

DESIGN OF A NOVEL SYNTHESIS FILTER FOR REAL-TIME MPEG-2 AUDIO DECODER IMPLEMENTATION ON A DSP CHIP

Won-Kyu Paik and Sun-Young Hwang

CAD & Computer Systems Lab, Dept. of Electronic Engineering

Sogang University, C.P.O. Box 1142, Seoul, 100-611, Korea

E-mail: hwang@ccs.sogang.ac.kr

Abstract – This paper describes the design of a novel synthesis filter for real-time implementation of MPEG-2 audio decoder on a single, programmable DSP chip. In the first phase of our design we propose a fast subband synthesis filter bank employing a decomposed IDCT block with reduced multiplication stages, shared by both MPEG-1 and MPEG-2 decoders. In the second phase of the design, a parameter analyzer block is adopted to take advantage of the multichannel computing structure in MPEG-2 thereby further reducing the number of MAC operations. In the final phase, we propose a low-cost decoder employing unorthodox methods of boosting performance by gracefully degrading quality of service.

The overall runtime of the decoder has been reduced by 60% when compared to the modified ISO decoder employing a synthesis filter bank with fast IDCT block. Real-time performance validations have been made by re-tuning time-critical code sections for both native VLIW and DSP architectures. The final implementation successfully passes all ISO/IEC MPEG-2 audio compliance tests and achieves near-transparent audio decoding and signal reconstruction on the fly.

1. Introduction

The demand for reductions in data rate for wideband digital audio signal transmission and storage in multimedia applications substantiates the need for efficient compression techniques. A serial bitstream of CD quality audio material consists of

44,100 16-bit PCM samples per channel which corresponds to 1.4 Mbps in stereo [1]. Audio at that bitrate contains a lot of redundancies and this led to the development of perceptually coded, lossy compression techniques to eliminate information that is irrelevant from a human sound-perception viewpoint. Studies have shown that by masking inaudible components of audio signals via psychoacoustic models [1][2], perceptual audio systems are able to achieve significant compression while maintaining perceived audio quality roughly equal to the original.

The International Organization for Standardization (ISO) has established a working group known as the Motion Picture Experts Group (MPEG), to develop standards for coding audio-visual presentation materials aimed at digital storage media such as CD-ROMs, not exceeding a combined continuous transfer rate of 1.5 Mbps. The audio portion specifies the coded representation of high-quality audio signals intended for stand-alone or multiplexed-with-video applications [3][4]. The MPEG audio system features a wide mix of channel configurations, multiple sampling and data rates, layer compatibility, decoder version scalability, dynamic range control, Dolby Pro Logic compatibility, and multi-lingual support, thereby addressing to a variety of consumer and professional applications. Application areas run a wide gamut from DVD players, multichannel High Definition Television (HDTV), Integrated Service Digital Network (ISDN) transmission codecs, cable and satellite-based TV (e. g. DVB, USSB) services, Digital Audio Broadcasting (DAB), studio recording and editing, to multimedia/Internet audio.

Contributed Paper

Manuscript received July 7, 1999

0098-3063/99 \$10.00 © 1999 IEEE

MPEG-2 was first developed with the broadcast application in mind. The concept of a "1-encoder to N-decoders," makes for non-symmetric systems with a relatively complex, high-quality encoder on one end and fast, low-complexity, low-cost decoders at the other delivering quality service on limited bandwidth. Hence, some researches leading to the real-time implementation of MPEG standards have based the decoder designs on programmable DSPs [5][6], for their merits in easy implementation of complex algorithms, reduced design cycles, and development cost. A floating-point implementation of a real-time MPEG-1 system was presented by Murphy [5]. But limitations to processing speed and memory of the host DSP chip prohibited the implementation of a fully functional MPEG-1 system on a single DSP, let alone multichannel systems. Similar work has been found implementing the MPEG-2 standard on a fixed-point, multimedia DSP [6], indicating a move toward low-cost, uniprocessor solutions. Fixed-point implementations are cost-effective, but complications may be introduced by the repetitive computational tasks typically populated in signal processing applications. Rounding-off error introduced by multiplying two fixed-point data are trivial as is, but the propagation of these smaller errors onto successive multiplication stages ultimately corrupts reconstructed signal quality to a noticeable level. The synthesis filter bank as defined by the standard, is considered by far the most computationally intensive block in the entire decoding system well populated with MAC operations, thereby efforts to reduce the complexity of the synthesis filter has been essential for real-time decoder implementations. The most common practice in reducing the number of multiplications and additions, has been by exploiting the structural properties of the IDCT block resulting in manifold decoder performance improvements [7]. There have been other fast IDCT schemes that are effective for designing speedy synthesis filter blocks [8]. However, not all IDCT schemes are suitable for fixed-point implementation, since a trade-off between the number of multiplication operations and rounding errors due to splitting the internal IDCTs exist. To achieve fast decoding of high quality audio, it is essential to keep the numerical errors to an acceptable level without running into hard real-time constraints of the decoding system. Further improvements to the synthesis filter design can be achieved by exploiting the multichannel computation structure of MPEG-2, thus contributing for a faster MPEG-2 dedicated

decoder operation. This paper fully exploits the multichannel operation of the MPEG-2 decoding system to reduce the number of MAC operations.

In this paper, we present the design and implementation of a real-time MPEG-2 audio decoder with an improved synthesis filter on a single, fixed-point DSP. The proposed system employs a decomposed IDCT block with reduced multiplication stages, and a parameter analyzer block for dedicated MPEG-2 decoding. The paper is organized as follows: Section II provides an overview into the MPEG-2 multichannel audio decoding process. In section III, we propose a versatile subband synthesis filter routine employing a decomposed IDCT block with reduced multiplication stages, followed by optimizations for dedicated multichannel computation structure. For cost-effective measures, we propose unorthodox methods of gracefully degrading audio signal quality for faster decoder speed. The enhanced decoder is realized on the DSP board, mounted on a host PC. In section IV, experimental results are reported, and a conclusion is drawn at the final section of the paper.

II. MPEG-2 Multichannel Audio Coding System

The MPEG audio standards, ISO/IEC 11172-3 (MPEG-1) [3] and ISO/IEC 138180-3 (MPEG-2) [4] define three layers of increasing complexity and subjective performance varying with compression efficiency. With successive layers, however, the coding/decoding system is subject to increased delay and sensitivity to transmission errors.

2.1 Overview of the MPEG Audio Coding System

The main decoding sub-blocks of the MPEG audio decoder are comprised of a frame unpacker, dequantizer, dematrixing block (MPEG-2), and a synthesis filter bank. After frame disassembly from the input encoded bitstream, the decoder passes the bit-allocation, scalefactor side-informations and audio samples to the inverse quantizer for restoring the quantized subband values. These subband samples are fed into the synthesis filter block for reconstruction and output of PCM samples on a per channel basis.

2.2 Multichannel Extension

Restrictions to two-channel stereophony imposed by the MPEG-1 standard are lifted with multichannel sound defined by ISO/IEC 138180-3. The extension to the multichannel system provides support for 5.1 channels without simulcast intended for broadcast complying with the HDTV format. The MPEG-2 decoding system shares the same audio algorithm core as MPEG-1, hence providing forward and backward compatibility [1]. An MPEG-2 decoder will accept an MPEG-1 formatted bitstream and an MPEG-1 decoder can derive a stereo representation from an MPEG-2 bitstream. The multichannel format uses an encoder matrix that allows a two-channel decoder to decode stereo compatible signals that is a subset of a multichannel bitstream. The multiple channels are matrixed to form compatible MPEG-1 left/right channels, as well as other MPEG-2 channels, as shown in Figure 1. These are encoded into MPEG-1 frames, and additional multichannel data is placed in the extended auxiliary data field. MPEG-2 features intensity coding, crosstalk reduction, inter-channel prediction coding, and center channel phantom image coding to achieve a combined bit rate of 384 kbps, using Layer II at a 48-kHz sampling frequency.

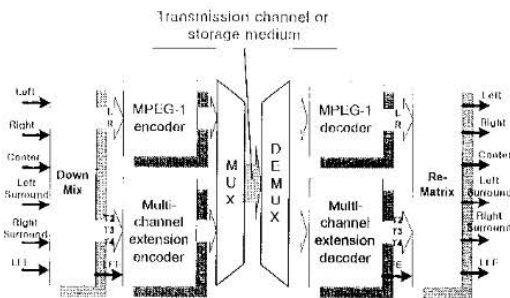


Fig. 1. Forward and backward compatible MPEG-2 audio system.

2.3 Synthesis Filter Bank

The subband synthesis filter bank is a polyphase frequency-time domain transform engine, shared by both MPEG-1 and MPEG-2 algorithms. After the dequantization process from the previous stage, 32 subband samples per channel are input to the

synthesis filter bank and 32 reconstructed time-domain PCM samples are output to a D/A converter for reproduction. As shown in Figure 2, the synthesis filter bank is defined by the MPEG standard with three key operations: Initialization, Matrix transformation, and Window/overlap-add. Initialization involves shifting of 64 samples, from a queue of 1024 points at a time. The matrix transform operation receives the 32 frequency-domain samples and returns 64 intermediate elements, which are written to the input IQMF buffer. The buffer has 1024 elements for every channel and is cyclic.

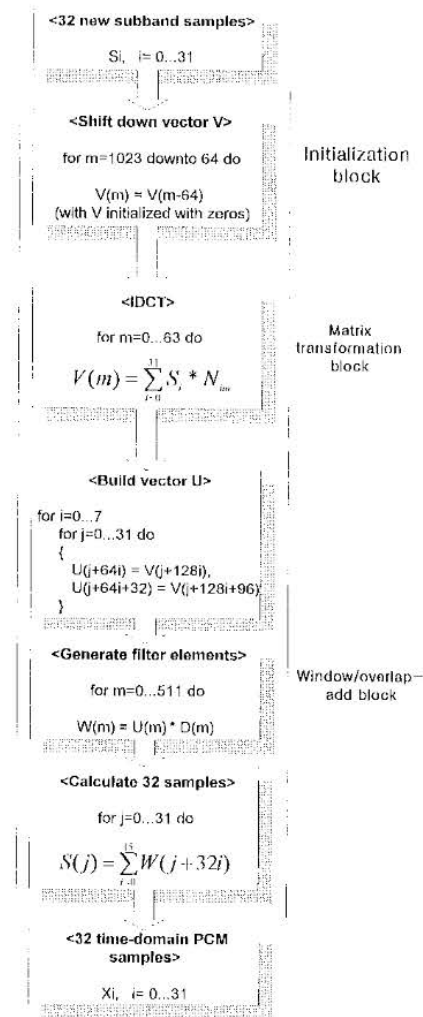


Fig. 2. Synthesis filter bank flow [3].

The window/overlap-add block performs the actual filtering operations. The filtering operation is performed on 512 elements that includes the last 64. A new vector is built from these elements and is multiplied with the tabulated windowing coefficients in the IQMF block to produce 32 time-domain samples. The 32 samples are formatted into PCM samples for output. In the next iteration, all the elements of V are shifted down 64 places in preparation for the next IDCT output. These operations are performed recursively for each represented channel.

Table 1 displays the profiling output of the ISO MPEG-2 decoder software implementation on UNIX and DSP platforms, respectively. The profile result is an average taken from 20 consecutive frames of a random mix of multichannel audio clips, including the ISO/MPEG-2 conformance test streams. This table indicates that the synthesis filter bank is the most computationally intensive block in the entire decoder, therefore any design improvements made here would yield a faster decoder. For MPEG-2 multichannel operation, the speedup is more pronounced since the synthesis filter iterations increase in direct proportion to the number of represented channels.

Table 1. Profiling of the ISO MPEG-2 decoder functions.

Decoding Stage	Percentage of total time spent in each decoding task	
	UNIX	DSP
Synthesis filter	90%	83%
Dequantizer	4%	7%
Stream decoder	2%	4%
Dematrix & Denormalizer	3%	4%
Others	1%	2%

The most DSP cycle-hungry block in Figure 2 of the synthesis filter bank is the matrix transformation block. As shown in Table 2, any improvements to the synthesis filter bank centers around a more efficient IDCT block, requiring optimization of nested loops

and reductions in number of multiplications in the loop kernels.

Table 2. Profiling of the ISO synthesis filter sub-blocks.

Synthesis filter bank sub-blocks	# multipli-cations	# addi-tions	# DSP cycles	Percentage of total cycles consumed
IDCT	2,048	1,984	410k	91%
Window/overlap-Add	512	480	32k	7%
Others	0	0	6k	2%

III. The Proposed Synthesis Filter

As presented in the previous section, optimizations to the synthesis filter are essential for a fast realization of the MPEG decoder. In this section, we first propose a subband synthesis filter bank employing decomposed IDCT and windowing blocks, shared by both MPEG-1 and MPEG-2 decoders. Next, we propose the addition of a parameter analyzer block for further performance improvements to MPEG-2 decoders, taking advantage of the multichannel computing structure. Finally we propose a synthesis filter employing graceful degradation techniques targeting low-cost implementations with marginally degraded audio output quality.

Figure 3 shows the proposed synthesis filter block shared by both MPEG-1 and MPEG-2 multichannel decoders. It implements a modified IDCT block with a multistage butterfly network at the input followed by decomposed IDCT and efficient windowing functions. Matched against other well-known 32-point IDCTs [8] the IDCT scheme employed here profits from minimal fixed-point arithmetic propagation errors while reducing MAC counts.

3.1 Design of a Fast MPEG-1/2 Synthesis Filter Bank

In this section we propose a fast subband synthesis filter bank employing a decomposed IDCT block with reduced multiplication stages, shared by both MPEG-1 and MPEG-2 decoders. The IDCT scheme employed in the proposed design recursively decomposes the IDCT block into smaller matrices by exploiting the symmetric properties of the cosine transform. An efficient windowing scheme has been proposed that exploits the symmetries observed in the IDCT output.

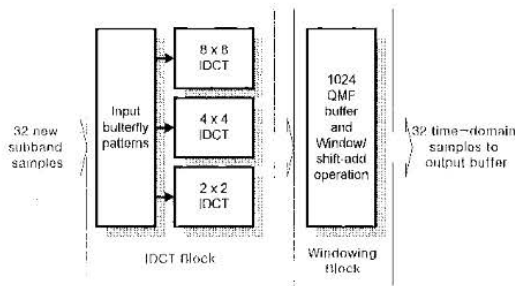


Fig. 3. Block diagram of the proposed synthesis filter.

3.1.1 IDCT Block with Decomposed Matrix Transformations

Symmetric Properties of the Cosine Transform: The IDCT transforms the input subband samples from frequency to time-domain by using Equation (1) as defined in the MPEG standard.

$$V(i) = \sum_{sb=0}^{31} N(i, sb) X(sb) \\ = \sum_{sb=0}^{31} \cos \left[\frac{\pi}{64} (2sb+1)(i+16) \right] X(sb), \quad i = 0 \dots 63 \quad (1)$$

Vector V denotes an array of 64-intermediate values needed at the window/overlap-add block, the discrete cosine terms $N(i, sb)$, and subband coefficients $X(sb)$. To evaluate V as is, 2,048 multiplications and 1,984 additions are required as shown in Table 2. However, V can be built exploiting the inherent symmetries of $N(i, sb)$ as shown in

Equation (2), thereby using only half the multiplications and additions required in Equation (1).

$$V(i) = -V(32-i) \\ V(32+i) = V(64-i), \quad i = 1 \dots 16 \quad (2)$$

Butterfly Stage: The 32-point IDCT derived from Equation (2) can be further decomposed into two 16×16 even and odd matrices by way of shuffling and grouping $X(sb)$ into butterfly patterns, denoted in Equation (3). Thus, an N -point IDCT of Equation (2) can be decomposed into sums of two $N/2$ -point IDCTs, obtaining a four-fold reduction in the number of MAC operations. As shown in Figure 3, the first input butterfly pattern of $X(sb)$, can be further decomposed in a similar manner by exploiting trigonometric and structural properties, thereby reducing the complexity of the IDCT block of the synthesis filter.

$$V(i) = \sum_{sb=0}^{16} N(i, sb) [X(sb) + X(31-sb)], \quad i = 0, 2, \dots, 30 \\ V(i) = \sum_{sb=0}^{16} N(i, sb) [X(sb) - X(31-sb)], \quad i = 1, 3, \dots, 31 \quad (3)$$

Successive Decompositions: By recursively breaking down the input even 16×16 butterfly pattern into 8×8 , 4×4 , and 2×2 parts, the IDCT can be further decomposed into smaller transform matrices. However, decomposing any odd butterfly pattern requires performing multistage multiplication with non-unity trigonometric multipliers, which are deleterious for fixed-point arithmetic. It is imperative to keep the propagating rounding errors to a minimum, hence multiplication is performed at the end of the butterfly decomposition stage and any splitting of internal IDCT that require intermediate non-unity multiplications are omitted. The generation of butterfly patterns requires only addition and subtractions but no multiplications. Once the butterfly patterns are completely broken down, matrix transformation is performed.

3.1.2 Windowing Block

A 511th-order IQMIF filter bank with a 96-dB sidelobe rejection and a steep pass-to-stop-band transition is used to perform the window/overlap-add block upon the 1024 intermediate element output from the IDCT block. A new vector is built from these elements and is multiplied with the tabulated windowing coefficients $D(m)$ to generate the 512-window elements. A 16-point filtering operation is performed on the generated filter elements to produce 32 time-domain samples $S(i)$. Each $S(i)$ are formatted into PCM audio samples, and output to the channel FIFO buffers for DAC.

$$\begin{aligned} U(j+64i) &= V(j+128i) \\ U(j+64i+32) &= V(j+128i+96), \\ j &= 0 \dots 31, \quad i = 0 \dots 7 \\ W(m) &= U(m)D(m), \quad m = 0 \dots 511 \\ S(j) &= \sum_{i=0}^{15} W(j+32i), \quad j = 0 \dots 31 \end{aligned} \quad (4)$$

Equation (4) holds when the inverse transform given in Equation (1) is re-defined, by exploiting the symmetrical properties of the windowing coefficients D . The odd and even input butterflies derived from Equation (3) for the calculation of V .

The 512 coefficients are split into 16 blocks of 32 V coefficient, the even numbered blocks are multiplied with the first 32 IDCT outputs, and the odd blocks are multiplied with the next 32 IDCT outputs. The windowed results of 16 blocks are then overlapped and added, resulting in half the multiplication and additions of the original.

3.2 Optimizations Exploiting MPEG-2 Computing Structure

As stated earlier, MPEG-2 provides full MPEG-1 support and features key multichannel operations such as dynamic crosstalk reduction, inter-channel prediction coding, and center channel phantom image coding. In this section, a parameter analyzer block is adopted to take advantage of the multichannel computing structure thereby achieving further performance improvements. With the availability of

dynamic bit-allocation information and multichannel coding parameters after frame parsing from the encoded bitstream, the synthesis filter loop can be optimized to exploit the redundancies and zero-allocation profiles of subband samples, resulting in the elimination of wasted MAC operations.

3.2.1 Exploiting Intra, Inter-channel Redundancies

The MPEG-2 multichannel decoder exploits the use of information from neighboring channels in order to achieve optimal data reductions without resorting to simulcast operations. One of the main features supported by the MPEG-2 decoding system is dynamic crosstalk reduction, which reduces the amount of data through copying stereo-irrelevant signal components for extension channel reconstruction [4]. With the existing ISO MPEG decoders, rescaling, dequantizing, dematrixing and denormalizing operations precede the synthesis filter bank operation. However, the proposed decoder performs the synthesis filtering prior to other operations, thereby if dynamic crosstalk reduction is used, the application of the synthesis filter bank to extension channels is bypassed altogether. Thus inter-channel redundancies can be exploited at the decoder level, eliminating unnecessary iterations of costly IDCT and windowing operations resulting in savings of the number of MAC operations.

In a similar context, intra-channel redundancies are exploited using a temporary buffer for storage of intermediate V s. If a match is found in the next iteration of the synthesis filter, the IDCT operation is omitted, permitting us to reuse older V coefficients for windowing. A match can be found by comparing the values of subband bit allocation and corresponding sample coefficients. The intra-channel technique exploits the temporal locality of periodic audio signals.

3.2.2 IDCT by Subband Coefficient Analysis

An examination into subband coefficient profiles for various audio streams indicate that not all subband coefficients are allocated with bits. For each zero subband coefficient in an N -point DCT, N multiplications and $N-1$ additions are saved. Skipping the DCT matrix operation for zero subband coefficients permits us to reuse older V vector

coefficients for windowing. Statistically then, about 50% of the 2048 multiplies and 1984 additions required in the original DCT implementation are eliminated. Empirical analysis results dictate to an average of 80% reduction in multiplies and 70% adds per 10 frame samples. These averages were taken over a wide range of diverse of audio clips, including jazz, classical, vocal pieces. A combination of DCT optimizations exploiting symmetric properties and subband coefficient analysis yields to empirical results of 90% reduction of multiplies and 80% reduction of additions compared to the DCT performance of the ISO/MPEG decoder.

Figure 4 shows the bit allocation profile across the spectral subbands of an audio frame fragment. If no bits were allocated for a particular subband index, any attempts at applying the matrix transform operation would only result in a waste of CPU cycles. For efficiency, zero bit allocation samples are dropped from the IDCT operation, thereby relieving the synthesis filter bank, with no performance compromises. Benefits are added by applying bit-allocation analysis to surround sound channels that have spares bit-allocation profiles, or to the bandwidth-limited phantom-coded center channel.

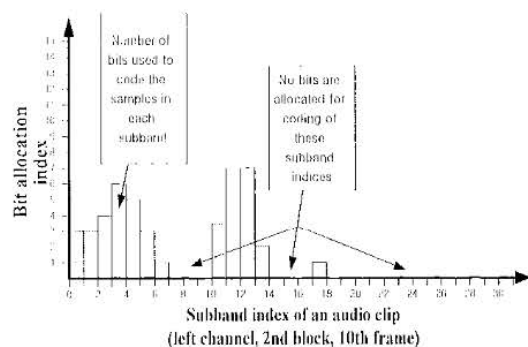


Fig. 4. Bit allocation across subbands.

Our statistical analysis of subband coefficients indicates that close to 50% of the subband coefficients are not allocated across a diverse selection of musical scores and compliance bitstreams. These are the average values for 10 frames of various bit and sampling rate configurations. Empirical results show that the upper

half of the subband indices have a higher probability of zeroing out.

3.3 Graceful Degradation

Providing uninterrupted audio at lower-cost requires accurate real-time decoder system complexity measure such that efficient resource allocation can be attained. However, the measure of processing power for real-time operation of a software-driven decoder remains a difficult task. We assume that the absolute measure of decoder complexity can be determined by the total decoding time with respect to the bitstream or in more abstract measures, elementary operations such as the IDCT / windowing operations.

For each audio frame, the sub-band filter is repeated 12 or 36 times per channels depending on the Layer. Since the IDCT and windowing blocks are executed with every iteration of the synthesis filter, they form the kernels that lie on the "critical path" of the decoding process. Apart from scaling down the multichannel stream (from a 3/2 configuration to 2/1 for example) a more fine-grained approach of gracefully degrading quality of audio service is proposed in this paper.

We devised two ways of gracefully degrading the audio output: Partial zeroing of the $D(m)$ window coefficients, and eliminating the upper half subband coefficients of the spectral image. As shown in the latter half of the window/overlap-add block of Figure 2, 32 time-domain samples $S(sh)$ are calculated by Equation (4) with the generated 512 $W(m)$. However, about 40% of the values of $D(m)$ are close to being zero, hence the 16-point windowing operation can be reduced. It is also observed that most of the upper half of the subband samples are zero. Hence, IDCT can be performed for only 16 subband samples. The combined effort provides for minimal quality of service degradation for speedy decoding.

3.4 DSP Implementation

The proposed MPEG-2 decoder designs were implemented on a high-performance DSP with a VLIW architecture. To maximize real-time performance, platform dependent and independent optimizations were performed on the critical code segments. Timing of real-time communication of the I/O subsystem, between host and the DSP chip and

control of the output channel buffers, is essential to ensure continuous I/O operations.

3.4.1 DSP Architecture

The proposed MPEG-2 audio decoder has been implemented and tested on a commercial DSP development board, mounted on a host PC [9]. The processor has a 200 MHz, 16-bit fixed-point VLIW core, claimed to run at 1,600 MIPS. It implements a load-store architecture and manifests two cross-linked datapaths, each having four execution units, a split 4-Mbyte on-chip program and data memories, a uniform general-purpose register file, operating on 32/40-bit operands. The eight execution units include two multipliers, four ALUs, two shift/logical units. Hence, the processor can execute up to eight orthogonal 32-bit RISC-like instructions in a single cycle, enabling it to achieve the high level of parallelism. Salient DSP features such as saturation logic and circular addressing are supported. The development board is equipped with an array of peripherals such as 8 Mbytes of global memory, Host Port Interface (HPI), external JTAG scan path emulation support, two-buffered serial ports, 16-bit stereo audio interface supporting standard sample rates of up to 48 kHz.

Implementing real-time multimedia standards such as MPEG audio on programmable DSPs requires code adaptations to the native hardware. Despite the ease in translating the MPEG standard by a high-level language such as C, the efficient use of the processing power of the VLIW machine remains a difficult task. For maximal utilization of on-chip hardware resources, we have adopted both DSP and VLIW architectural optimizations to the MPEG algorithm. VLIW architecture dependent and independent optimization techniques such as software pipelining, loop unrolling and other DSP intrinsics [9] have been employed to resolve time-critical loop blocks of the decoder.

3.4.2 Real-time Communications

Timing of real-time operations ensures correct synchronous operation: all data is produced and consumed at exactly the rate that it is output or input. Decompression of audio signals are characterized by an amplified data output that far exceeds the consumed input of encoded data, creating a

processing core that far exceeds that of the I/O subsystem. The hardware architecture is designed to connect directly to a PCI bus, as a PCI bus master, and to enable efficient use of the PCI bus for real-time communication of data between the DSP and the host. These built-in features enable efficient software to be produced for processing the streaming audio data. The software does not need to be aware of the input data format, as it will be converted to the standard 16 bit stereo format without intermediate modifications. In addition, the software need not be aware of the origin of the data. The DMA engine automatically places the input data into a circular buffer. When data has been consumed from the buffer the hardware automatically create the DMA request and queue it for filling. A similar process is used to empty out buffers to the DAC.

Once the data is available upon the completion of a synthesis filter iteration, the reconstructed audio PCM data are rounded to the DAC word width and copied to the corresponding output channel FIFO buffer. When buffers are full for all the specified channels, audio data are output through the DAC. In contrast with the input buffer stage, the decoder output processing uses very few MIPS but a relatively large amount of memory. For each output channel, the decoder requires 1152-long PCM sample buffers.

IV. Experimental Results

The performance of the proposed MPEG algorithm has been validated on a UNIX and implemented on the DSP chip, using MPEG-2 compliance test stream sequences. Tables 3 and 4 show the experimental results for the proposed decoder designs, compared with the performance of the ISO decoder and a modified ISO decoder with the synthesis filter employing a fast IDCT [7].

In Table 3, the total number of multiplications and additions required in the proposed synthesis filter has been reduced to an average of 60% when compared to the modified ISO synthesis filter sub-blocks. The average execution time information of the proposed DSP-based decoder is reported in Table 4, for MPEG-1 and MPEG-2 multichannel bitstreams. Note that the direct implementation of the ISO decoder on the DSP chip does not provide for successful real-time performance, as portrayed by the Jazz and Classical clips. The proposed decoder

achieves an average of 60% speedup over the modified ISO decoder.

The results of Table 4 have direct implication upon deciding the lower bound for DSP platforms for which decoder systems can be implemented. Albeit the modified ISO decoder meets the current real-time specifications, it is certainly not a cost-effective implementation since the hardware on which it was implemented is one of industry's high-end DSP cores available today. Consequently, a faster decoder design opens the possibility for a more cost-effective implementation. Since real-time environments put a premium on the availability of hardware resources, cost-effective implementations are imperative.

Another low-cost implementation for the MPEG algorithm lies in adopting computationally graceful degradation (CGD) techniques to computation

intensive blocks of the decoder. CGD-based designs provide minimum Quality of service (QoS) when the processing needs exceed the available resources. We found those further reductions in MAC counts and decoder speedup, were possible by a combination of eliminating the upper half of the subband coefficients and utilizing a trimmed $D(m)$ windowing function. The combined effort gives a guarantee towards meeting hard real-time constraints of the cost-effective decoding systems.

The proposed decoder has been implemented on high-performance fixed-point VLIW DSP. The final implementation provides for near-transparent real-time audio decoding and reconstruction of PCM samples.

Table 3. Number of operations in IDCT and window schemes.

Operations		ISO	Modified ISO	Proposed					
				For MPEG-1/2		For MPEG-2 only		Degraded MPEG-1/2	
	#				$\Delta\%$		$\Delta\%$		$\Delta\%$
IDCT block	Multipli-cations	2,048	512	384	-35	288	-43	156	-70
	Additions	1,984	496	350	-30	204	-58	168	-66
Window ing block	Multipli-cations	512	512	256	-50	256	-50	198	-61
	Additions	496	496	256	-52	256	-52	225	-54

* Experimental results are compared against the modified ISO decoder.

Table 4. Execution time profiles of real-time DSP-based MPEG decoders.

Audio Clips	Total play time (sec)	Execution times of decoders (sec)				
		ISO	Modified ISO	For MPEG-1/2	For MPEG-2 only	Degraded MPEG-1/2
Jazz	32.8	36.5	18.7	11.2	9.6	7.5
Classical	58.1	59.1	32.5	22.3	18.9	14.1
Vocal (MPEG-1)	24.3	21.4	10.5	7.1	-	4.8
Conformance test	5.6	4.2	2.3	1.6	1.4	0.9

* Experimental results are compared against the modified ISO decoder.

V. Conclusion & Future Researches

In this paper, we described a real-time realization of an MPEG-2 audio decoder on a single, fixed-point, VLIW DSP chip. The proposed subband synthesis bank employs a decomposed matrix transform block for both MPEG-1 and MPEG-2 decoding, and an additional parameter analyzer block for multichannel decoding. A decoder design implementing graceful degradation techniques has been proposed as a low-cost alternative, by keeping output audio quality to a minimal level while hiking up performance. Our effort concentrated on reducing the number of MAC operations of the computationally intensive synthesis filter bank while preserving signal quality integrity.

For real-time realization on a DSP chip, both DSP and VLIW architectural optimizations were performed. Optimal performance results have been obtained by judiciously adopting platform dependent and independent optimization techniques such as software pipelining loop unrolling, and DSP intrinsics, without falling back on the usual bottom-up assembly patch-up. Despite the versatility of translating and mapping complex standards onto the native DSP hardware via high-level language, efficient compiler targets, robust and tightly integrated debug/simulation environments are still yearned for.

The final realization provides a 60% faster decoder when matched against the modified ISO decoder implementing a fast IDCT block in the synthesis filter bank. Real-time performance obtaining near-transparent audio decoding and reconstruction of PCM samples, have been achieved. The techniques described in this paper can be extended the implementation of multiplexed-with-video applications, stand-alone low-cost decoder solutions, or elevated to core-based ASIC designs with few modifications. As for future CGD-based real-time decoder implementations, a practical complexity estimation policy for programmable cores need to be developed in order to accurately predict decoding time, and thereby guaranteeing real-time performance with resource allocation constraints.

Acknowledgement

This work was supported partly by Daewoo Electronics Co. as part of Digital VDR Technology Development Project.

References

- [1] K. Pohlmann, *Principles of Digital Audio*, 3rd edition, McGrawHill: New York, 1997.
- [2] E. Zwicker and H. Fastl, *Psychoacoustics: Facts and Models*, Springer-Verlag: New York, 1990.
- [3] ISO-IEC JTC1/SC29/WG11, 'Information Technology - Coding of Moving Pictures and Associated Audio for Digital Storage Media at up to about 1.5 Mbit/s - Part 3: Audio,' IS 11172-3, Nov. 1992.
- [4] ISO-IEC JTC1/SC29/WG11, 'Information Technology - Generic Coding of Moving Pictures and Associated Audio Information - Part 3: Audio,' IS 13818-3, May 1995.
- [5] C. Murphy and K. Anandakumar, 'Real-Time MPEG-1 Audio Coding and Decoding on a DSP Chip,' *IEEE Trans. on Consumer Electronics*, Vol. 43, No. 1, Feb. 1997, pp. 40~47.
- [6] J. Chen and H. Tai, 'Real-time Implementation of the MPEG-2 Audio Codec on a DSP,' *IEEE Trans. on Consumer Electronics*, Vol. 44, No. 3, Aug. 1998, pp. 866~871.
- [7] T. Tsai, L. Chen, and Y. Liu, 'A Novel MPEG-2 Audio Decoder with Efficient Data Arrangement and Memory Configuration,' *IEEE Trans. on Consumer Electronics*, Vol. 43, No. 3, Aug. 1997, pp. 598~604.
- [8] K. Rao and P. Yip, *Discrete Cosine Transform*, Academic Press Inc.: London, 1990.
- [9] Texas Instruments, TMS320C62x/C67x Programmer's Guide, 1998.

Biographies



Won-Kyu Paik received the B.S. degree in electronic engineering from Sogang University, Seoul, Korea, in 1997. He is currently working towards the M.S. degree in electrical engineering at Sogang University.

His current research interests include digital video/audio compression, real-time DSP systems, and CAD systems.



Sun-Young Hwang received the B.S. degree in electronic engineering from Seoul National University, Seoul, Korea, in 1976, the M.S. degree from Korea Advanced Institute of Science in 1978, and the Ph.D. degree in electrical engineering from Stanford

University, California, U.S.A., in 1986. Since 1986, he has been with the Center for Integrated Systems at Stanford University, working on design of a high-level synthesis and simulation system. In 1986 and 1987, he held a consulting position at Palo Alto Research Center of Fairchild Semiconductor Corporation. In 1989, he joined the Department of Electronic Engineering at Sogang University, where he is now a professor. His current research interests include hardware/software co-design, and DSP/VLSI systems design.

