

HD급 멀티미디어 Streaming을 위한 DSP 플랫폼

*홍근표, 박종순, 문재필, 김동환, 장태규
중앙대학교 전자전기공학부

e-mail : *hkp002@dmc.cau.ac.kr tgchang@cau.ac.kr*

A DSP Platform for the HD Multimedia Streaming

*Keun-Pyo Hong, Jong-Soon Park, Jae-Pil Moon, Dong-Hwan Kim,
Tae-Gyu Chang
School of Electrical Electronics Engineering
Chung-Ang University

Abstract

This paper proposed the design and implementation of a DSP platform for the various multimedia streaming. The DSP platform synchronizes with host PC to configure DSP and to transmit multimedia streaming through PCI. The suggested DSP platform decodes high-capacity video/audio data using the suggested high-speed FIFO, CPLD and memory interface. The buffer control techniques is proposed in other to avoid the under/over-run of the audio/video data during the audio/video decoding. For the DSP platform test, host PC transmits program stream(PS) that consists of the MPEG-2 video MP@ML and 5.1ch AC3 audio data (Coyote.mov file, half hour running time) to DSP platform. The DSP platform plays continuously back the high sound-quality audio and high-definition video at once.

I. 서론

인터넷의 보급화와 네트워크의 고속화가 급격이 이루어짐에 따라 사용자는 좀 더 나은 화질, 고음질의

멀티미디어를 원하고 있다. 이러한 고용량의 멀티미디어를 압축하거나 복원하는데 신호처리가 요구하는 연산량이 크기 때문에 멀티미디어를 처리하는 플랫폼에는 신호처리 전용의 DSP를 탑재하는 것이 유리하다. 최근에 들어서 공중과 방송도 HD 방송을 시험적으로 송출하고 있기 때문에 고화질의 HD급 멀티미디어의 처리가 불가피하다. 또한 고유의 기능만을 담당했던 생활가전, 핸드폰, PC, TV, 셋탑 등의 전자제품과는 달리 두 가지나 그이상의 기능을 융합하는 디지털 컨버전스[1] 경향이 두드러지고 있다. 다양한 형식의 멀티미디어 콘텐츠를 처리할 수 있는 유연한 하드웨어 플랫폼이 요구된다.

본 논문에서는 확장성과 유연성을 가진 General Processor (DSP) 기반의 S/W를 사용하기 위해 DSP 플랫폼 H/W구조를 설계하였다. 호스트 PC에 DSP 플랫폼을 장착하여 DSP configuration을 용이하게 하였다. 이렇게 함으로써 다양한 오디오, 비디오 코덱을 처리할 수 있는 유연한 하드웨어 플랫폼을 확보할 수 있다. MPEG-2 video MP@ML[2] 비디오 코덱과 5.1ch 48kHz AC3 오디오 코덱을 디코딩하기 위한 소프트웨어를 함께 포팅하였다. 본 논문에서 제시하는 H/W 구조는 비디오와 다채널의 오디오 입출력이 가능하며 실시간의 복잡한 연산을 필요로 하는 멀티미디어 응용 프로그램을 범용의 DSP에서 수행하도록 하였다. 재구성 가능한 DSP 플랫폼의 H/W 구조, 오디오 및 비디오의 디코딩 구조, 호스트 PC와의 연동을 통한 스트리

밍 환경, 그리고 비디오와 오디오 데이터의 오버런과 언더런이 발생하는 스트리밍 상황을 극복할 수 있는 buffer control기법을 기술하였다.

II. DSP 플랫폼 H/W 구조 연구

고속의 데이터 처리를 위해 600MHz의 TI DSP인 TMS320C6416[3]을 사용하여 33MHz의 PCI를 통한 PCI 인터페이스가 가능하도록 하였으며, 외부 메모리로는 150MHz의 256MByte와 100MHz의 64MBytes를 가지고 있다. 166MHz의 고속 메모리를 사용하여 150MHz 인터페이스가 가능하도록 하였다. 외부 인터페이스를 지원하기 위해 64개의 EDMA(Enhanced Direct Memory Access) channel이 있으며 이는 오디오를 위한 MCBSP, PCI, 메모리를 위한 EMIF, 비디오를 위한 GPIO interrupt를 RTOS에서 우선순위에 따라 컨트롤한다.

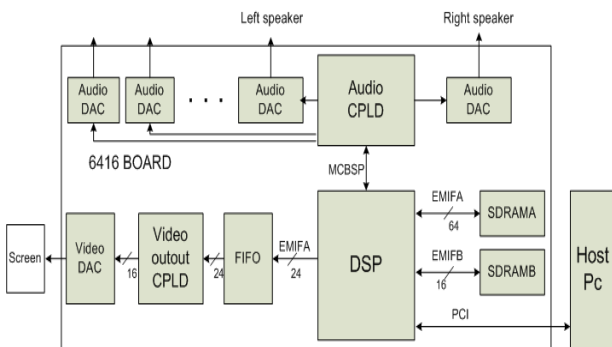


그림 1. DSP 플랫폼 H/W 구성도

위 그림 1은 전체적인 플랫폼 구성도를 나타낸 것이다. 비디오와 오디오의 출력을 위해 고성능의 DAC를 사용하였으며 비디오와 오디오 데이터의 다채널 데이터를 전송, Muxing 그리고 Control을 위한 CPLD를 사용하여 구현하였다. 오디오 CPLD의 구조는 최대 16 channel의 data를 처리할 수 있다. DSP의 EDMA를 통한 블록단위의 전송을 위해 고속, 고용량의 FIFO를 사용하여 고용량, 고화질의 비디오 출력이 가능하면서 오디오의 출력과 오디오/비디오의 디코딩을 동시에 구현하였다.

III. 멀티미디어 Streaming 구조 및 버퍼 Control

3.1 멀티미디어 Streaming 구조

멀티미디어 Streaming 구조는 최초 DSP 플랫폼과

호스트 PC 간의 메시지 통신을 통해 DSP 초기화, DSP configuration download, 그리고 오디오/비디오 라이브러리 초기화순으로 실행한다. 호스트PC가 PCI를 통해서 오디오/비디오 데이터가 포함된 MPEG-2 Program Stream(PS)를 DSP 플랫폼에 전송한다. 이를 구현하기 위해서 API 기반의 전송 프로그램을 별도로 개발하였다. 호스트 PC에서는 일정한 시간(200ms)마다 DSP 플랫폼의 외부 메모리(SDRAM)에 접근하여 일정한 크기의 PS 데이터를 쓴다. 이 때 DSP의 EDMA를 이용하기 때문에 DSP instruction 실행에는 영향을 주지 않는다.[4]

3.2 HD급 Streaming의 디코딩 구조

PCI를 통해서 PS 데이터를 쓰면 인터럽트가 발생하고 Demuxer 루틴을 호출하여 가장 오래된 PS를 읽어 오디오와 비디오의 데이터를 각각의 버퍼에 저장한다. 비디오와 오디오 디코딩은 Timer Interrupt를 사용한다. 35ms(29~30frame/s)마다 Interrupt가 걸리면 비디오 버퍼에서 데이터를 읽어 1 frame 디코딩을 수행하고, 디코딩된 결과인 RGB 데이터를 FIFO로 출력한다. Interrupt가 걸리면 오디오 데이터도 디코딩을 실행한다. 오디오 raw 버퍼가 언더플로우 또는 오버플로우 상황이 아니라면 1 frame 디코딩을 해서 오디오 raw 버퍼에 저장한다. 그리고 EDMA(MCBSP channel) interrupt가 발생하면 디코딩된 오디오 데이터를 오디오 출력 버퍼로 전송한다.

기본적으로, 콘텐츠 Streaming에 맞는 디코더 프로그램을 DSP configuration 파일로 생성한다. 그리고 이 파일을 DSP 플랫폼에 다운로드 함으로써 여러 가지 콘텐츠를 처리할 수 있는 이점이 있고 또한 Demuxer, 오디오 디코더, 비디오 디코더 구조를 같이 쓸 수 있기 때문에 H/W 재구성의 장점이 있다. 다음 그림 2는 system 디코딩 구조를 나타내었다.

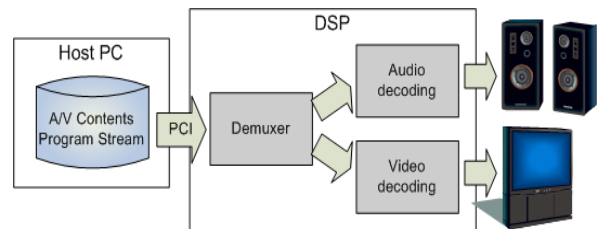


그림 2. 디코딩을 위한 System 구성도

3.3 버퍼 Control 기법 연구

본 논문에서의 제안한 스트리밍 환경은 DSP 플랫폼의 데이터 상황을 고려하지 않고 일정한 시간마다 데

이터를 전송한다. 또한 비디오와 오디오의 디코딩에 필요한 데이터 크기가 가변적이기 때문에 버퍼의 언더런 또는 오버런이 발생할 수 있다. 이러한 제약을 극복하기 위하여 오디오와 비디오의 디코딩은 순차적으로 하지 않고, 버퍼의 포인터를 기준으로 언더런과 오버런을 판단하여 디코딩과 출력의 skip/wait 기법을 제안하였다. 아래 그림. 3은 본 논문에서 제안한 DSP 플랫폼 S/W의 전체적인 버퍼 구조를 나타내었다.

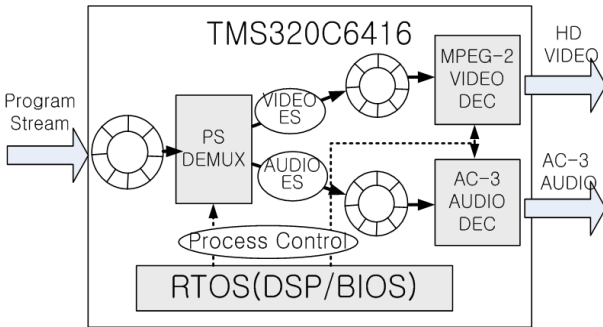


그림. 3 DSP S/W의 Process block diagram

DSP 플랫폼 프로그램에서 사용하는 PS 데이터 버퍼, 오디오/비디오 버퍼, 디코딩 된 데이터를 저장하는 버퍼들은 모두 circular 버퍼이다. 각 버퍼들은 write/read pointer, gap 변수를 가지고 있다. 버퍼에 데이터를 시점에서 쓴 양 만큼 write pointer와 gap 변수를 증가시킨다. 처리하는 시점에서 읽은 양 만큼 read pointer를 증가시키고 gap 변수를 감소시킨다. 버퍼의 마지막 부분을 접근한 후에는 이 포인터 값들을 버퍼의 처음 주소 값으로 지정한다. write/read pointer를 이용하여 항상 버퍼 이외의 메모리영역을 접근하지 못하게 하여 일정한 크기의 메모리 공간만을 하여 지속적으로 발생하는 데이터를 처리할 수 있다. 버퍼를 접근할 때마다 변수 gap을 항상 체크하여 gap이 음수가 되었다면 언더런이 발생한 것이고 버퍼크기보다 크게 되었다면 오버런이 발생한 것을 알 수 있다. read/write pointer를 항상 업데이트하여 중복되거나 끊김 없는 연속적인 데이터의 접근을 확보할 수 있다.

그림. 4는 버퍼 오버런 및 버퍼 언더런을 나타내었다. 이 부분은 오디오와 비디오 디코딩 루틴에서 판단한다. 첫 번째 경우는 버퍼 언더런이 발생할 경우이다. 버퍼 언더런이 발생하기 전에, gap이 일정한 margin보다 작아지면 데이터가 부족한 경우이다. 이전에 출력한 프레임에 다시 출력하여 gap 변수가 margin 보다 커질 때까지 wait해야 한다. 오디오와 비디오는 디코딩을 하지 않고 넘어간다. 그러므로 비디오, 오디오는 정지된 상태이다. 디코딩을 하지 않고 계속 데이터를 저장해서 gap이 margin보다 커지면 이어서 디코딩을

수행한다. 다른 한 경우는 버퍼 오버런이 발생할 경우이다. 버퍼 오버런이 발생하기 전에, gap이 일정한 margin보다 커진다면 사용해야 할 데이터를 훼손할 수 있기 때문에 남아 있던 데이터들을 쓰는 속도보다 빨리 소비해야 한다. 비디오의 경우는 최대한의 화면의 깨짐을 막기 위하여 다음 I-FRAME으로 skip 하여 디코딩하고 오디오의 경우는 다음 프레임으로 skip 하여 디코딩한다. [5]

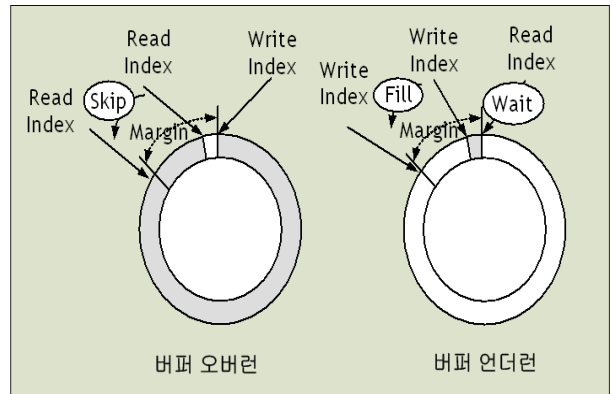


그림. 4 버퍼 오버런/언더런에 따른 skip/wait 기법

VI. DSP 플랫폼을 위한 테스트

MPEG-2 video MP@ML 과 48kHz 5.1ch AC3(5.1ch 디코딩 후 2ch로 downmix) 가 포함된 MPEG-2 program stream coyote.mov file을 호스트 PC에서 본 논문에서 제안한 DSP 플랫폼으로 스트리밍해서 테스트하였다. TEST 파일은 800MB정도의 크기이며 약 30분 정도의 분량이다. 호스트 PC에서 DSP 플랫폼으로 200ms마다 128KB씩 전송하였다. 스트리밍이 끝날 때까지 버퍼 언더런은 경험하지 않고 버퍼 오버런을 3회 정도 경험하였다. 화면이 깨지는 현상을 최소화하였기 때문에 자연스러운 skip을 보였다. 오디오의 경우도 비디오와 같은 구간에서 버퍼 오버런을 경험하였다. 이는 PS의 비디오와 오디오 데이터의 빈도가 일정하게 되어 있기 때문이다. 화면의 변화가 심한 source를 스트리밍 할 때는 호스트PC에서 보내는 시간을 빨리 해주면 된다. I-frame이 자주 들어가기 때문에 데이터의 소비량이 크다.

본 논문에서 제안한 DSP 플랫폼에서 위에서 언급한 PS 파일로 테스트한 정량적인 결과를 다음 표. 1에서 나타내었다. 소요되는 시간을 측정하여 DSP성능 내에서 실행되는지 검증하였다. Interrupt가 걸리는 시간 내에 디코딩에 필요한 모든 일을 끝내고 있음을 보여주고 있다. DSP는 대부분의 시간을 비디오 디코딩하는데 소비한다.

표. 1 기능 블록별로 처리되는 시간

기능 블록	소요시간 (평균)	처리 량 (Byte, 평균)
PS data writing to DSP	1ms	128K
demuxer/frame	2ms	128K
video decoding/frame	20ms	가변적
audio decoding/frame	6ms	가변적
video output moving to FIFO	1ms	675K
합 계	30ms	

V. 결 론

본 논문에서는 고속의 비디오 데이터 및 다채널 오디오 데이터의 처리를 목적으로 general processor인 DSP를 이용하여 유연 플랫폼 기술에 기반한 하드웨어 및 소프트웨어 구조를 제시하였다. 고속 데이터 전송을 위한 하드웨어 구조를 구현하였으며, 이에 탑재될 소프트웨어 모듈과 구조에 대하여 기술하였다. 호스트 PC와 DSP 플랫폼간의 연동을 통해서 HD급 멀티미디어 스트리밍에 대하여 기술하였다. 일정한 크기의 데이터 스트리밍으로 인한 버퍼의 오버런 또는 언더런을 방지하기 위하여 skip/wait 버퍼 control을 제시하였다. 또한 test source를 사용하여 버퍼 오버런이 발생했을 때 깨지지 않는 자연스러운 skip 및 재생을 확인하였다. 호스트 PC상에서 스트리밍을 위한 프로그램도 API 기반에서 구현하였다. 구현한 하드웨어는 PC와 DSP 보드를 PCI 버스를 이용하여 플랫폼을 구현하였지만, 차후 논문에서는 독립적인 형태의 하드웨어로 PCI bus를 통한 연결이 아닌 호스트 CPU와 DSP가 결합된 진보된 형태의 임베디드 플랫폼을 구현할 예정이다. 이는 유무선 네트워크 환경과 RF 모듈이 통합된 구조로 다양한 고용량의 멀티미디어 콘텐츠를 처리할 수 있는 플랫폼으로의 연구가 진행될 것이다.

참고문헌

- [1] Decina, M. "The Internet revolution: reshaping business for the 21st century," Broadband Switching Systems, 1997. Proceedings. 2nd IEEE International Workshop on , 2-4 Dec. 1997.
- [2] ISO/IEC 13818-2 : 2000 Information technology - Generic coding of moving pictures and associated audio information: Video.
- [3] Texas Instrument, SPRC137, "TMS32064X DSP Reference Design"
- [4] Texas Instrument, SPRA636A, "Application

- Using the TMS320C6000 Enhanced DMA"
- [5] P.N. Tudor "MPEG-2 VIDEO COMPRESSION", IEE J Langham thompson Prize, Electronics & Communication Engineering Journal, Dec. 1995 Tutorial.