

AN ASIC IMPLEMENTATION OF THE MPEG-2 AUDIO DECODER

Sung-Chul Han,* Sun-Kook Yoo,** Sung-Wook Park,* Nam-Hun Jeong,*
Joon-Suk Kim,* Ki-Soo Kim,* Yong-Tae Han,*** and Dae-Hee Youn*

*ASSP Lab., Dept. of Electronic Eng., Yonsei University

**Dept. of Medical Eng., Yonsei University

***Research Center, Korea Telecom

Abstract - MPEG-2 audio is the subband coding technology used for various audio applications. This paper presents a semi-custom ASIC design for the MPEG-2 audio decoder. The decoder implemented in this paper meets the requirements of MPEG-2 international standard, and is divided into three parts: the preprocessor, the multichannel processor, and the synthesis filter. The decoder system has been designed in VHDL(VHSIC Hardware Description Language), and developed as a single chip in a 0.6 μm CMOS semiconductor process.

I. Introduction

The ISO(International Standardization Organization) MPEG-2 audio standard is the result of many efforts that have been dedicated to overcoming problems in storage and transmission of digital audio. MPEG-2 audio is basically a subband analysis which exploits the human auditory characteristics to achieve a low bitrate with minimum perceptual loss of signal quality. It also utilizes various multichannel data compression techniques to adopt the extension of 5 channels[1][2][3][4].

In many digital signal processing applications, commercially available digital signal processors are used. But the use of a general-purpose DSP(Digital Signal Processor) is not always desirable since additional hardware is required to realize a complicated algorithm and some parts of the DSP are not used at all. These wasted gates make it impossible to obtain an optimized system. Advances in ASIC technology has given us the ability to design a large system and realize it on a chip in a reasonably short time. In order to be an efficient signal processing system, it should be fully application-specific by designing a processor core and additional hardwired logic that exactly meet the requirement of the algorithm to be implemented.

The goal of this paper is to present a real-time audio decoder implemented using ASIC technology, which

is capable of decoding MPEG-2 standard multichannel audio bitstreams. This paper is also intended to show a system architecture for MPEG-2 audio decoder that has been carefully designed to reduce the chip area and lessen the difficulty in test and verification.

Firstly, the configuration and features of the implemented system are introduced. In this section, the basic concept embedded in the system architecture is also discussed. Then the main functional blocks comprising the system are described in detail.

II. System Configuration and Fuctionality

The MPEG-2 audio decoding process begins with receiving encoded bitstreams from transmission channels. The received bitstream is stored in a buffer, and then transferred to the decoder when asked for to be analyzed.

Firstly, the decoder performs analysis on the received bitstream. This process consists of header interpretation, parameter extraction, and data extraction. Some important information to control overall operation of the decoder is included in the header, and it is interpreted to gather system control signals and the information on encoding mode. In the parameter extraction step, such parameters as bit allocation information, scalefactor select information, and scalefactor information are extracted. The data extracted from the bitstream are quantized samples which have been encoded based upon the psychoacoustic model.

The extracted data are multiplied by the corresponding scalefactors to become subband samples, which would be equivalent to the output of the analysis filterbank in the encoder if encoding and decoding process were neglected. Since the MPEG-2 encoder compresses multichannel data and scalefactors using the interchannel similarity among the signals and inserts the additive multichannel coding information into the header, the reverse process called the multichannel decoding

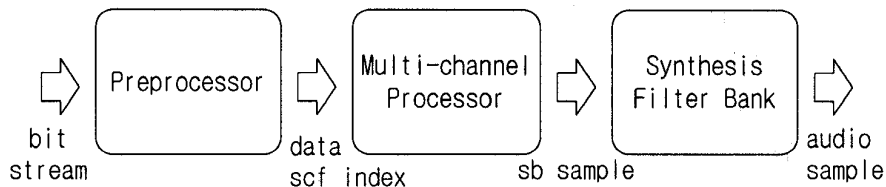


Figure 1. MPEG-2 Audio Decoder Architecture

should be performed on the header and data acquired in the analysis step.

After the multichannel decoding process the subband samples are transformed to time-domain samples when they are processed by the synthesis filterbank. This process is the most time-consuming one in the MPEG audio decoder[2].

The MPEG-2 decoder contains 3 primary modules called the preprocessor, the multichannel processor, and the synthesis filterbank, respectively, as shown in the Figure 1. The architecture of the proposed audio decoder has been partitioned considering the efficiency in design and verification, and each part has been designed using VHDL, synthesized, and then verified with post-synthesis simulation.

III. System Descriptions

A. Preprocessor

Preprocessor extracts header information from an audio bitstream, which is used to obtain multichannel modes and other control information. Audio data contained in the bitstream are also extracted, and sent to the multichannel processor via a buffer. Using the sampling frequency information, the preprocessor also generates timing information such as F_s and d_{32} to synchronize system components, where F_s represents sampling frequency and d_{32} is the time within which the decoder should process 32 samples. A frame contains 1152 samples so that the time duration of a frame is equal to that of 36 occurrences of the d_{32} signal.

Since the decoding process varies according to the header information regarding coding modes, bit allocation, and scalefactor select information, the preprocessor should employ a data extraction algorithm that can accommodate

various coding modes and changes in bit allocation scheme. To meet these requirements a specially designed microprocessor core is used and the extraction algorithm is microprogrammed in it. The microprocessor core operates at 54MHz, and, as shown in Figure 2, consists of an ALU, on-chip RAM, and an overall controller, ROM for programs and data tables, and an external DRAM interface.

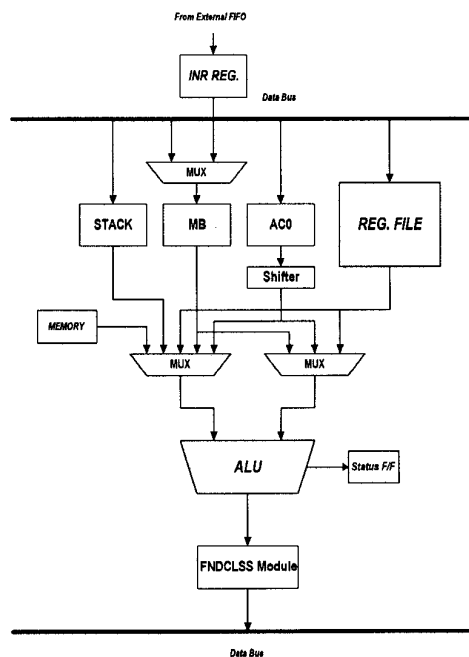


Figure 2. The Structure of the Processor Core

To obtain an efficient preprocessor, the processor core has been designed to have the following features.

Firstly, a separate logical circuit block which controls the output of the channel buffer, has been fabricated so that no additional programming efforts for buffer control might be necessary. When the processor

controller reads one bit, 1-bit request is applied to the INR register unit. The MSB of INR register is shifted into the memory buffer register MB. After the LSB, the eighth bit, is shifted to the MB, a new 1 byte is transferred from the FIFO to the INR, and then the shift operation is repeated. This operation is shown in Figure 3. If the FIFO is empty, the controller stops all the operations of the processor and waits for the new bitstream to come in from the channel.

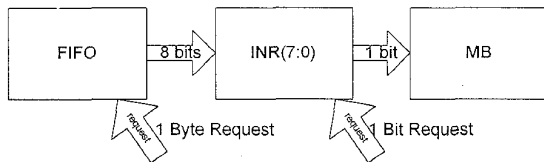


Figure 3. Bit Extraction from a bitstream

Secondly, an automatic address increment is possible for easy sequential access to the memory block without an additional programming. Among the information extracted from the bitstream, parameters for parsing audio data such as bit_allocation, scfsi, tc_allocation, and dynamic_crosstalk are frequently used, so that each

parameter is assigned a separate memory block, and MPEG-2 decoder sequentially scans these memory blocks with the automatic address increment feature.

Thirdly, an indirect addressing is possible for look-up table search which frequently happens.

Fourthly, it has an 8-level hardware stack, which is used to store the contents of the program counter, loop counter, etc.

Fifthly, it has an 1k*16 bit internal RAM and 32 registers in total. Using these the time consumption in accessing external DRAMs is minimized.

Finally, it has a 16 by 16 bit multiplier and divider, and other arithmetic logic units. It also supports application-specific instructions for analysis of the MPEG-2 bitstream. For example, "bread N" instruction in Table 1 enables the core to read N bits from the bitstream.

One instruction normally consumes 4 clock cycles as shown in Table 1, but division and memory read/write instructions require 20 and 7 cycles, respectively.

B. Multichannel Processor[2]

While the preprocessor takes charge of the bitstream analysis under the time constraint for real-time operation, the multichannel processor reconstructs subband samples from the compressed data, and passes the subband samples to the synthesis filter.

The compressed data result from the normalization, channel matrixing, and composite encoding such as dynamic crosstalk and phantom coding. To reconstruct original subband signals, multichannel processor consists of a composite decoding unit, a dematrixing unit, a denormalization unit, and a control module to control all of these units. In addition, an IIR filter is included in the multichannel processor to support the dematrixing procedure 2. Figure 4 shows a schmetic diagram of the multichannel processor.

Since the multichannel processor requires less computations than other modules, it operates at 27MHz clock speed which is the half of the 54MHz system clock. This processor is activated by the d32 signal coming from the preprocessor. When the d32 signal goes to high, this processor begins to construct 32 subband signals. The processor begins processing for the first subband. After finishing processing for 5 channel data belonging to the first subband, the same processing for the second subband starts. The processing continues until the entire 32 subbands are processed, and then the

Table 1. Instruction Set

<i>Nmonic</i>	<i>Function</i>	<i>cycle</i>
read MB.(mem)	MEM → MB	7
write [mem].MB	MB → MEM	7
move R1 . R2	R2 → R1	4
move R .value	val. → R	8
read MB . indexR	(indexR)→ MB	7
write indexR . MB	MB→(indexR)	7
push R	R +1 → ST	4
pop R	ST→ R	5
add R1 . R2	R1 + R2 → R1	4
shiftR N	sftRN(AC0)→AC0	4
shiftL N	sttLN(AC0)→AC0	4
sub R1 . R2	R1 - R2 → R1	4
mult R1 . R2	R1 x R2 → R1	4
div R1 . R2	R1/R2→R1 (mod→R2)	20
cmp R1 . R2	R1 - R2 Status F/F set	4
inc R	R +1 → R	4
findclass R	R→(fndcls module)→ R	4
Bread N	INR→MB (N bit)	4 + N
Bread RDC	INR→MB ((RDC)bit)	4+ (RDC)
Bacc	INR→MB (1 bit)	5
JIZ addr.	Jump.inc indexR MB is zero	8
jump addr.	Jump to Addr.	4
cjmp addr.	Jump if Cond. is true	4

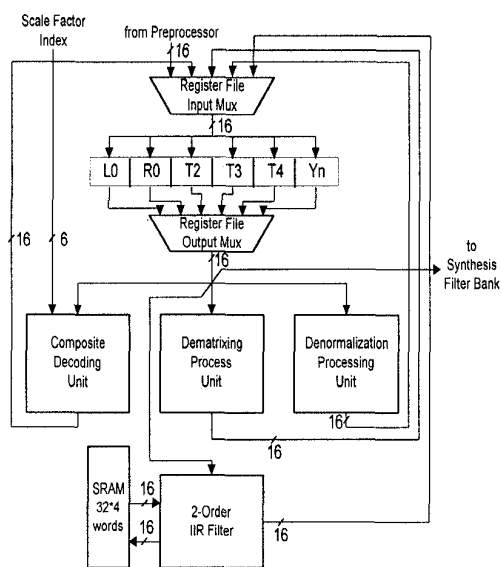


Figure 4. The Structure of the Multichannel Pro

multichannel processor goes to the waiting state until the next d32 signal goes to high. Figure 5 explains the behavior of the multichannel processor.

The composite decoding unit reconstructs the matrixed five channel signals(L0, R0, T2, T3, T4) by multiplying compressed data by the corresponding scalefactors. To do this, a 16 by 16 bit sequential multiplier and a scalefactor table containing 16 bit scalefactors have been included. When the multichannel processing for a subband begins, the multichannel processing information and 5 channel data are read and stored in separate internal registers. After these data are loaded, the composite decoding unit reads the scalefactor index corresponding to each channel and reconstructs 5 channel signals by multiplying the data by the scalefactors. The resultant 5 channel signals are stored in 5 separate data registers.

The dematrixing unit has an accumulator for addition and subtraction to reconstruct weighted five channel data. For the case of dematrixing process 2, the filtered signal of $(T3+T4)/2$ is used. Regardless of the matrixing procedure, the IIR filtering is performed prior to the dematrixing procedure for the convenience of designing and the consistency in timing.

The IIR filter consists of a multiplier, accumulator, and memory blocks to store the past data. For the simplicity of the hardware, the multiplier is designed to compute the multiplication of a 16 bit signed and 16 bit unsigned numbers. The negative coefficient means

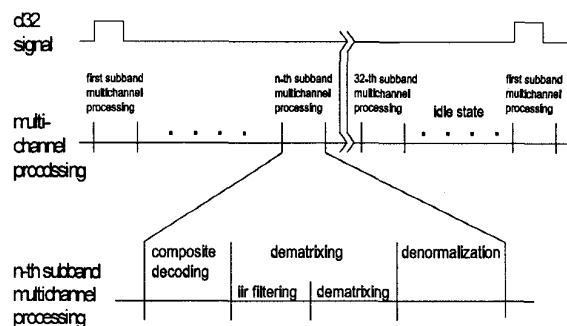


Figure 5. Behavior of the Multichannel Processor

subtraction in the accumulator. Since a second order IIR filter is used, 4 memory blocks are provided to store 4 past input and output samples for 32 subbands.

Five channel data processed through the dematrixing unit are transformed into the final subband signals by multiplying the denormalization factor defined by the dematrixing procedure. This process is accomplished by the denormalization unit which employs a multiplier performing the multiplication of a 16 bit signed and 18 bit unsigned numbers.

At the end of the procedures described above, five channel signals are stored in five registers(L0, R0, T2, T3, T4), respectively. Each register and the corresponding channel signal are determined from the channel switching information. Register contents together with the corresponding channel and subband information are passed to the synthesis filterbank.

C. Synthesis Subband Filter[5]

The synthesis filter reconstructs the time-domain signal from the subband samples transferred from the multichannel processor. Since synthesis filtering is the most time-consuming process in the MPEG audio decoder, it should be divided into smaller functions and each part should work in parallel for real-time operation.

As specified in the MPEG audio international standard, 32 subband samples from the multichannel processor are processed in a few stages till 32 new audio samples are built. These processing steps can be simplified to two stages: the multiplication of cosine matrix and input subband samples and windowing/overlap-add. Two identical MAC units perform operations for each step, forming a two-stage pipelined structure. Each MAC unit consists of a 16 bit by 16 bit array multiplier and a 36 bit accumulator to allow up to

32 accumulations without loss in precision. Also contained in the synthesis filter are ROMs to store the cosine matrix and window coefficients, RAMs to store intermediate data, and controllers to generate addresses for internal memory access and provide control signals for the MAC units. Figure 6 show the structure of the synthesis subband filter.

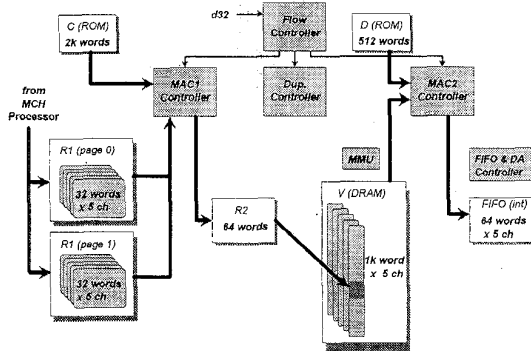


Figure 5. The Structure of the Synthesis Filter

The MPEG-2 subband filter uses 1024 samples of past intermediate data in overlap-add process. This means that 1024 words for each channel should be available in memory at any time, which is too large an amount to be integrated into a single chip. Therefore, a memory management unit is included in the synthesis filter to control an external DRAM.

Basically, it requires 5 identical synthesis filters to transform subband samples from as many channels into audio signals. In the implemented decoder, however, only one synthesis filter performs all the operations for 5 channels by time sharing. The way how 5 channels share one synthesis filterbank is shown in Figure 7. In the first time slot the MAC-1 performs cosine matrix

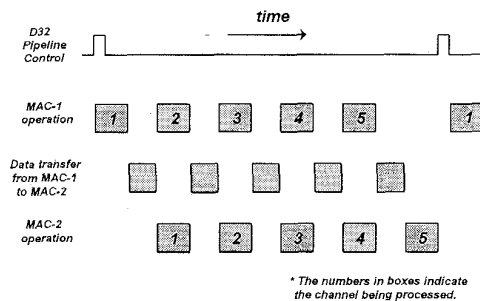


Figure 6. The Timing Diagram of the Synthesis Subband Filter

multiplication using the data from the first channel, and when the intermediate data transfer is over, it repeats the same process for the second channel. The other unit, MAC-2, begins overlap-add process for the first channel as soon as the MAC-1 starts operations for the second channel. From this time on, the two MAC units always work together except for the last time slot. All the necessary processes for the remaining channels are performed in pipeline in a similar way.

The time-domain samples, which are the final result of synthesis filtering, are temporarily stored in the external DRAM, and read back later and converted to serial data to provide convenient interface with external DACs(Digital-to-Analog Converters).

IV. Conclusion

The MPEG-2 audio decoder implemented and presented in this paper was divided into a number of parts according to the their functionality. Each module comprising the system has been designed rather independently of one another to achieve efficiency in design and verification. The proper operation of each module has been verified by comparing the result of computer simulation with that of post-synthesis simulation. All modules operate at the 54 MHz system clock. To reduce the chip area, an external DRAM support logic was installed in the chip., Arithmetic units are designed differently at each module deeply considering the trade-off relationship between speed and area. Bitstreams for each channel are applied to the input of the system through FIFO buffers and the signals of up to five channels are output in serial format as well as timing information for synchronization. The proposed system works as a core of the layer II multichannel decoder with limited accuracy. It supports mono, stereo, dual, intensity stereo modes, phantom channel coding, dynamic crosstalk, and dynamic transmission channel switching. Dematrixing procedure 0, 1, 2, and 3, and the decoder configuration 1/0, 2/0 and 3/2 are also supported.

References

- [1] ISO-IEC JTC1/SC29/WG11 "Coding of Moving Pictures and Associate Audio for Digital Storage Media at up to about 1.5 Mbps-CD 11172(Part-3,MPEG-Audio)" 1991, Nov.
- [2] ISO-IEC JTC1/SC29/WG11/No803 "Coding of Moving

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts



Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research



With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips



Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS

Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS

Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS

Sync your system to PACER to automate legal marketing.