DSP 기반 HD급 비디오/오디오 디코더 시스템 개발

박영근, 김봉주, 김병일, 김정근, 장태규, 이전우*
중앙대학교 전자전기공학부, 전자통신연구원*

Development of a DSP based Decoder for High-definition Video/Audio System

Young-Keun Park, Bong-Ju Kim, Byeong-Il Kim, Jung-Keun Kim, Tae-Gyu Chang, Jun-Woo Lee*
School of Electrical and Electronics Engineering, Chung-Ang University, ETRI*

Abstract - 본 논문에서는 HDTV(High Definition TV) 방송수신을 위한 DSP기반의 HD급 비디오/오디오 디코더 시스템을 개발하고 그 성능을 확인하였다. DSP 플랫폼은 TI사의 TMS320C6415를 대상으로 하였으며 TI의 DSP RTOS인 DSP/BIOS를 이용하여 방송스트림인 TS (Transport Stream)을 분리하기 위한 TS Demuxer, MPEG-2 비디오 디코더 및 AC-3 오디오 디코더 알고리즘을 통합하였으며, 각각의 알고리즘은 대상 DSP 플랫폼인 TMS320C64x에 맞게 고정소수점 구조화 및 최적화를 실시하였다. 테스트를 위한 시스템은 스트리밍을 위한 호스트 PC와 PCI버스를 통해 연결된 DSP보드로 구성하였으며 실제 HDTV방송용 스트림과 SD급 스트림을 이용하여 성능을 확인하였다.

1. 서 론

근래 본격적인 HDTV 공중파 방송이 각 방송사들에 의해 시작되고, 방송프로그램 또한 HD급 영상 및 오디오를 기준으로 제작되고 있으며 점점 그 점유율을 높여가고 있다. HDTV방송 규격에는 크게 유럽방식 DVB (Digital Video Broadcast)와 미국방식의 ATSC방식이 있는데, 우리나라의 경우 미국방식인 ATSC방식을 채택하고 있으며 비디오는 MPEG-2[3]를, 오디오는 AC-3[4]를 사용한다[1]. 현재 HDTV의 수신이 가능한 셋탑박스(Settop box)가 다수 출시되어 있지만 이들 대부분은 HD급 비디오/오디오 디코딩을 위해 ASIC기반의 디코딩칩을 사용하고 있다. 아직까지는 DSP (Digital Signal Processor)에 비해 값이 싸고 성능면에서 우위를 보이고 있는 ASIC기반의 디코더칩이 많이 사용되고 있지만 ASIC기반의 디코딩칩은 추후 알고리즘의 변경이나 재구성이 불가능하다는 단점이 존재한다. 이 반면에 DSP기반으로 디코딩 알고리즘을 구성할 경우 알고리즘의 모듈화가 가능해 알고리즘의 선택적 구성 및 추후 재구성이 가능하며 알고리즘의 변경 및 유지보수가 용이하다는 장점이 있다. 현재 알고리즘을 DSP에 소프트웨어적으로 구현하는데있어 걸림돌은 ASIC구현에 비해 상대적으로 비용이 많이들고 성능이 떨어지는 문제점이다. 하지만 최근 DSP기술의 급속한 발달로 인해 그러한 문제점들은 점차 해결되어지고 있다. 이에 본 논문에서는 HDTV방송 수신에 필요한 HD급 비디오/오디오 디코더 시스템을 DSP상에 구현하여 그 성능을 확인하고 현시점에서의 가능성을 알아보고자 한다.

2. 본 론

2.1 DSP기반 HD급 비디오/오디오 디코더 시스템

본 절에서는 본 논문에서 구현한 DSP기반의 HD급 비디오 및 오디오를 위한 디코더 시스템의 구성에 대하여 기술하였다.

그림. 1은 본 논문에서 구현한 Texas Instrument사의 TMS320C6415 DSP기반의 HD급 비디오/오디오 디코더 시스템의 구성을 나타낸 것이다.

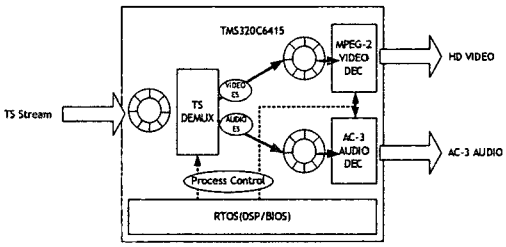
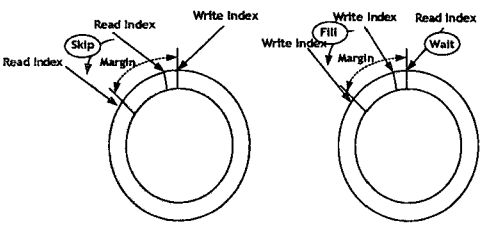


그림. 1 TMS320C6415기반 HD급 비디오/오디오 디코더 시스템 구성도

DSP상에 구현된 디코더 시스템은 전송된 TS (Transport Stream)스트림을 MPEG-2 비디오 스트림과 AC-3 오디오 스트림으로 분리하여 주는 TS Demuxer와 분리된 비디오 및 오디오스트림을 디코딩하기 위한 MPEG-2 비디오 디코더, AC-3오디오 디코더와 같이 크게 세부분으로 구성된다. 이 알고리즘들은 RTOS인 DSP/BIOS상에서 각각의 동작이 컨트롤되게 된다. 그리고 원활한 디코딩을 위한 스트림버퍼가 TS Demuxer, MPEG-2 비디오 디코더, AC-3 오디오 디코더 앞단에 위치하게 된다. 방송의 특성상 공중파를 통해 전달되는 TS 스트림은 항상 일정한 비트율로 수신기에 전달되게 된다. 하지만 시스템상의 문제 혹은 TS 스트림 인코딩 시스템과 디코딩 시스템간의 시간적 불일치 같은 요인들로 인해 전달되는 스트림의 속도보다 디코딩 속도가 느린 경우 또는 전달되는 스트림의 속도보다 디코딩속도가 빠른 경우에 스트림버퍼는 오버런(Over Run)상태가 되거나 언더런(Under Run)상태가 되게 된다. 이러한 상황을 막기위해서는 스트림 버퍼를 적절히 컨트롤하여야 한다. 아래의 그림. 2는 두가지 경우에서 스트림버퍼를 컨트롤하는 방법을 보여준다.



(a) 오버런의 경우 (b) 언더런의 경우

그림. 2 원활한 디코딩을 위한 스트림버퍼 컨트롤

(a)의 경우는 디코딩 또는 Demuxing 속도가 전달되는 스트림 속도에 비해 느릴 경우로 스트림데이터가 덮어쓰여지게 되면 올바른 디코딩이 되지 않거나 스트림버퍼의 크기 전체에 해당하는 데이터가 손상되어 그만큼을 생략하게 되는 결과를 초래하게 된다. 그러므로 Wirte Index와 Read Index사이에 일정한 크기의 여유를 두

어 둘사이의 간격이 이 여유치보다 작아졌을 경우 여유치를 다시 유지하도록 그에 상당하는 스트림데이터를 건너뛰게 한다. (b)의 경우는 이와는 반대의 경우로 디코딩 또는 Demuxing속도가 전달되는 스트림 속도보다 빨라 버퍼에 충분한 스트림데이터를 유지하지 못하는 경우로 이때는 디코딩 혹은 Demuxing을 버퍼에 설정된 여유치 이상이 찰 때까지 기다려야 한다. 이 때 디코더의 출력은 이전 정상적인 프레임출력을 반복해서 내보내는 식으로 처리 할 수 가 있다. 여유치이상이 다시 차게 되면 다시 디코딩 혹은 Demuxing이 시작되게 된다.

디코딩에 사용되는 스트림데이터는 호스트PC로부터 PCI버스를 통해 실시간 스트리밍을 통해 DSP보드로 전달되게 된다. 그림. 3은 PCI버스를 통한 실시간 스트리밍구조를 나타낸 것이다. PCI버스를 통한 데이터의 전송을 위해 DSP의 메모리맵상에 위치한 디스크립터 테이블을 이용하게 된다. PCI버스를 통해 데이터를 전송할때는 DSP를 기준으로 Slave전송 및 Master전송이 가능하다. 하지만 전달된 데이터를 하드웨어 인터럽트를 통해 처리하기 위해서는 Master전송을 이용하여야 한다. Master전송은 PCI버스를 통한 데이터의 전송이 끝났음을 알리는 하드웨어 인터럽트인 MASTEROK를 이용하게 된다. 대응되는 인터럽트 서비스인 INT13 ISR에 전송된 스트림데이터의 처리를 위한 코드를 구성함으로써 PCI를 통한 실시간 스트리밍이 이루어지게 된다.

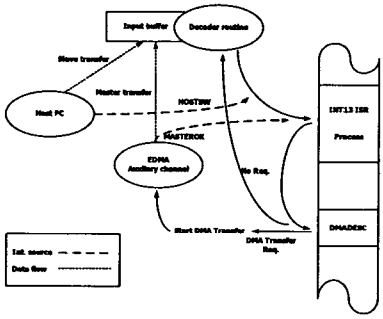


그림. 3 PCI버스를 통한 실시간 스트리밍 구조

2.2 알고리즘의 최적화

본 절에서는 TMS320C6415 DSP상에서 구현된 TS Demuxer, MPEG-2 비디오 디코더 및 AC-3 오디오 디코더의 최적화를 위한 기법을 설명하였다.

그림.4 는 TMS320C64x DSP 코어의 VLIW(Very Long Instruction Word)구조인 VelociTi.2를 나타낸 것이다.

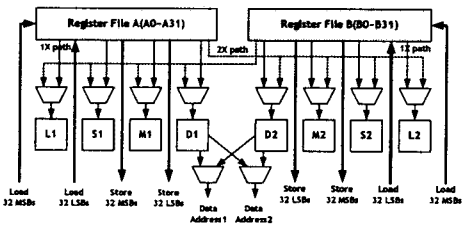


그림. 4 TMS320C64x코어의 VelociTI.2 구조

VLIW구조의 장점은 서로 독립적으로 수행이 가능한 여러개의 명령어를 모아 동시에 병렬수행함으로써 효율적이고 빠른 연산을 가능하게 하는 것이다. TMS320C64x의 VelociTI.2코어는 대칭형 데이터경로와 연결된 2쌍의 기능유닛 블록, 그와 대응되는 32개의 32비트 레지스터로 이루어진 2개의 레지스터 파일로 구성되어 있다. 각 기능유닛 블록은 4개의 기능유닛을 가지고 있고, 각 기능유닛은 논리연산/대수연산/비교, 쉬프트/대수연산/분기, 곱셉, 데이터 어드레스 계산의 역할을 한다. 기능유닛블록은 대응되는 레지스터파일에 대해서는 1사이클에, 반대쪽 레지스터파일에 대해서는 2사이클에 접근할 수 있다. 반대쪽 레지스터 파일 자원을 사용하게 됨으로써 명령어 패치패킷(Fetch pecket)사이클에서 더욱 원할한 병렬연산이 가능하다[2]. 이러한 TSM320C64x의 구조를 활용하여 알고리즘의 최적화를 위해 다음과 같은 사항을 고려하였다.

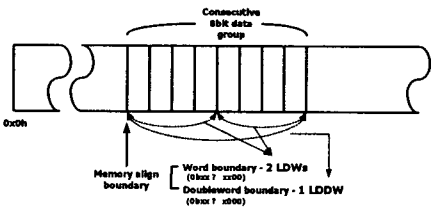


그림. 5 메모리정렬을 통한 최적화

첫째, 메모리와 관련된 사항으로 TMS320C 64x계열은 바이트, 하프워드(2바이트), 워드(4바이트)단위의 Load/Store명령어는 물론 메모리경계에 맞추어 2워드를 처리할 수 있는 정렬된 더블워드 처리명령어 및 비정렬된 더블워드처리 명령어를 지원한다. 이 들 명령어를 활용하여 처리 메모리 대역폭을 확보하고 데이터의 메모리정렬을 통해 추가적인 사이클의 소비를 방지할 수 있다. 그림. 5는 메모리정렬을 통한 알고리즘 최적화의 예를 보여준다. 그림에서는 워드단위 또는 더블워드단위의 데이터를 처리할 경우 처리할 데이터 단위의 메모리 경계에 정렬을 함으로써 한사이클에 효과적으로 처리하게 된다. 그리고 시스템 내부메모리와 외부메모리간의 적절한 분배를 통해 중요도가 높은 코드는 내부 메모리를, 중요도가 낮은 코드는 외부 메모리를 활용함으로써 성능저하를 최대한 억제하였다. 또한 TMS320C64x의 2레벨 메모리를 효과적으로 사용할 수 있는 캐쉬(Cache) 시스템의 활용을 통해 디코더의 성능저하를 막았다. 이에 부가적으로 EDMA채널을 이용한 백그라운드 메모리 전송을 활용하였다.

둘째, 알고리즘 코드 및 컴파일러관련 사항으로 C코드로 이루어진 알고리즘 코드는 컴파일러의 최적화 옵션을 통하여 대략 30~40%정도의 성능향상을 이룰 수 있지만 그 이상의 최적화를 위해서는 어셈블리와 일대일로 대응되는 Intrinsic의 활용 및 어셈블리레벨의 코드최적화가 이루어져야한다. 성능에 큰 영향을 주는 부분은 Intrinsic 및 직접 어셈블리 코딩을 통해 최적의 성능을 이끌어 낼 수 있도록 하였다.

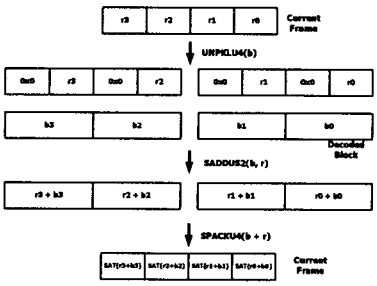


그림. 6 Addblock에서의 Packed Operation의 예

셋째, 명령어의 레이턴시관련 사항으로 대부분의 64x 명령어의 경우 1사이클의 레이턴시를, 곱셉명령어의 경우에는 2사이클의 레이턴시가 생기게 된다. Load/Store 명령어나 분기명령어의 경우는 5사이클에서 6사

이클의 레이턴시를 가진다. 이러한 각 명령어의 레이턴시를 고려하여 패치패킷(Fetch pecket)에 적절히 분배함으로써 VelociTI.2의 구조를 효율적으로 활용하는 코드를 구성하였다. 또한 그림.6에서처럼 TMS320C64x에서 멀티미디어의 효율적 처리를 위해 제공하는 Packed Operation 명령어를 적절히 사용하여 데이터 처리에 다수의 사이클이 걸리는 것을 단 한사이클로 줄임으로써 알고리즘의 성능을 최적화하였다.

마지막으로, 메모리 뱅크 충돌관련 사항으로 TMS320C6000DSP는 4개의 하프워드 크기의 Interleaved 메모리뱅크로 구성된 메모리 구조를 사용한다. 만약 동일 메모리뱅크에 대해 동시에 메모리처리 명령어를 수행하게 되면 1사이클의 딜레이를 발생시키는 스톨(Stall)현상이 발생하게 된다. 이러한 스톨은 알고리즘의 성능에 큰 영향을 주므로 메모리 뱅크 충돌을 피할 수 있도록 코드를 최적화하여 구성하였다.

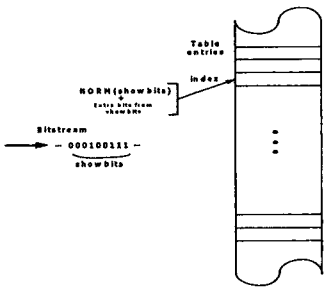


그림. 7 NORM을 이용한 VLC(Variable Length Code)테이블의 참조

이에 더불어 VLD(Variable Lenth Decoding)과 같은 큰 연산량이 필요한 부분은 NORM과 같은 특수 명령어를 사용하여 그림.7처럼 효율적인 테이블참조구조를 사용함으로써 연산량을 최소화하여 디코더 알고리즘을 최적화하였다.

2.3 테스트 시스템 구성 및 성능분석

본 절에서는 구현된 DSP기반의 HD급 비디오/오디오 디코더 시스템의 성능을 측정하기 위한 테스트 시스템의 구성 및 디코더 시스템의 성능에 대해 기술하였다.

그림.8은 테스트 시스템을 나타낸 것으로 실시간 스트림데이터는 호스트PC의 HDD에 저장된 실제 방송용 TS 스트림을 PCI버스를 통해 DSP보드로 전송하게 되며, DSP에 구현된 TS Demuxer를 통해 비디오와 오디오 스트림으로 분리되어 MPEG-2 비디오 및 AC-3 오디오 디코더로 전달되고, 디코딩된 결과를 모니터 및 스피커를 통해 확인하게 된다.

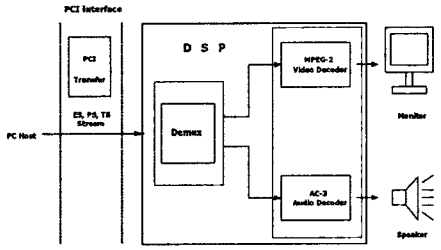


그림.8 디코더 시스템의 테스트를 위한 구성

실제 구현에 사용된 보드는 ATEME사의 IEKC64 이미지 프로세싱보드로 외부 메모리는 SDRAMB 8MB와 SDRAMA 256MB로 구성되어 있다. 표.1에 구현된 디코더 시스템의 메모리 소요량을 나타내었다. AC-3 오디오 디코더의 코드사이즈가 상대적으로 큰 이유는 고정소수점 구조로 변경후 코드의 최적화가 제대로 이루어지지 않았기 때문이며 heap의 대부분은 버퍼메모리로 사용되었다.

	MPEG-2 비디오 디코더	AC-3 오디오 디코더	TS demuxer
.bss	2B		
.far	8,672B		
.cinint	13,252B		
.const	1,720B		
.switch	96B		
.text	39,936B	45,096B	9,088B
.buffers	1,179,648B		
.table	28B		
.heap	16,777,216B		
total	17.24MB		

표. 1 메모리 소요량

표. 2에 DSP상에 구현된 MPEG-2 비디오 디코더의 성능을 나타내었다. DSP이식을 위해 구성한 Visual C 기반의 C코드기준으로 MPEG표준에 준하는 적합성테스트를 통해 신뢰성을 확인하였으며[3], SD의 경우는 약 40%, HD의 경우는 약 270~300%정도의 로드율을 보이고 있다. AC-3 오디오 디코더의 경우 최적화가 많이 이루어지지 않았지만 약 10%미만의 로드율을 보였다. AC-3 오디오 디코더의 음질은 여러 가지 분석기법을 통해 부동소수점 구조의 디코더와 대등함을 확인하였다[4]. 현 시점에서 TMS320C6415(480MHz)을 기반으로 한 HD급 비디오/오디오 디코더 시스템은 성능 요구량의 약 1/3정도를 보여준다.

	HD stream (1920×1088)			SD stream (720×480)		
bitrate (Mbps)	17.7	17.45	17.55	7.63	9.35	9.54
cycles/frame (profile Avg.)	25,945,945	27,906,976	26,519,337	3,689,469	3,735,356	3,924,775
frame/sec (profile Avg.)	18.5	17.2	18.1	130.1	128.5	122.3
frame/sec (board Avg.)	11.2	10.4	10.1	74.2	71.0	69.6
Load율(%)	267	288	297	40	42	43

※ TMS320C6415(480MHz) 동작기준

표. 2 구현된 MPEG-2 비디오 디코더의 성능

3. 결 론

본 논문에서는 HDTV방송 수신을 위한 DSP기반의 HD급 비디오/오디오 디코더 시스템을 구현하였다. 테스트 결과 현재 최상의 성능을 보여주는 TI의 TMS320C64x 계열 DSP상에 구현된 디코더 시스템은 요구성능의 1/3정도를 보여주었다. 하지만 DSP의 발전추세를 본다면 곧 1GHz의 속도 및 다수의 코어를 이용한 DSP가 개발 및 출시될 것이며 추가적인 최적화작업을 통해 구현된 디코더 알고리즘을 개선한다면 성능은 더욱 향상될 것이고, 기존 ASIC칩 기반의 디코더를 충분히 대체할 수 있을 것이다.

[참 고 문 헌]

[1] Advanced Television Systems Committee, ATSC standard:Digital Television Standard, Revision B, Doc. A/53B, 7 August, 2000

[2] Texas Instruments, TMS320C6000 CPU and Instruction Set Reference Guide, Revision F, SPRU189F, 2000

[3] ISO/IEC 13818-2:2000, Generic coding of moving pictures and associated audio information-Part 2: Video, 2000

[4] Advanced Television Systems Committee, Digital Audio Compression Standard(AC-3), Doc. A/52, 20 Dec. 1995