# 복수 포멧 지원 오디오 복호화기 설계

# A Design of Multi-Format Audio Decoder

박성욱
Sung-Wook Park

삼성전자 DM연구소

요 약

본 논문은 AC-3와 MPEG-2를 모두 복호화할 수 있는 오디오 복호화기 구조를 제시한다. MPEG-2 합성필터는 AC-3와 공통점을 얻기 위해서 32 포인트 FFT를 활용하도록 변환되었다. 복호화기는 프로그래머블 오디오 DSP 코어와 공용의 합성필터로 구성되어 서로 다른 포멧을 효과적으로 분석, 복호화가 가능하다.

키워드 : 오디오, 멀티포멧, 복호화, AC-3, MPEG-2

Abstract

This paper presents an audio decoder architecture which can decode AC-3 and MPEG-2 audio bit-streams efficiently. MPEG-2 synthesis filtering is modified by the 32-point FFT to share the common data path with the AC-3's. A programmable Audio DSP core and a hardwired common synthesis filter are incorporated for effective decoding of two different formats.

Key Words : Audio, Multi-Format, Decoder, AC-3, MPEG-2

## 1. Introduction

The Dolby AC-3 and MPEG-2 audio standards are two representatives as a high quality audio coding (HQAC) algorithm [1, 2]. A multi-format AC-3 and MPEG-2 audio decoder can provide consumer-economical profits if it implements both algorithms in a single hardware block. In other to get the benefit, the approaches using common synthesis filter were presented [5, 6], which is essential to the implementation of the multi-format decoder.

This paper presents a mult-format audio decoder consisting of a dedicated DSP core and a fast common synthesis filter. The architecture of the DSP core is specially fitted to process a control-intensive part of both AC-3 and MPEG-2 decoding. The common synthesis filter was designed to be a hardwired logic with a common data-path for processing both AC-3 and MPEG-2 synthesis filtering.

The common filter is faster than those of previous works [5,6], and achieves better hardware utilization than that of Lau's [5].

The paper is organized as follows. Section 2 presents algorithmic design for the common data-path. Section 3 shows the architectural design of the multi-format decoder. Results of the simulation and systemization are presented in Section 4, and conclusions are reached in Section 5.

## 2. Algorithmic Design

### 2.1 Decoding Process

Both AC-3 and MPEG-2 decoding process consist of bit-parsing, bit-allocation decoding and unique synthesis filtering of their own [1,2]. Each decoding process can be partitioned into two parts by the task's algorithmic property. Figure 1 shows both formats are partitioned into a control-intensive (CTI) part and a computation-intensive (CPI) part. Synthesis filtering and windowing of AC-3 and MPEG-2 belong to CPI part, and other decoding tasks belong to CTI part.

The CTI part needs various operations of relatively low computational power, which makes the control complex. The CPI part needs repeating operations with high computational load. The fast computation is the most

접수일자 : 2007년 6월 11일
완료일자 : 2007년 7월 30일

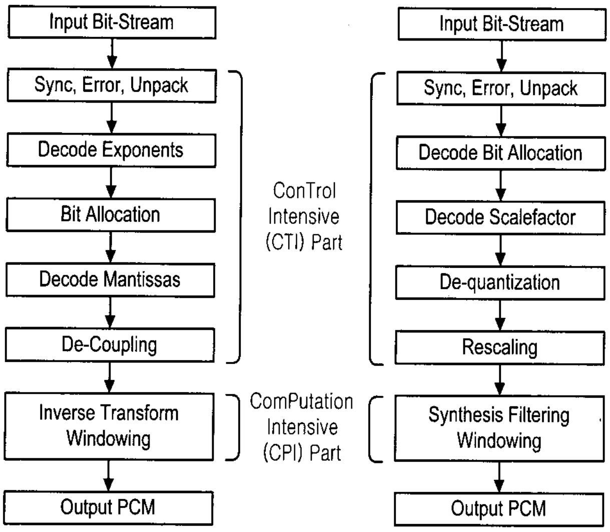


Fig.1 Partitioning of decoding process

essential to real-time implementation.

Considering the property of each part, a programmable core is desirable to perform the CTI part, and a fast hardwired logic is suitable to perform the CPI part.

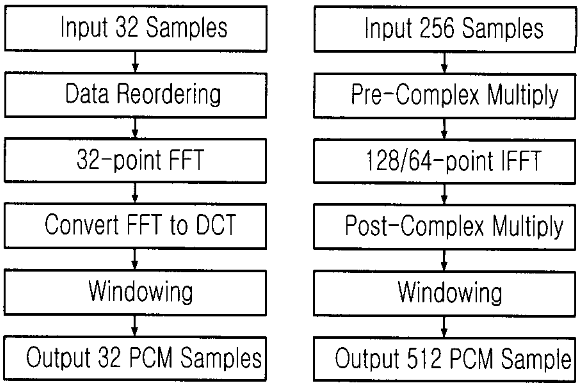
While a single programmable core can easily implement both the AC-3 and MPEG-2 CTI parts of decoding process, the fixed hardwired logic makes the implementation of both AC-3 and MPEG-2 CPI parts almost impossible because each CPI part has its own unique structure of operations. Thus, making common data structure for both CPI parts is the most essential to the common hardwired logic design.

### 2.2 Synthesis Filtering

This paper presents a modified MPEG-2 synthesis filtering process to enable the hardwired common synthesis filter to be designed. The proposed filtering process uses 32-point complex FFT to establish the common data-path with AC-3's.

As for the MPEG-2 fast filtering, Konstantinides suggests a fast algorithm using 32-point DCT [3]. This algorithm reduces the computational load by a factor of about 2, but it doesn't provide the structure common to the well-known AC-3's fast IDCT algorithm using 128/64-point complex IFFT [1]. Thus, the previous works use 64-point complex IFFT to make the common structure with the AC-3 fast algorithm [5, 6].

Figure 2 shows the MPEG-2 modified synthesis filtering suggested in this paper. The Konstantinides' fast algorithm is once again modified by the fast DCT algorithm using 32-point complex FFT [4].



(a) MPEG-2 (suggested) (b) AC-3

Fig.2 Suggested common structure for synthesis filtering

This modification reduces the computational load by a factor of about 4, and establishes the common data-path (FFT/IFFT) for both AC-3 and MPEG-2. The modification also makes both synthesis filterings have the similar pattern, which consists of pre-processing, FFT/IFFT, and post-processing. This similarity can reduce the architectural complexity of the common filter.

Another advantage of suggested filtering scheme is that it decreases the size of required memory. The MPEG-2 standard scheme needs buffer of 64 words size [2], but modified one needs only the half, 32 words size. This reduction is made directly from the use of 32-point FFT.

## 3. Architectural Design

### 3.1. Overall System

The system architecture of the designed multi-format decoder is shown in Fig.3. The multi-format decoder consists of a dedicated DSP core, a hardwired common synthesis filter and several memories.

The DSP core is a dedicated processor suitable for audio signal processing. It performs the CTI part of AC-3 and MPEG-2 decoding process by switching the programs in program ROM (PROM).

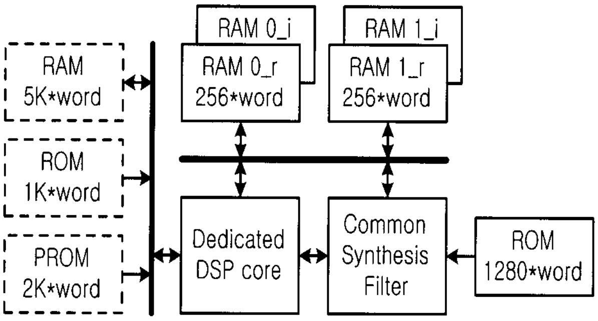


Fig.3 System architecture of multi-format decoder

The common synthesis filter is a fast hardwired logic. It performs the CPI parts of AC−3 and MPEG−2. Because the CPI parts require high computational load, its architecture must be focused on the fast operation. Through internal RAM (RAM0, RAM1), the common filter moves data to and from the DSP core.

Because the complex data operations occupy the most part of the synthesis filterings, the internal RAM is partitioned to be suitable for efficient access to complex data. For example, RAM0 is divided into a real RAM (RAM0_r) and an imaginary RAM (RAM0_i) for concurrent access to the real and imaginary part of the complex data. RAM0 and RAM1 exist in pair, and enable the common filter to read the data and write the result simultaneously.

The part with dotted line in Fig.3 represents the use of external memories. These memories are used only by the DSP core to perform the CTI part of both decoding processes. Figure 3 shows these memories are divided into the data and program memory.

### 3.2 Common Synthesis Filter

The common synthesis filter is a fast hardwired logic, and performs the CPI part of AC−3 and MPEG−2. Each CPI part consists of synthesis filtering and windowing process (see Fig.1). The MPEG−2 synthesis filtering is modified by the FFT, and most computational load lies on the FFT operations of each synthesis filtering. Thus, the design of the common filter is focused on the fast implementation of complex data operation, which is the main operation of the FFT.

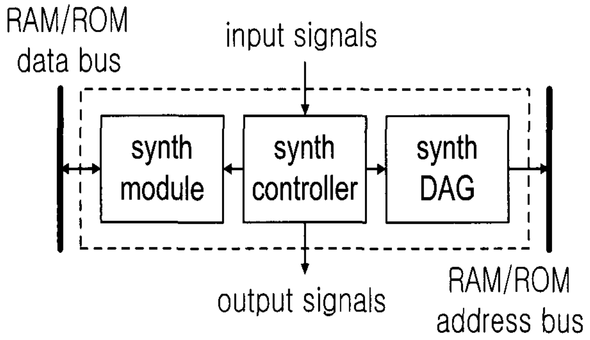


Fig.4 Architecture of the common synthesis filter

Figure 4 shows the architecture of the common filter. The filter consists of a controller, data−address generator (DAG) and a data−path (synth module). The input and output signals of the controller is for the communication with the DSP core. The DAG supports the circular addressing and index addressing mode. The controller is designed by finite state machine (FSM) technique.

The architecture of the data−path, synth module, is shown in Fig.5. The multiplier−multiplier−adder (MMA) logic connected without clock delay in Fig.5 performs the complex multiply in 2 clock cycles. Two parallel adder−rounder (AR) logics, through the pipelining scheme with MMA logic, enable the filter to perform 1 butterfly operation in 2 clock cycles.

The synth module adopts a 7−stage pipeline to perform the butterfly operation of FFT. Since the filter keeps each pipelining stage full at all time, the utilization of all the multiplier and adders is 100% during the FFT/IFFT. This utilization ratio is higher than Lau's approach [5] which uses only 2/3 of the system resources (1 multiplier and 1 adder only).

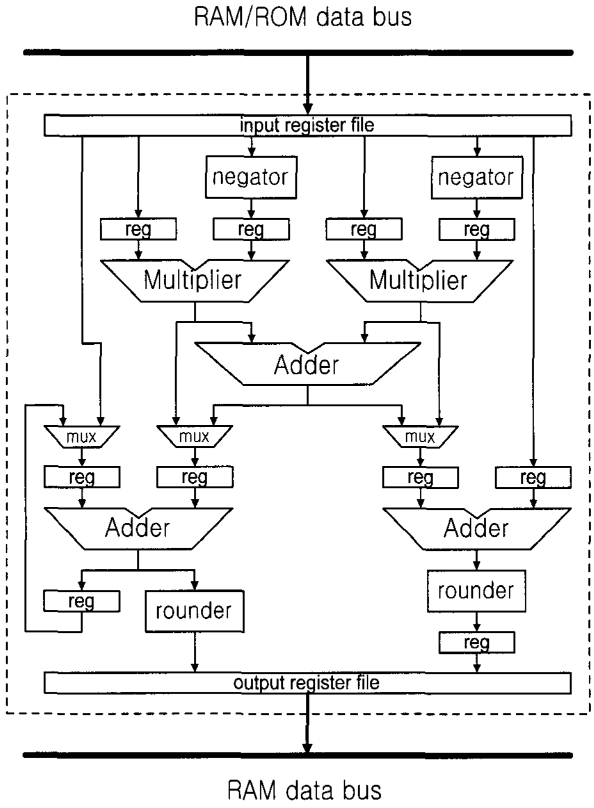


Fig.5 Architecture of the synth module

The MMA logic also contributes to the pre− and post−reordering process of FFT/IFFT. Because these reorderings consist of complex multiplications, MMA also performs these task efficiently. As for the windowing, the MMA multiplies 2 samples by window coefficients concurrently for both AC−3 and MPEG−2 as well as performs the summation for MPEG−2 within a single cycle.

The AR logic performs the overlapping for AC−3 by adding previous block to the current result from MMA. The negator in Fig.5 enables the module to perform both

FFT and IFFT by negating the sign of twiddle factors of internal ROM table. The rounder converts a double precision to a single precision value to minimize the rounding error.

### 3.3 DSP core

The designed DSP core is an application-specific dedicated processor, whose architecture is suitable for audio signal processing and bit-parsing. It adopts 3-stage pipeline for enhancing the performance and the Harvard architecture for efficient pipelining. The pipelining structure consists of instruction fetch, operand read and execution.

The DSP core supports special instructions like MIN, MAX, UNPACK as well as general arithmetic and logical instructions including hardware MAC. Especially, UNPACK is a useful instruction for fast bit-parsing. All the instructions are completed within a single cycle.

The major processing units of the DSP core are a program sequencer (PS), DAG and a data-path. The PS fetches the instruction and decodes it. The DAG uses integer arithmetic to perform the effective address calculation.

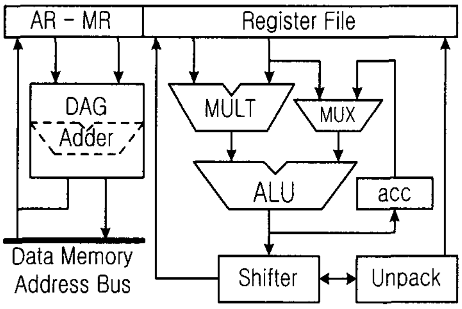


Fig.6 Architecture of the DSP core data-path

The architecture of the data-path and DAG is shown in Fig.6. For sharing resource, two modules are integrated together. Multiplier and ALU are connected without clock delay to support the hardware MAC operation. The shifter performs barrel shift operation, and unpack module supports the hardware bit-unpacking operation within a single cycle. Address register (AR) file and modification register (MR) file is used for supporting various addressing modes.

## 4. Results

### 4.1 Design Validation

Three steps of design validation are applied to the designed multi-format decoder. First step is to examine the validation for the algorithms used. The multi-format decoder adopts the modified MPEG-2 synthesis filtering. The validation of this modification was carried out by floating-point and fixed-point simulation using C. The result of the simulation is that the finite wordlength restriction does not significantly affect this modified scheme.

Second step tests the validation of the design at the architectural level. The multi-format decoder was described by the VDHL. This described system was compiled, and functional simulation was carried out. The result of simulation was compared with the results of fixed-point C simulation. At this step, the emulator for the DSP core was developed by C language to speed up the validation process.

At final step, the hardware simulation with gate delay was checked to validate the real-time implementation. The system clock frequency was given by 20MHz.

Table 1.(a) Clock cyles to decode AC-3 bit-streams(1 frame, 5.1 channel, bit-rate=448 Kbps, fs=48KHz)

| Module | Decoding process | No. of cycles |
|---|---|---|
| DSP core | bit-unpacking | 75 |
| | exponent decoding | 25,134 |
| | bit allocation | 169,038 |
| | mantissa decoding | 137,460 |
| | channel de-coupling | 8,490 |
| | pre-multiply step | 7,680 |
| Common Filter | IFFT | 27,990 |
| | post-multiply | 7,680 |
| | window and overlap/add | 11,700 |
| | Total sum | 395,247 |

Table 2. Gate count for the dual decoder

| Module | Sub-Module | No. of gates |
|---|---|---|
| DSP core | Program Sequencer | 4,428.0 |
| | ALU and DAG | 17,564.8 |
| | Controller | 897.5 |
| | Sum | 22,890.3 |
| Common Filter | Synth module | 15,588.8 |
| | Synth DAG | 3,068.0 |
| | Synth controller | 2,884.3 |
| | Sum | 21,541.0 |
| | Total sum | 44,431.3 |

### 4.2 Multi-Format Decoder System

For the system to operate as a multi-format decoder which can decode each bit-stream in real-time, systematic delay due to the sequential nature of the algorithm must be considered. The systematic delay is directly decided from the amount of clock cycles needed to decode bit-stream. Table 1 shows the processing cycles obtained from the functional simulation tool.

The required cycles for the AC-3 decoding is shown in Table 1.(a). It is 640,000 cycles which is available at maximum for decoding one frame with bit-rate of 448kbps. The result of simulation shows that about 400,000 cycles are required at total. Thus, the designed multi-format decoder satisfies the condition for real-time implementation.

Table 1.(b) Clock cyles to decode MPEG-2 bit-streams(1 frame, 5.1 channel, bit-rate=128 Kbps, fs=48KHz)

| Module | Decoding process | No. of cycles |
|---|---|---|
| DSP core | MPEG-1 bit-unpacking | 3,903 |
| | MPEG-1 de-quantization | 51,630 |
| | MPEG-2 bit-unpacking | 20,963 |
| | MPEG-2 de-quantization | 77,475 |
| | Rematrixing | 53,568 |
| Common Filter | 32-point FFT | 33,660 |
| | converting FFT to DCT | 6,840 |
| | window and decimate/add | 48,600 |
| | Total sum | 296,639 |

Similarly, 480,000 cycles are the maximum number of cycles available for decoding MPEG-2 bit-stream with bit-rate of 128 kbps. Table 1.(b) shows the required cycles for MPEG-2 decoding. The total cycles are about 300,000 cycles, and the multi-format decoder also satisfies the real-time condition for MPEG-2 decoder.

From the results of Table 1, about 240,000 cycles for AC-3 and 180,000 cycles for MPEG-2 are remained as a cycle margin. Thus, the calculated minimum operating frequency of the designed multi-format decoder will be about 12.4MHz.

From the point of cost-effectiveness, the system gate count is a dominant factor for evaluating the designed decoder system. The gate count of the synthesized multi-format decoder is shown in Table 2. The total number of gates is about 45,000.

## 5. Conclusion

We have designed the multi-format audio decoder which consists of a dedicated DSP core and a fast common synthesis filter. The MPEG-2 modified synthesis filtering algorithm using 32-point FFT is suggested to make the common data-path for both AC-3 and MPEG-2 synthesis filtering.

The designed DSP core is an application-specific processor, whose architecture is suitable for bit-parsing and bit-allocation decoding of each algorithm. The DSP core adopts 3-stage pipeline and Harvard architecture.

The common synthesis filter is a fast hardwired logic, and fitted to perform fast FFT/IFFT operation. The common filter adopts 7-stage pipeline, and achieves 100% hardware utilization during FFT/IFFT.

The designed common filter is faster than that of previous Lau's or Jhung's approach, and gives more available cycles to the DSP core. Thus, the multi-format decoder can decode bit-streams of high bit-rate, and can support full set of options of both AC-3 and MPEG-2 standards in real-time.

## References

[1] Advanced Television Systems Committee (ATSC) Standard Doc. A/52, "Digital Audio Compression Standard (AC-3)", Nov. 1994

[2] ISO/IEC JTC1/SC29/WG11 No. 703 "Generic Coding of Moving Pictures and Associated Audio- CD 13818-3 (MPEG-Audio)" Mar. 1994

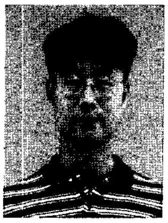
[3] K. Konstantinides, "Fast Subband Filtering in MPEG Audio Coding", *IEEE Signal Processing, Letters*, vol.1, no.2, pp.26-28, 1994

[4] M. J. Narasimha and A. Peterson, "On the Computation of the Discrete Cosine Transform", *IEEE Trans. Comm.*, vol. COM-26, pp. 934-946, Jun. 1978

[5] Winnie Lau and Alex Chwu, "A Common Trans- form Engine for MPEG & AC3 Audio Decoder", *IEEE Trans. on Consumer Electronics*, Vol. 43, No. 3, pp. 559-566, Aug. 1997

[6] Y. Jhung and S. Park, "Architecture of Dual Mode Audio Filter for AC-3 and MPEG", *IEEE Trans. on Consumer Electronics*, Vol. 43, No. 3, pp. 575-585, Aug. 1997

# 저자소개



**Sung-Wook Park** received the B.S., M.S., and Ph.D. degree in Electronic Engineering from Yonsei University in 1993, 1995, and 1998, respectively. He is now working for Samsung Electronics. His research interest includes VLSI signal processing and Multimedia Signal Processing.