

MPEG Audio 비트스트림의 효율적 처리를 위한 입력 버퍼에 관한 연구

임 성 룡(林成龍)*, 공 진 흥(孔鎭興)**

광운대학교 컴퓨터공학과

전화 : (02) 940-5126 / 팩스 : (02) 940-5121

A Study on the input buffer for efficient processing of MPEG Audio bitstream

Seong-Ryong Lim, Jin-Hyeung Kong

Department of Computer Engineering Kwangwoon University

E-mail : dara@explore.kwangwoon.ac.kr

Abstract

In this paper, we described a design of the input buffer system for efficiently dealing with MPEG audio bitstream to demux header and side information, audio data. In order to overcome the limitations of fixed-word manipulation in bitstream demuxing, we proposed a new variable length bit retrieval system with FSM sequencer supporting MPEG audio frame format, and serial buffer demuxing audio stream, FIFO circular buffer including header and side information.

I. 서론

범용의 DSP 프로세서는 다양한 DSP의 알고리즘 효율적 구현을 위해 개발된 프로세서이다. 특히 DSP 프로세서로 구현되는 알고리즘들은 대부분이 MAC (multiply-accumulate) 와 선형 데이터 접근 같은 전형적인 DSP 연산에 적합한 알고리즘들이다. 그러나 MPEG Audio 알고리즘 같은 지각부호화 알고리즘^{[2][3][4]}은 비선형적인^[6] 데이터 액세스와 많은 조건 및 분기문 처리를 요구하는 복잡한 알고리즘이다. 따라서 이와같은 알고리즘을 기존의 범용 DSP 프로세서로 구

현함은 다음과 같은 비효율성을 갖게된다. 고성능의 DSP (floating point) 프로세서로 구현하게 되면 긴 시간 주기동안 idle 상태를 유지하게 되어 성능의 낭비를 초래할 수 있고, 반면에 저성능의 DSP(fixed point) 프로세서를 사용하게 되면 실시간 처리를 위해 많은 최적화 작업이 필요하게 된다는 단점이 있다. 특히 MPEG 알고리즘은^[1] 데이터 처리에 있어서 크게 두 파트로 나누어서 최적화가 가능하다. 하나는 비트스트림 핸들링이 요구되는 디택싱부 와 순수 DSP 연산이 추가 되는 데이터 연산부 이다. 범용의 DSP 프로세서는 순수 DSP 연산(MAC) 처리와 선형적인 데이터 액세스에 적합한 구조를 갖고 있기 때문에 순수 데이터 연산부 처리에는 적합하지만 디 택싱 부처럼 많은 분기문과 비트단위 데이터 처리, 비 선형적인 데이터 액세스를 요구하는 처리에는 비효율성을 갖게 되며 비트스트림 입력데이터 처리를 순수 DSP 프로세서가 전담하게 되면 비트입력에 따른 빈번한 인터럽트 처리에 따른 성능의 과부하를 안게 된다.

이러한 문제점을 해결하고자 본 연구는 비트스트림 입력 데이터 처리를 위한 별도의 HW 전용 모듈을 설계하여 입력데이터 처리에 소모되던 MIPS치를 낮추고자 한다.

II. 본론

1. MPEG 오디오 스트림(Audio Stream) 구조

MPEG 오디오 프레임^{[1][3][4][5]}은 크게 헤더(header)와 부가정보(side Information), 그리고 오디오 데이터의 세 부분으로 나뉘어 진다. 헤더는 싱크(sync)와 전송률(bit rate), 샘플링(sampling rate) 등의 프레임(frame)에 관련된 정보를 담고있고 부가정보(side information)는 MPEG 데이터에 관한 추가적인 정보를 담고 있으며 마지막으로 오디오데이터가 실제적으로 압축된 음악정보를 담고 있다.

가. 헤더 정보(header information)^[4]

헤더 정보는 4바이트로 구성되어 있고 각 비트에 해당되는 정보는 그림과 1과 같다. 따라서 헤더 정보의 디렉스는 4바이트의 데이터로부터 각각의 비트를 아래 정의된바와 같이 끊어 읽어서 메인 데이터 디코딩에 필요한 변수에 적절히 변환하여 저장하는 일을 하게 된다.

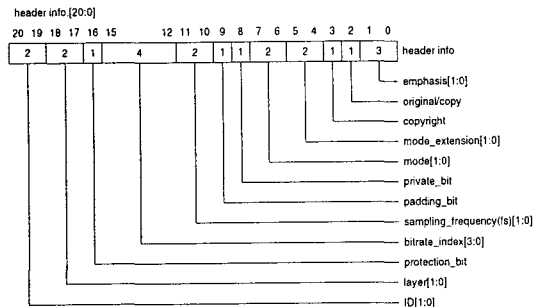


그림 1 헤더 정보 구조

Fig.1 Header information structure

나. 부가 정보(side information)^[4]

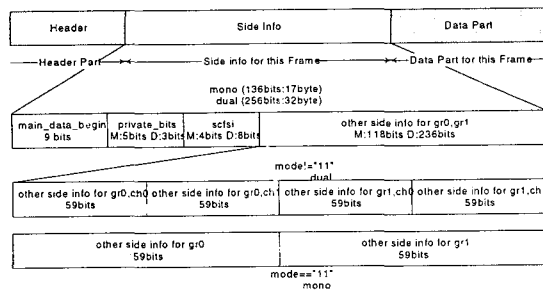


그림 2 부가 정보 구조

Fig.2 Side information structure

부가 정보(side info)는 오디오 압축에 관련된 요소(Parameter)들의 집합이며 구조는 그림 2 와 같다. Decoding 시에 모든 요소들이 디코딩 되지 않고 헤더 정보 및 부가 정보에 따라 선택적으로 디코딩 된다.

다. 오디오 데이터(Audio Data)^[4]

메인(Main data)라고 하며 역양자화를 위한 요소값에 대한 정보를 가지고 있는 스케일 팩터 부분과 허프만 인코딩을 통해 압축된 허프만 데이터로 구성되며 구조는 다음 그림 3과 같다.

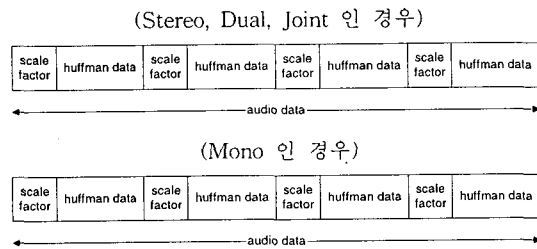


그림 3 오디오(main) 데이터 구조

Fig.3 Audio data structure

2. 입력 버퍼 시스템 구성 및 설계

가. 입력 버퍼 시스템구성

전체 버퍼 시스템 구성은 그림 4와 같다. 먼저 비트 스트림이 입력되면 싱크 검출 모듈이 싱크를 검출하게 되는데 싱크를 검출하지 못하면 비트스트림은^[3] 16 bit 크기로 DSP 프로세서의 데이터 메모리로 저장된다. 이는 MPEG 오디오 표준안에도 나와 있듯이 현재 프레임의 오디오 데이터가 이전 프레임에도 있을 수 있는 특성에 기인한다. 싱크를 검출하게 되면 FSM은 쉬프트 카운터를 시작하고 8비트 레지스터에 데이터를 로딩 한다. 이후 헤더 정보 전송 상태에 의해 이미 주어진 헤더 길이 정보를 가지고 가변 비트단위의 헤더 정보를 fifo 메모리로 전송한다. 이 상태가 끝나게 되면 이어서 side 정보 전송상태가 시작되는 데 이 상태에서 주어진 side 정보 길이에 의해 side 정보를 fifo 메모리로 전송한다 전송이 끝나면 FSM은 DSP 프로세서에게 fifo 메모리 사용을 허가하는 신호를 DSP 프로세서에게 주게 되고 DSP 프로세서는 FSM이 디렉스해는 헤더 및 부가 정보를 내부로 가져가게 된다. 다음 상태는 main data 전송 상태로 FSM은 DSP 프로세서와 DMA 방식에 의한 main data 전송을 시작하게 된다. DSP 프로세서가 fifo 메모리에 있는 헤더 및 부가 정보를 읽어내려 할 때 FSM은 main data를

DSP 프로세서의 데이터 메모리로 16 bits 단위로 전송하게 된다.

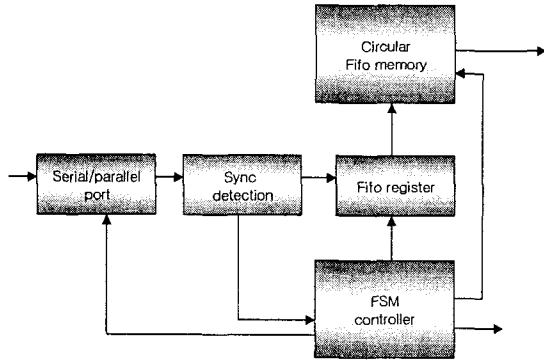


그림 4 입력버퍼 시스템 블록도
Fig.4 Block diagram of input buffer system

나. 모듈별 기능 및 동작

Sync data check 블록(sync detection 블록)

12개의 연속되는 1의 data가 들어오면 sync_enable 신호를 발생하여 FSM으로 보내게 된다. 이는 FSM의 신호에 대하여 내부 전체 블록의 동작 제어를 FSM이 하도록 제어한다.

FIFO memory (144 by 16bits)

fifo 메모리는 2 프레임의 헤더 및 부가정보를 저장할 수 있는 크기로 설계하였으며 다음 그림 5와 같은 순환 기법을 적용하였다.

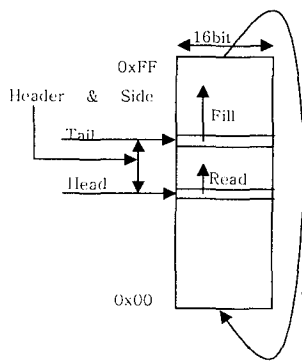


그림 5 순환 fifo 메모리
Fig.5 Circular fifo memory

fifo register 블록

fifo 레지스터 블록은 항상 온 상태로 있다가 SIPO

레지스터에서 parallel로 출력을 하면 serial-in/parallel-out block에서 8bit를 한 clk에 load한다. 이후 serial to serial를 통하여 한 clk 당 한 비트씩 shift 시킨다. FSM에서 head 정보 및 side 정보 상태가 시작되면 serial to parallel 블록으로 shift 되어 FSM의 head 상태에 따라 가변 비트 길이의 data를 FIFO memory로 write한다. side 정보 또한 이와 마찬가지로 정보를 보낸다. fifo 레지스터의 데이터 패스를 그림 6과 같다.

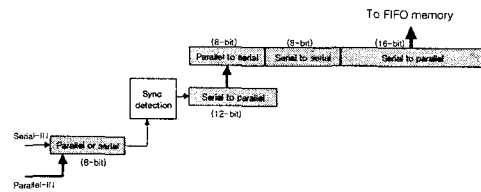


그림 6 fifo 레지스터의 데이터 패스
Fig.6 Data path of fifo register

FSM controller

FSM의 기능은 다음과 같으며 표 1과 같으며 그림 7은 FSM의 상태천이를 보여준다.

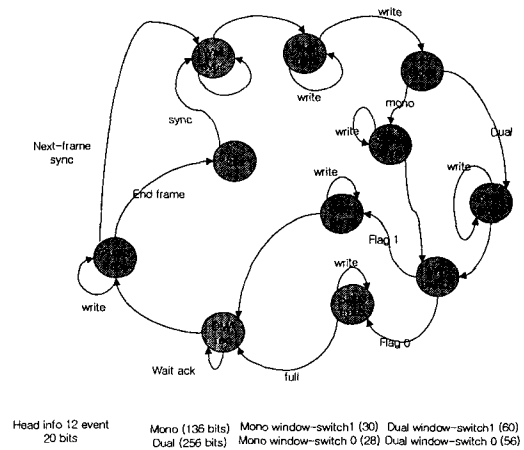


그림 7 FSM 상태 천이도
Fig. 7 FSM state transition diagram

FSM의 상태는 처리기능중심으로 분류하였으며 시작 상태는 idle 상태에서 시작하게끔 하였다. 헤더 정보는 정보의 길이가 프레임마다 변하지 않는 고정길이를 가지고 있지만 부가 정보는 헤더 정보의 모드값에 따라서 다른 길이를 갖게 되는 특징이 있다. 이러한 점

출 과정을 FSM 모듈안에서 자체 처리하도록 구현하여 되도록 DSP 프로세서와의 인터페이스를 최소화하는 방향으로 구현하였다.

표 1 FSM의 상태
Tab.1 FSM States

상태	의미
wait-sync	sync를 받기 위해 대기하는 상태.
head-trans	헤더정보를 fifo 메모리로 내보내는 상태
mode	head info로부터 mode bits값을 검출하는상태
side1-trans	부가정보를 fifo 메모리로 내보내는 상태.
side2-trans	부가정보를 fifo 메모리로 내보내는 상태
win	window switch flag 값을 검출하는 상태
side3-trans	부가정보를 fifo 메모리로 내보내는 상태.
side4-trans	부가정보를 fifo 메모리로 내보내는 상태.
DMA req	DMA 요구인식을 받기 위해 대기하는 상태
audio-trans	오디오 데이터를 데이터 버스에 내보내는 상태
idle	데이터 이동이 전혀 없는 상태

III. 실험 및 결과

설계는 Synopsys를 이용한 VHDL로 설계하여 각 모듈별 합성하고 이를 Altera tool에서 port 맵을 통해 전체 시스템을 구성하였다. 각 모듈별 합성은 Altera FLAX 라이브러리를 사용하였고 Altera FPGA로 타이밍 검증을 수행하였다. 합성된 입력 버퍼 시스템은 총 GATE 수(fifo memory 제외)가 1411 개이고 동작 속도는 11.69MHz이다.

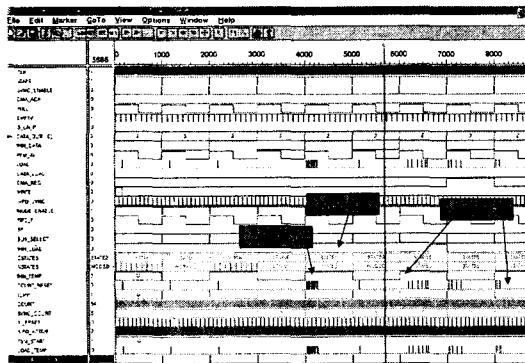


그림 8 입력 버퍼시스템 기능검증
FIG.8 Functional simulation of input buffer

설계된 HW 전용 입력 버퍼 시스템의 성능분석은 다음 표 2 와 같다. 표에서 "*"은 fifo 메모리를 제외한 성능이며, "**" 는 fifo 메모리를 포함한 성능분석이다.

표 2 성능 분석

Tab.2 Performance analysis

modules	gate 수	MHz	clock (ns)
S/P-port	18	125.0	8.0
Sync-detection	363	13.31	75.1
FIFO-register	114	125.0	8.0
fifo-memory	6218	26.95	37.1
FSM controller	916	20.16	49.6
입력버퍼 시스템	1411	11.69*	85.5*
		14.08**	71**

IV. 결론

본 논문에서는 MPEG 오디오 데이터인 비트스트림 형식의 입력데이터 처리에 따른 빈번한 인터럽트 처리에 소모되는 DSP 프로세서의 부하를 덜고 MIPS치를 낮추고자 비트스트림 입력 데이터를 전달하여 Demux 하는 별도의 전용 HW 모듈을 설계하였다. 설계된 전용 HW 모듈인 입력 버퍼 시스템은 총 게이트 수가 1411 개이고(fifo memory 제외) 동작 속도는 11.69 MHz 이다. 설계된 모듈은 DSP의 입력 데이터 처리에 소모되는 시간을 절약할 수 있을것으로 기대되며 추후 연구과제로는 이를 입증할 실험을 수행하는 것이며 또한 FSM 모듈의 상태값을 programmable 가능하게 확장시키는 연구가 필요하다.

참고문헌

- [1] ISO/IEC 11172-3, Coding of moving pictures and associated audio for digital storage media at up about 1.5Mbit/s,1993.
- [2] E. Torick, "Triphonic Sound System for Television Broadcasting", 124th Techn. Conf of the SMPTE
- [3] ISO/IEC JTC1/SC29/WG11 No.972, "Dolby AC-3 Multi-channel Digital Audio Compression System Algorithm Description", Nov., 1993
- [4] ISO/IEC JTC1/SC29/WG11 No.1519 "Generic Coding of Moving Pictures and Audio - CD 13818-3(Part3.MPEG-Audio)", 2nd Edition, Feb., 1997
- [5] ISO/IEC JTC1/SC29/WG11 No.1650 "IS 13818-7(MPEG-2 Advanced Audio Coding, AAC)," Apr., 1997
- [6] N. Jayant, J. Johnston, R. Saffrnek, "Signal Compression Based on Models of Human Perception", Proc. of IEEE Oct., 1993