I  INTRODUCTION

This paper presents the implementation of the AC-3 Digital Audio Compression Standard, issued by Advanced Television Systems Committee (ATSC) on the MAS-like DSP platform. In Chapter 2 are described the basic principles of the Standard and the main features of the used platform, whilst Chapter 3 contains the details of the implementation itself.

II  AC-3 STANDARD

This Chapter briefly presents the basic principles of the AC-3 Standard as well as the main features of the used platform. The AC-3 Standard fully defines the AC-3 audio decoder, while the AC-3 audio encoder is defined to be productive of the such bit stream that the standardized decoder would correctly decode [1].

The motivation for this Standard lies under the increasing need for more efficient broadcast of recorded audio signals. It must be noted that the term Compression in this Standard is used to describe the compression of the amount of digital information which must be stored or recorded, and not the compression of dynamic range of the audio signal. The AC-3 digital compression algorithm can encode from 1 to 5.1 channels of source audio from a PCM representation into a serial bit stream at data rates ranging from 32kbps to 640 kbps. The 0.1 channel refers to a fractional bandwidth channel intended to convey only low frequency (subwoofer) signals. Typical applications are in satellite and terrestrial audio broadcasting, delivery of audio over metallic or optical cables and the storage of audio on magnetic, optical, semiconductor and other storage media [1][3].

The process of decoding is shown in Figure 1. Upon synchronization to the encoded bit stream, error checking, and de-formatting the various types of data such as the encoded spectral envelope and the quantized mantissas, the bit allocation routine is run and the results are used to unpack and dequantize the mantissas. The spectral envelope is decoded to produce the exponents. The exponents and mantissas are transformed back into the time domain to produce the decoded PCM time samples [1].

4. The synthesis filterbank resolution must be dynamically altered in the same manner as the encoder analysis filter bank had been during the encoding process [1].

Bit stream of the AC-3 encoded audio data is a sequential set of frames referred to as synchronization frames. Each synchronization frame consists of 6 encoded audio blocks, each representing 256 samples per audio channel (1536 samples per synchronization frame for each audio channel). At the beginning of each frame the 16-bit synchronization word is placed, followed by the CRC word and information about the bit stream. Six audio blocks are succeeding. The frame ends with the second CRC word. The AC-3 synchronization frame is presented in Figure 2. [1]

The AC-3 decoding process flow diagram is presented in Figure 3. The Standard suggests the Figure 3 be one example of the decoder, and recognizes the possibilities of various improvements in certain areas (number of instructions, memory requirement, number of transforms required, etc.) [1]. The input bit stream will typically come from a transmission or storage system. The interface is not specified in the Standard [1]. The AC-3 format allows rapid synchronization. The 16 bit Synchronization word has a low probability of false detection. When a synchronization pattern is detected the decoder may be estimated to be in synchronization and one of
the two CRC words may be checked. First CRC word covers the first 5/8 of the frame, therefore the CRC may be checked after only 5/8 of the frame has been received. Or, the entire frame size can be received and the second CRC word checked. If either CRC checks, the decoder may safely be presumed to be in synchronization and decoding and reproduction of audio may proceed. The chance of false synchronization in this case would be the concatenation of the probabilities of false synchronization word detection and a CRC misdetection of error [1].

Inherent to the decoding process is the unpacking (de-multiplexing) of the various types of information included in the bit stream. The Standard suggests these items be copied from the input buffer to dedicated registers or specific working memory location, and some of the items may simply be located in the input buffer with pointers to them saved to another location for use when the information is required [1]. The exponents are delivered in the bit stream in an encoded form. In order to unpack and decode the exponents two types of side information are required. First, the number of exponents must be known. For full bandwidth channels this may be determined from either channel bandwidth code for each channel (uncoupled ones) or from coupled begin frequency (coupled channels). For the coupling channel, the number of exponents may be determined from coupling begin and end frequencies. For the Low Frequency Effects channel (when present) there are always seven exponents. Second, the exponent strategy in use (D15, etc.) is indicated by the corresponding bit allocation pointer. In order to decode the mantissa data more efficiently, some mantissas are grouped together into a single transmitted value [1]. The decoder, by default, shall use this value to alter the magnitude of the coefficient (mantissa) [1].

When coupling is in use, the channels that are coupled must be decoupled. Decoupling involves reconstructing the high frequency section (exponents and mantissas) of each coupled channel, from the common coupling channel and the coupling coordinates for the individual channel. Within each coupling band, the coupling channel coefficients (exponent and mantissa) are multiplied by the individual channel coupling coordinates [1].

In the 2/0 audio coding mode rematrixing may be employed, as indicated by the rematrix flags. When the flag indicates a band is rematrixed, the coefficients encoded in the bit stream are sum and difference values instead of left and right values. For each block of audio a dynamic range control value may be included in the bit stream. The decoder, by default, shall use this value to alter the magnitude of the coefficient (exponent and mantissa) [1]. The decoding steps described above will result in a set of frequency coefficients for each encoded channel. The inverse transform converts the blocks of frequency coefficients into blocks of time samples [1]. The individual blocks of time samples must be windowed, and adjacent blocks must be overlapped and added together in order to reconstruct the final continuous time output PCM audio signal [1].

If the number of channels required at the decoder output is smaller than the number of channels that are encoded in the bit stream, then downmixing is required. Downmixing can be performed in the time or the frequency domain, since the inverse transform is a linear operation [1].

Typical decoder will provide PCM output samples at the PCM sampling rate. Since blocks of samples result from the decoding process, an output buffer is typically required. The Standard does not specify nor describe output buffering [1]. The output PCM samples may be delivered in form suitable for interconnection to a digital to analog converter (DAC), or in any other form. The Standard does not specify the output PCM format [1].

The platform used for the implementation is based on Micronas Intermetall’s digital signal processor (DSP) MASC 3500 [4]. Since the original MAS DSP does not possess enough on-chip ROM and RAM for the implementation to be evaluated, the design of the assembly language software was based on the same platform simulator, with the full address space option. This modified platform is non-existing one, but the idea is to design the implementation specific chip with full reference to the developed algorithm, based on the existing platform.
III IMPLEMENTATION

This Chapter presents the structure of the developed software as well as the analysis of required resources. The structure of the implemented algorithm is shown in Figure 4.

Since the Referent C code [2] was developed on the 16-bit word basis, and double precision floating point number presentation, some significant modifications had to be performed in order to implement the algorithm on the 20-bit fix point DSP. Also, optimization of time and memory resources had to be introduced.

The input buffer is designed to operate in burst mode, while the output buffer operates in continuous DMA mode. The core of the algorithm is the unpacking routine. It is realized to extract the given number of bits from the input bit stream. The input bit stream is a set of 16-bit words received as a set of 20-bit MASC words. Therefore, special precautions were used to make this routine optimal and accurate.

First, the synchronization word, first CRC word, sampling rate and frame size codes are extracted. Next step is the first CRC calculation for the first 5/8 of the frame. If the CRC check approves the validity of the frame, the unpacking of encoded audio can be performed.

Upon initialization of the unpacking process, the extraction of the bit stream information data is executed. Those data encompass the following parameters: bit stream identification, bit stream mode, audio coding mode, center and surround mixing levels (if any), Dolby surround mode (if any), low frequency effects channel presence, dialog normalization, compression factors (if any), language codes (if any), mixing levels and room types (if any), copyright bit, original bit stream bit, time codes (if any) and the number of additional bit stream information. The following is the extraction process of the AC-3 fixed data. Those data are: block switch flags, dither flags, dynamic range control fields, coupling strategy fields, coupling coordinates fields, rematrixing operation parameter, exponent strategy fields, exponent fields, bit allocation strategy fields and delta bit allocation fields. Those data are needed for the control of unpacking process for each audio channel.

Since the Standard supports encoding of 5.1 channels, and the user can choose one from several modes of reproduction, the downmixing process is also defined. This process is executed in three steps. First, the downmix table is formed from a load of received parameters. This table is 6x6 matrix from which are the scaling parameters taken in the process of downmixing. Upon extraction and back transformation of each audio channel, the downmixing is executed. When all six (or less) audio channels are extracted and downmixed, the unused downmix buffers are cleared, the extracted and decoded audio samples are windowed and output.

The extraction and decoding of audio channel is executed for each of the present encoded audio channels ones per audio block. The process consists of extracting and delta expanding the exponents and the extracting of the mantissas. There are several possible modes for these processes. Upon the reconstruction of actual spectral coefficients, rematrixing (if needed) and inverse transform is executed. The result is a set of PCM samples for the processed channel.

This algorithm is designed in less than 6000 instructions. The necessary coefficient tables occupies 2034 words of data ROM. The critical resource for this implementation is data.

**Figure 4.** The structure of the implemented algorithm
RAM. The developed implementation requires less than 7500 data RAM words. This number is a result of reusing and dynamic allocation of several data memory areas. The program is executed in less than 50 MIPS.

IV CONCLUSION

In this paper is presented the implementation of the AC-3 Digital Audio Compression Standard on the MAS-like DSP platform. The implementation is assembly language program code that is executed on MAS-like digital signal processor simulator platform verified by the test vectors issued by Dolby Laboratories [2]. Optimization of the time and memory requirements for the designed implementation as well as the analysis of the necessary resources are also presented. Although the design is based on the Referent C code written by Dolby Laboratories [2], it also introduces some significant improvements and optimization required for the described MAS-like DSP implementation.

REFERENCES


Summary: The main features of the AC-3 Digital Audio Compression Standard and the MAS-like platform are described. The detailed structure of the developed software is presented with the description of each step of the algorithm. Finally, the evaluation of necessary resources is analyzed.

APPROACH TO IMPLEMENTATION OF THE AC-3
AUDIO DECODER ON THE MAS-LIKE DSP
PLATFORM

Domazetovic A., Lukac Z., Samardzija D.