implementations described in this paper and, therefore, are not covered here.

2.4 Scalability

Scalability in the context of MPEG-4 denotes the capability to decode a useful subset of a bitstream. This allows to adapt to the instantaneous channel capacity during transmission, independently of the encoding process. On the Internet, for example, it is possible to provide one encoded audio signal for the decoding of a high quality signal if the complete bitstream is transmitted, or - if the available channel capacity is low - for a still useful, although lower quality version of the encoded material. Similarly, for broadcasting applications a fall-back to a lower quality signal is possible in case of a transmission channel degradation. An important property of this concept is that the available channel capacity does not to be known at the time of encoding. The selection of the bitrate can be done any time and anywhere in the transmission chain without feed-back to the encoder or server site.

The following example illustrates a typical application scenario: An audio encoder or streaming server connected to the Internet broadcasts a radio news program. Four different types of receivers are expected:

1. A mobile phone connected to the net with a bit rate of 8 kbit/s
2. A PC using an analog modem connection of say 28.8 kbit/s

3. A PC connecting via an ISDN line with 64 kbit/s
4. An Internet Terminal on a local Network connected to the Internet with a high-speed DSL line

All these receivers can be supplied from the single MPEG-4 scalable encoder / streaming server which produces the following hierarchical, scalable bitstream layers:

1. One layer with 6 kbit/s using the CELP Object Type
2. The second layer adds 22 kbit/s using the AAC scalable objective with one mono channel
3. A third layer adds another 36 kbit/sec AAC scalable object type layer. This time, however, the layer is in stereo, extending the de-codable signal from mono to stereo.
4. The fourth layer adds another AAC layer with 64 kbit/s

The bit rate of all four layers adds up to a total rate of 128 kbit/s, which should give perfect audio quality, if all layers are decoded.

The audio bandwidth and the sound quality increases from one layer to the next. The first CELP layer delivers a good quality only for speech signals, the second layer a quality comparable to analog AM, the third one a FM-like audio quality and the last layer will deliver near CD quality. More information on scalability can be found in [18, 19, 20].

2.5 The MPEG-4 Audio Profiles and Levels

Due to the wide range of functionalities the implementation of the complete set of MPEG-4 tools is unlikely to happen within the next few years. Especially the synthetic audio component contains modes which have a significant complexity tag attached. However, already in MPEG-1/2 subsets were defined, like Layer 1,2 or 3 in Audio, or the Video profiles. These allowed the implementation of the standard on the limited hardware resources available at the time the standard was defined. MPEG-4 continues this approach. Currently for MPEG-4 Audio eight profiles are defined. There are four from Version 1: Speech, Scalable, Main, and Synthetic and four from Version 2: High-Quality, Maui, Low-Delay, and Natural. With the merging of both versions the total number might be reduced in the future, since there is some overlap. On the other hand special profiles for specific applications can be defined on request, if there is sufficient support for such a profile. Table 1 shows the object types contained in each profile.

In MPEG-4 each profile may have several levels, limiting some parameters of the tools present in a profile. In Audio these parameters usually are the sampling rate and the number of audio channels that can be decoded at the same time. It is beyond the scope of this paper to explain all the details of the levels of the audio profiles. All implementations described

Object Type	Speech	Scalable	Main	Synth.	High-Qual.	Maui	Low-Delay	Natural
AAC Main			X					X
AAC LC		X	X		X			X
AAC SSR			X					X
AAC LTP		X	X		X			X
AAC Scalable		X	X		X			X
ER AAC LC					X	X		X
ER AAC LTP					X			X
ER AAC Scal					X	X		X
ER AAC LD						X	X	X
TwinVQ		X	X					X
ER TwinVQ						X		X
ER BSAC						X		X
ER HILN								X
CELP	X	X	X		X		X	X
HVXC	X	X	X				X	X
ER-CELP					X		X	X
ER-HVXC							X	X
TTSI	X	X	X	X			X	
Main Syn.			X	X				
Wave Syn.			X	X				
Gen. MIDI			X	X				
Alg. Syn.			X	X				

Table 1: Object Types in each MPEG-4 Audio Profile

later on, however, support sampling rates up to 48 kHz or the highest sampling rate defined for a specific object type, respectively. Furthermore, at least two channels are possible per object.

In MPEG-4 the conformance points are hooked to specific profiles and levels. The coders and decoders described in this paper conform to several of these profiles/levels. The specific properties of the profiles realized in these implementation swill be explained shortly:

Scalable Profile The scalable profile, a Version 1 profile, is a rich, comprehensive subset of the MPEG-4 Audio standard, including most of the new features, like bit rate scalability and speech coding and can therefore be considered to give a good impression about the complexity of the various components of MPEG-4 Audio. It doesn't include, however, any synthetic coding methods. Also only the low-complexity variant of AAC is included. This profile is widely considered to be a good candidate for first real world applications of MPEG-4.

Speech Profile The speech profile of Version 1 is a subset of the scalable profile. As the name suggests, it only includes the MPEG-4 speech coders.

High Quality Profile The High Quality Profile is a Version 2 profile which is very similar to the scalable profile. However, it additionally contains the error-resilient versions of the Version 1 algorithms. On the other hand neither Twin-VQ nor HVXC are included. The CELP coder is included since for applications using very low bitrates the CELP coder provides superior quality for speech material.

Mobile Audio Intercommunication Profile (MAUI) Added in Version 2, this profile is tailored for applications which already have a speech coder available, like mobile telephones. It doesn't include any speech coder, but a comprehensive set of general audio object types. Although the CELP/AAC scalable combinations (see 2.4) are not available within this profile, it still contains the mechanisms to integrate other speech coders, like the GSM AMR coder, or G.729, or G.723.1. It has been verified by the authors that these can directly replace the MPEG-4 CELP coder in this role.

Low Delay Profile This profile targets two-way communication applications. Therefore, it includes only LD-AAC and the speech coders.

3 MPEG-4 Audio Transport Multiplex

Originally MPEG-4 didn't define any transport multiplex layer (Called TransMux in MPEG-4 terminology). While MPEG-1 and MPEG-2 include specifications for self-contained bitstream no such thing exists in MPEG-4. The specification for the coded representation and the decoding process of the audio and video objects and the scene composition description ends on an abstract level, where only the syntax and decoding method for an Access Units (AUs), the smallest decodable unit a decoder can work with (usually referred to as "frame" in other coder), is defined. In Version 1 the multiplexing and frame synchronization of these AUs has been completely outside of the MPEG-4 specification and part of an application specific transport layer, not defined in the MPEG-4. Basically this concept is still valid, although some multiplexing schemes have been added recently in Version 2, or are developed jointly between MPEG and other organizations, like the IETF. The implementations described in this paper support several of these options, which are briefly described in the following paragraphs:

MP4FF The MPEG-4 File Format (MP4FF) has been added to the MPEG-4 Systems specification in Version 2 [21]. It is intended to be the preferred method for the storage of any MPEG-4 content. MP4FF supports the full MPEG-4 functionality.

LATM The Low-overhead Audio Transport Multiplex (LATM) is an audio specific TransMux, added in Version 2 Audio, which serves two purposes: First, by defining a multiplex for multiple audio AUs it can be used to lower the overhead of the actual transport layer. E.g. instead of mapping single AUs to IP-packets a LATM multiplex unit can be carried on a

single IP-packet. Furthermore, it is used in the Low Overhead Audio Stream (LAOS), which provides a synchronization layer for LATM, to form a bitstream similar to the MPEG-1/2 audio bitstream. LATM only supports a subset of the MPEG-4 functionality, as it is limited to natural audio objects only.

FLEXMUX FlexMux supports the full MPEG-4 functionality and is an non-mandatory part of the MPEG-4 Version 1 Systems specification. It defines a multiplex for the audio and video AUs and the scene description and setup information and carries all the information necessary to synchronize the various objects. It does, however, not contain any mechanism, which allows re-synchronization e.g. in case of erroneous transmission or break in into an ongoing FlexMux stream. Therefore, for practical applications an additional TransMux layer is required to provide these functions. In the described implementation such a layer has been added. Basically it reserves one FlexMux channel for the MPEG-4 Systems initial object descriptor, which is inserted occasionally into the multiplex. This format seems like a good candidate for the connection of two MPEG-4 devices via a cable, if an IP based solution is not used.

RTP For the transport of AV-content over IP networks RTP, the Real-time Transport Protocol, has established itself as the primary choice. It is supported well by the Internet infrastructure. Techniques like RTP header compression will reduce the overhead in the future to a rather low value. Currently several options for the mapping of MPEG-4 AUs and the MPEG-4 System layer into RTP packets are discussed. The implementations described in this paper are based on either mapping LATM units or individual AUs into single RTP packets. Furthermore, currently several options to exploit the scalability feature of MPEG-4 Audio in conjunction with differentiated services on IP networks are explored based on the described implementations.

ADIF and ADTS In MPEG-4 terminology the MPEG-2 AAC ADIF and ADTS specifications just represent two more transport multiplex options. However, these can only be used together with the AAC object types. The described coder and decoder implementations support these, to allow inter-operability with MPEG-2 AAC coder and decoder.

Application Specific TransMux An application specific TransMux layer has been developed and integrated into encoder and decoder for the Digital Radio Mondial (DRM [22]) system. DRM is a digital broadcasting system with programs only containing a single object. Therefore, none of the complex MPEG-4 Systems functionality is needed for DRM and the main attention has been given to an optimal error protection scheme.

MPEG-2 Systems This TransMux option is mainly intended for broadcasting applications. The MPEG-4 AUs are mapped into a MPEG-2 Systems layer [23].

4 Decoder Implementation

4.1 Software Decoder

The implemented software decoder supports five of the eight MPEG-4 Audio profiles. These are: Scalable, Speech, High Quality Audio, and Low Delay and partially the MAUI profile, which at the time of writing this text is still missing the implementation of BSAC. The original implementation goal had been to realize a decoder for the Scalable profile. However, as can be seen from table 1 above, the additional overhead for the other four is relatively small, as most of the object types are already available in the Scalable profile. The decoder is available as a stand-alone command line program, or with a Qt-lib [24] based GUI program, or as a library which can be integrated in other applications. Supported operating Systems include Windows, Linux, and Solaris.

The starting point had been an optimized MPEG-2 AAC decoder for the MPEG-2 AAC Main and Low Complexity profiles. By adding the Perceptual Noise Substitution (PNS) Tool the decoder was made ready to decode the AAC Main and the AAC Low Complexity object types. The required modifications of the original code for PNS are relatively small, as only a noise generator and a scaling unit are required additionally. The additional computational complexity is more or less concentrated in the noise generation process, which doesn't add significantly to the required computing power. The current figures for the AAC-LC object type on a Pentium II processor suggest that a 40 MHz processor is sufficient for a stereo decoder at 48 kHz sampling rate, including PNS.

The addition of the LTP on the other hand significantly increases the computational complexity by around 50%, as an additional MDCT transform has to be calculated for each channel, if the LTP is active. The most complex coder combination with the Scalable object type is quite comparable to the LTP object type. If only AAC-layers are used the complexity is very close to the AAC-LC object type, since in total the same amount of bits has to be decoded and the inverse transform is identical. The remaining functions of such a coder do not add significantly to the memory and computational requirements. However, if a CELP coder is used as a first coding layer an additional MDCT is required. Since such a combination is only defined for a mono CELP layer, the worst-case computational load in this case is not burdened by the requirement to compute two MDCTs. On the other hand the CELP decoder has to be taken into account. However, since the most complex element of the CELP-decoder, the post-filter, is not used for such a combination the worst-case complexity of a scalable combination with a CELP coding layer is roughly equivalent to the AAC LTP object type, which in the worst-case requires two MDCTs.

The requirements of the TwinVQ object type, stand-alone or in a scalable combination with AAC layers, are quite similar to AAC-LC, or if the LTP is used to AAC LTP, as most of the components are identical for AAC-LC and TwinVQ. Only the re-quantization process of the spectral samples differs, but this doesn't add significantly to the required computational power.

Compared to other speech coder standards the implementation of the MPEG-4 speech

coders is relatively simple, as bit-exactness is not required to be compliant to the MPEG-4 standard. Instead other criteria have to be fulfilled which are much less dependent on the specific properties of the arithmetic unit of the processor. The computational complexity of the speech decoders is below that of the general object types.

For a software decoder memory is only an issue as far as the size of the executable is concerned. Work space memory is not an issue as the total required size is quite low for the natural audio object types and computers do have plenty of memory nowadays. The synthetic object types require much more work space, however, these are not included in the current implementation. The actual figures show around 800 kbyte for the executable for the complete decoder. However, a lot of debug code is still included and no attention has been given yet to memory size.

All object type decoder modules are re-entrant. Therefore multiple objects can be decoded at the same time.

4.2 Implementation on DSP

The AAC Scalable object type has been implemented on a Analog Devices Sharc DSP21060. Only AAC coding layers are available. The computational load on this DSP is below 25% for a stereo decoder at 48 kHz for up to eight layers (The maximum number of layers allowed in the standard). Since the figure is so low, the optimization process has been rather short and lower numbers may be possible.

The implementation includes application specific error concealment techniques and a special transport layer. For this reason it is hard to tell the memory requirements for the audio decoder part, since especially the error concealment functions are very tightly integrated with the rest of the audio decoder. However, the complete decoder fits into the internal memory.

5 Encoder Implementation

An MPEG-4 decoder is required to completely support a specific profile and level in order to guarantee the decoding of any material produced for this profile and level. This means the decoder must be capable to handle all object types of the profile and level and the full range of parameters, like bitrate or sampling rate. For an encoder the situation is different. Basically it is sufficient for to implement one object type at a single bit rate and sampling rate, if this is sufficient for a specific application. Consequently the encoder implementations described in the subsequent paragraphs are less complete than the decoders, although a considerable subset of MPEG-4 Audio has been integrated.

5.1 Software Implementation

The software encoder supports the following MPEG-4 Audio Object Types: HVXC, CELP, TwinVQ, AAC Main, AAC LC, AAC LTP, AAC Scalable and ER AAC LowDelay. It is available as a stand-alone command line program and as a library which can be integrated in other applications. Supported operating Systems include Windows, Linux, and Solaris. It is based on the software which was used to generate the bitstream for the MPEG-4 verification tests [25, 26].

The following object types have been integrated: HVXC, CELP, TwinVQ, and AAC in all variants. The encoder supports scalable coding with up to 8 layers. The first layer can be encoded with one of the following object types:

1. CELP, which gives the optimum quality for speech at bit rates below 16 kbit/s
2. TwinVQ Object Type for optimum quality for music at bit rates below 16 kbit/s
3. AAC-scalable, if the bitrate of the first layer is above 16 kbit/s

All remaining layers are always encoded with the AAC Scalable object type, whereas the bit rate for one single layer should be greater or equal to 16 kbit/s, although smaller step sizes are possible.

It is possible (and for the CELP Object Type mandatory) to encode a mono down-mix of a stereo input signal in the first layer. Subsequent layers can be in mono as well or can add one channel to deliver a full stereo signal.

The Encoder has two additional features, which can be very helpful:

1. A built in high quality re-sampler and low pass filter. This is useful if the input sampling rate needs to be adapted to the optimum sampling rate that fits to the desired target bit rate for the highest layer. Moreover, if a CELP/AAC-scalable combination is chosen the part of the signal that is encoded by CELP layer needs to be re-sampled at 8 or 7.35 KHz (depending on the AAC Object Type sampling rate)
2. A stereo preprocessor which helps to avoid coding artifacts especially for bit rates below 32 kbit/sec per channel.

Audio Quality The coder is based on the version used to generate all AAC object type bitstreams, including the scalable CELP+AAC combinations, for the MPEG-4 verification tests. The tests have shown that the audio quality for the AAC-Scalable object type is between single layer AAC and MPEG layer 3 (MP3), if 3 scalable layer are used. Generally spoken one can say that the more scalable layer are used the more audio quality is degraded compared to a single layer encoder. This is caused by the overhead required for additional side information and other effects which to describe would go beyond the scope of this paper.

5.2 Real-time Software Encoder

The MPEG-4 Audio coder described above is basically a non-real-time off-line Encoder. It is not optimized for high encoding speed, but just for highest audio quality.

Since there are a lot of applications needing real-time encoding, we have also developed a prototype real-time software encoder. It is derived from the MPEG-4 Software Encoder described above. For a few configurations it is optimized for higher encoding speed so that it runs in real-time on a Pentium 300. Still it is a "C" only implementation without any Pentium specific assembler optimizations, so additional optimizations are possible. Currently the software real-time encoder supports 3 configurations and the AAC Scalable object type only with a maximum of 2 scalable layers. These are:

1. One layer stereo at 96 kbit/s and 44.1 kHz sampling rate
2. One mono layer at 24 kbit/s and one stereo layer at 72 kbit/sec (total bit rate of 96 kbit/s) both at 44.1 kHz sampling rate
3. One mono layer at 24 kbit/s, one stereo layer at 16 kbit/s (total bit rate of 40 kbit/s) both at 44.1 kHz sampling rate. Other combinations could be added easily.

5.3 Implementation on DSP

Using the software real-time encoder described above as a template we have further developed a DSP-based real-time encoder running on two 40 MHz Sharc DSPs (Analog Devices 21060). This systems supports the same three configurations as the Software real-time encoder described above.

6 Low-Delay AAC

Derived from the MPEG-2 AAC scheme, MPEG-4 Low Delay (LD) AAC has been optimized for achieving a minimum end to end coding delay. During the evaluation phase of AAC LD various listening tests have shown a similar performance compared to MPEG-2 AAC at a bit rate about one third higher [27]. AAC LowDelay is included in the implementations described before. However, these do not yet reach the low end-to-end delay possible with this object type. However, the DSP implementation of AAC LowDelay, which is presented in the next paragraphs, does get quite close to the theoretical limits.

6.1 Delay in a Real-time Codec Chain

First the contributions of the various components in an AAC LowDelay transmission chain will be investigated. In figure 2 the delay in respect to the signal location in the transmission

chain is shown on the lower diagram. The values are related to a signal sampling frequency of 48 kHz. Given in numbers the individual contributors are:

1. Encoder input buffering: 10 ms
2. Encoder core: signal look-ahead, bit-reservoir, computation time: 0 + 1 + 10 ms
3. Encoder output buffering: 0 ms
4. Continuous transmission with typical network delay: 10 + 3 ms
5. Decoder sync validation: 1 ms
6. Decoder core: filterbank delay + computation time: 10 + 2.5 ms
7. Decoder output buffering: 0 ms

Looking at these numbers the goal of reaching an overall coding delay below 50 ms can be achieved even for a real-time implementation on a DSP. The contributions of limited computational performance, transmission delay and synchronization on streaming input data add an overhead of about 27.5 ms to the algorithmic delay of 21 ms. The algorithmic delay has been minimized compared to standard AAC by reducing the frame length to 480 samples, removing the signal look-ahead in the encoder, and using a minimum sized bit-reservoir. More details can be found in [28].

6.2 Implementation of MPEG-4 LD AAC on the Motorola DSP56300 family

Compared to the DSP implementation of MPEG-2 Advanced Audio Coding [29], MPEG-4 LD AAC can be characterized for the specific implementation by having an increase in computing time of about 25% together with a significant reduction of data memory resources. Major parts that influence the workload are the relation between call overhead and vectorized assembler functions caused by the shortened frame length. Additionally a certain transform overhead arises for the MDCT filterbank due to its non power of two length. Further on the computing time for the Error Resilience (ER) tools increases the necessary number of cycles. On the other hand the smaller frame length and the absence of 'short blocks' reduces the amount of data memory.

At a signal sampling frequency of 48 kHz the real-time requirements are fulfilled for the Stereo-Encoder running on a DSP 56300 with 100 MHz. The Stereo-Decoder requires 25 MHz. With respect to the computational delay a higher clock rate is preferred, since the processing time will be shortened. The Encoder has been implemented using about 32 kWords of Program and 70 kWords data memory. The Decoder currently uses 12 kWords of program and 24 kWords of data memory

Despite having all of the time critical parts "hand-coded" in Assembler, the Encoder implementation does not claim to be optimal especially in terms of memory usage when

compared to a code that is specifically designed for a chip implementation. Flexibility for the use in different applications is the major goal of the 56300 family code that originally started with MP3 and AAC. It covers an application range from portable Recorders and Players, over ISDN-Codecs, to source coding for satellite radio systems. Especially for all kinds of bidirectional communication, e.g. over analog and digital telephone lines, LD-AAC fulfills the requirements for a coding scheme that is able to transmit music and speech in high quality in real-time.

7 Conclusions

Although MPEG-4 Audio, due to its wide scope and comprehensiveness, poses new challenges to the implementers, real world implementations of some of the profiles have been successfully developed. The hardware requirements of the coding techniques for natural audio material are relatively modest and in the same range as those of other audio coding standards. Real-time implementations on DSPs and in software for general purpose computers have been created. The next step will now be the integration of the MPEG-4 Audio decoders into complete MPEG-4 systems.

References

[1] ISO/IEC JTC1/SC29/WG11 MPEG. International Standard IS 11172-3 Coding of moving pictures and associated audio for digital storage media at up to 1.5 Mbit/s, Part 3: Audio, 1991.

[2] ISO/IEC JTC1/SC29/WG11 MPEG. International Standard IS 13818-3 Information Technology - Generic Coding of Moving Pictures and Associated Audio, Part 3: Audio, 1994.

[3] ISO/IEC JTC1/SC29/WG11 MPEG. International Standard IS 13818-7 Information Technology - Generic Coding of Moving Pictures and Associated Audio, Part 7: MPEG-2 AAC, 1997.

[4] ISO/IEC JTC1/SC29/WG11 MPEG. International Standard IS 14496-3 Coding of audio-visual Objects, Part 3: Audio, 1999.

[5] ISO/IEC JTC1/SC29/WG11 MPEG. International Standard IS 14496-3 Coding of audio-visual Objects, Part 3: Audio Amendment1: Audio Extensions, 2000.

[6] ISO/IEC JTC1/SC29/WG11 MPEG. International Standard IS 14496-3 Coding of audio-visual Objects, Part 1: Systems, 1999.

[7] D. Schulz. Improving audio codecs by noise substitution. *Journal of the AES*, 44(7/8):593–598, July/August 1996.

[8] J. Herre and D. Schulz. Extending the mpeg-4 aac codec by perceptual noise substitution. In *104th AES Convention*, May 1998. preprint 4720.

[9] Mikko Suonio and Mauri Vaananen. A New Backward Predictor for MPEG Audio Coding. In *103. AES Convention, preprint 4521*, 1997.

[10] N. Iwakami and T. Moriya. Transform domain weighted interleave vector quantization (twinvq). In *101th AES Convention*, 1996. preprint 4377.

[11] Jürgen Herre, Eric Allamanche, Karlheinz Brandenburg, Martin Dietz, Bodo Teichmann, Bernhard Grill, Akio Jin, Takehiro Moriya, Naoki Iwakami, Takeshi Norimatsu, Mineo Tsushima, and Tomokazu Ishikawa. The integrated filterbank based scalable mpeg-4 audio coder. In *105. AES-Convention, preprint 4810*, 1998.

[12] Sung-Hee Park, Yeon-Bae Kim, Sang-Wook Kim, and Yang-Seock Seo. Multi-Layer Bit-Sliced Bit-Rate Scalable Audio Coding. In *103. AES-Convention, preprint 4520*, page September, 1997.

[13] B. Edler, H. Purnhagen, and C. Ferekidis. Concepts for hybrid audio coding schemes based onn parametric techniques. In *104th AES Convention*, 1998. preprint 4808.

[14] M.Nishiguchi, J. Matsumoto, S. Omori, and K. Iijima. MPEG95/0321 Technical Description of Sony IPC's proposal for MPEG-4 Audio and Speech Coding, November 1995.

[15] M.R. Schroeder and B.S. Atal. Code-excited linear prediction (celp): High-quality speech at very low bit rates. In *Proceedings of the ICASSP*, pages 25.1.–25.1.4, 1985.

[16] B.S. Atal and M.R. Schroeder. Stochastic coding of speech signals at very low bit rates. In *Proceedings International Conference Communications ICC84, part2*, pages 1610–1613, 1984.

[17] T. Nomura, M. Iwadare, M. Serizawa, and K. Ozawa. A bitrate and bandwidth scalable celp coder. In *Proceedings of the ICASSP*, 1998.

[18] B. Grill and K. Brandenburg. A Two- or Three-Stage Scalable Audio Coding System. In *99. AES Convention, preprint 4132*, 1995.

[19] B. Grill. A Bit Rate Scalable Perceptual Coder for MPEG-4 Audio. In *103. AES Convention, preprint 4620*, 1997.

[20] B. Grill and Bodo Teichmann. Scalable Joint Stereo Coding. In *105. AES Convention, preprint 4851*, 1998.

[21] ISO/IEC JTC1/SC29/WG11 MPEG. International Standard IS 14496-3 Coding of audio-visual Objects, Part 1: Systems Amendment1: Systems Extensions, 2000.

[22] Martin Dietz. Audio coding in digital broadcasting systems. In *Proceedings of the AES 17th International Conference*, pages 11–17, September 1999.

[23] ISO/IEC JTC1/SC29/WG11 MPEG. International Standard IS 13818-1 Information Technology - Generic Coding of Moving Pictures and Associated Audio, Part 1: Systems, 1994.

[24] "URL: http://doc.trolltech.com/aboutqt.html". "qt cross-platform c++ gui application framework".

[25] ISO/IEC JTC1/SC29/WG11 MPEG. Report on the MPEG-4 audio NADIB verification tests, July 1998. Document N2276 of the Dublin MPEG Meeting.

[26] ISO/IEC JTC1/SC29/WG11 MPEG. MPEG-4 Audio verification test results: Audio on Internet, October 1998. Document N2425 of the Atlantic City MPEG Meeting.

[27] Eric Allamanche, Ralph Geiger, Juergen Herre, and Thomas Sporer. Mpeg-4 low delay audio coding based on the aac codec. In *107th AES Convention*, Mai 1999.

[28] Johannes Hilpert, Marc Gayer, Manfred Lutzky, Thomas Hirt, Stefan Geyersberger, Josef Hoepfl, and Robert Weidner. Real-time implementation of the mpeg-4 low delay advanced audio coding algorithm on motorola dsp 56300. In *108th AES Convention*, February 2000.

[29] Johannes Hilpert, Michael Braun, Manfred Lutzky, Stefan Geyersberger, and Rainer Buchta. Implementing iso/mpeg-2 advanced audio coding in realtime on a fixed point dsp. In *105th AES Convention*, September 1998.

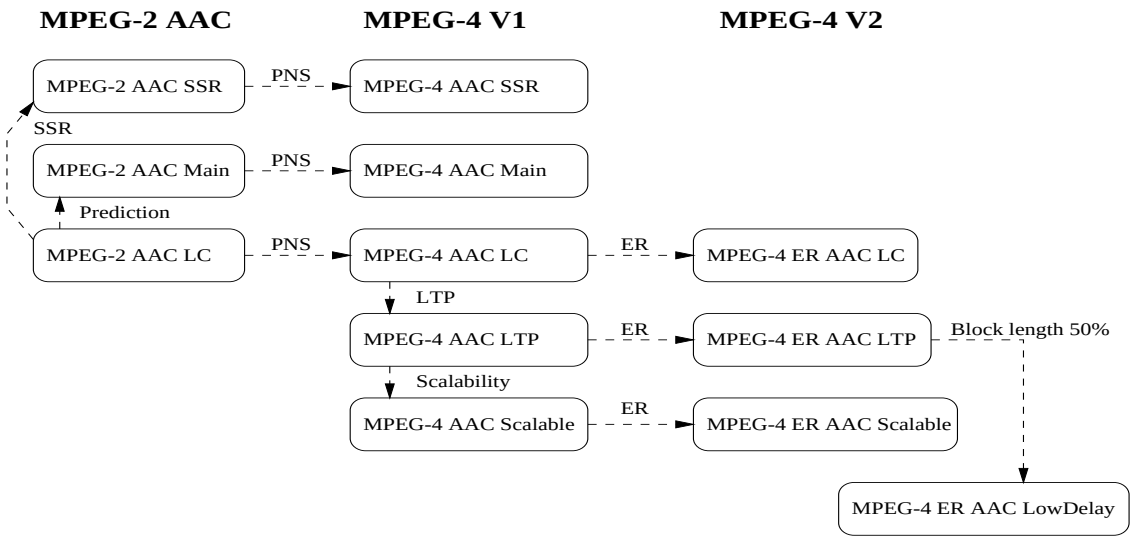

Figure 1: Overview of the AAC Coder Family

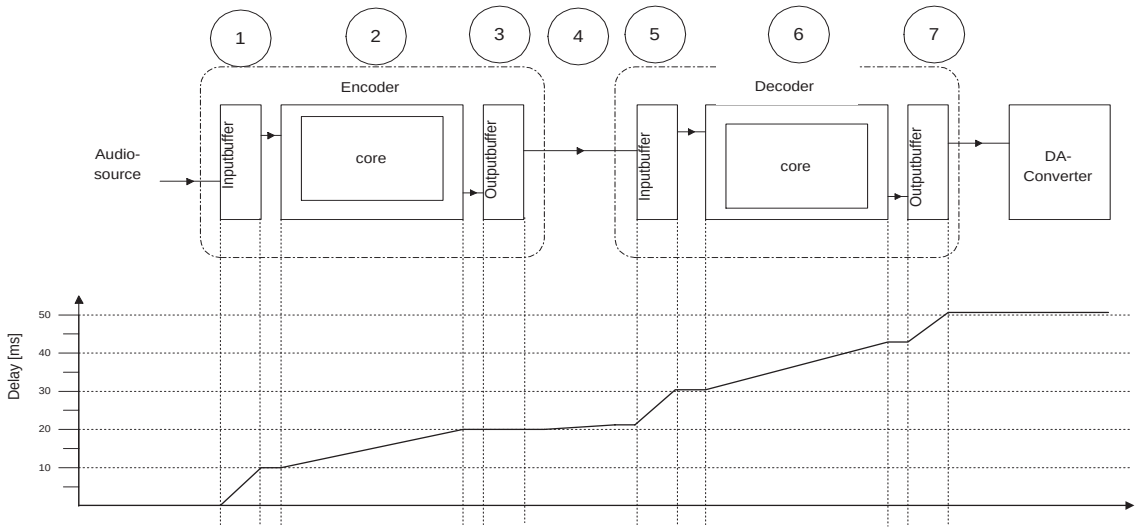

Figure 2: Delay in a LD-AAC Codec Chain