

분리형 구조의 고화질 멀티 포맷 비디오 복호기: MPEG-2/MPEG-4/H.264와 VC-1

배 종 우[†] · 조 진 수^{††}

요 약

본 논문에서는 MPEG-2, MPEG-4, H.264/AVC 및 VC-1 코덱 표준을 동시에 지원하는 멀티 포맷 비디오 복호기 (MFD)의 설계 방법을 제안한다. 제안하는 MFD는 디지털-TV SoC 에 필요한 고사양의 고화질급 비디오 처리를 목표로 하였다. 리스크 프로세서, 온칩 메모리 및 주변 회로 등의 크기가 큰 공용 자원들을 공유하여 크기를 최소화 하였다. 또한, 코덱 별로 추가 및 제거가 용이한 분리 가능한 구조를 사용하였다. 이러한 구조는 이미 설계되고 검증된 코덱의 안정성의 유지를 용이하게 해준다. 설계된 MFD는 65nm 공정에서 크기가 약 2.4M 게이트이며, 동작속도는 225MHz이다. 본 논문에서 제안한 MFD는 현재까지 알려진 MFD 중 최고 성능인 고화질급(1080p@30fps) 이상의 비디오 디코딩을 지원하며, 가장 많은 종류의 비디오 코덱 표준을 지원한다.

키워드 : 멀티 포맷 비디오 복호기, H.264, VC-1, 디지털 TV SoC

A Detachable Full-HD Multi-Format Video Decoder: MPEG-2/MPEG-4/H.264, and VC-1

Bae, Jongwoo[†] · Cho, Jinsoo^{††}

ABSTRACT

In this paper, we propose the VLSI design of Multi-Format Video Decoder (MFD) to support video codec standards such as MPEG-2, MPEG-4, H.264 and VC-1. The target of the proposed MFD is the Full HD (High Definition) video processing needed for the high-end D-TV SoC (System-on-Chip). The size of the design is reduced by sharing the common large-size resources such as the RISC processor and the on-chip memory among the different codecs. In addition, a detachable architecture is introduced in order to easily add or remove the codecs. The detachable architecture preserves the stability of the previously designed and verified codecs. The size of the design is about 2.4 M gates and the operating clock frequency is 225MHz in the Samsung 65nm process. The proposed MFD supports more than Full-HD (1080p@30fps) video decoding, and the largest number of video codec standards known so far.

Keyword : Multi-Format Video Decoder, H.264, VC-1, D-TV SoC

1. 서 론

최근 D-TV나 휴대폰과 같은 가전시장에서 기존의 MPEG-2 코덱 만을 지원하는 것 이외에 다양한 비디오 코덱 들을 지원해야 할 필요성이 빠르게 확산되고 있다. H.264/AVC(이하 H.264)가 국내의 지상파 DMB, 유럽의 위성방송 및 DVB-H 등에도 쓰이고 있으며, 점차 그 시장을 확대해 가고 있다. IPTV 또는 여러 웹과 모바일 콘텐츠를 지원하기 위하여 MPEG-4 계열의 코덱들인 DivX 와 XviD 등(이하 MPEG-4로 통칭)의 지원도 필요로 하고 있다. 특히 최근까지는 표준 화질급 (480p@30fps) 콘텐츠가 주류를 이

루었으나 디스플레이 기기가 대형화 되어가는 추세에 맞추어 이들 콘텐츠도 고화질급으로 점차 바뀌어 가고 있다. 또한 WMV9 이라는 이름으로도 불리고 있는 VC-1은 H.264와 유사한 성능을 보이는 최신 코덱으로서 차세대 DVD 표준인 HD-DVD와 블루레이 디스크에서 MPEG-2 및 H.264와 함께 반드시 지원해야 하는 코덱으로 선정되었다.

위와 같은 코덱들을 지원하는 다양한 하드웨어와 소프트웨어 IP들이 개발되어 왔다. MPEG-2는 오랜 기간동안 시장에서의 주류 코덱으로 자리를 잡아 왔고, 최근 몇 년 사이 MPEG-4, H.264 및 VC-1 코덱을 위한 여러 솔루션들이 등장하기 시작했다. 특히 하드웨어 IP를 만드는데 있어서는 이러한 개별 코덱들을 따로따로 제작하여 하나의 SoC 에 집약시켜 사용하는 것은 크기 면에서 상당한 오버헤드를 가지게 된다. 이러한 문제점을 해결하기 위해서 여러 코덱들

[†] 정 회 원 : 명지대학교 정보공학과 조교수

^{††} 정 회 원 : 경원대학교 소프트웨어공학부 전임강사(교신저자)
논문접수: 2007년 11월 20일, 심사완료: 2008년 1월 26일

을 효과적으로 융합하여 크기가 작은 하나의 코덱 IP 로 만들고자 하는 다양한 시도들이 있어 왔다[1-3]. 이와 같은 것들을 본 논문에서는 멀티 포맷 비디오 코덱이라 칭하며, 복호기의 경우는 멀티 포맷 비디오 복호기(MFD)라 칭한다.

본 논문에서는 이러한 MFD 로써 MPEG-2, MPEG-4 계열(DivX 및 XviD 포함), H.264 및 VC-1 코덱 표준을 동시에 지원하며, 고화질급(1920x1088@30fps) 이상의 디코딩을 지원하는 IP 설계방법을 제안한다. 지원 사양은 MPEG-2 메인 프로파일 하이 레벨(MPEG-1 포함), MPEG-4 심플과 어드밴스드 심플 프로파일 레벨 5, DivX 홈씨어터 프로파일 버전 3.xx-6.xx, H.264 하이 프로파일 레벨 4.2, VC-1 어드밴스드 프로파일 레벨 4.0 및 모든 하위 프로파일과 레벨들이다. MFD 설계 시 크기를 최소화하기 위해서 하위 레벨의 논리회로까지 모든 부분을 융합시키게 되면 설계의 복잡도 증가와 그로 인한 설계시간의 증가, 에러 발생의 가능성 증가 및 검증의 어려움이 발생한다. 또한 시장상황에 따라 요구되는 코덱들의 종류가 바뀌어 코덱들을 추가하거나 혹은 제거해야 할 필요가 있는 경우 하위 레벨 논리회로까지의 융합설계는 그다지 바람직한 방향이 아니다. 따라서 제안된 MFD에서는 우선 리스크(RISC) 프로세서, 온칩 메모리 및 외부 인터페이스와 같은 전체의 반 이상을 차지하는 크기가 큰 자원들을 코덱 간에 공유하여 MFD의 크기를 축소시킨다. 그 외의 매크로블록 레벨의 디코딩을 담당하는 하드웨어 로직 부분은 꼭 공유가 필요한 부분을 제외하고는 새로운 코덱의 추가나 기존 코덱의 제거가 용이하도록 서로 분리 가능한 방식으로 설계한다.

본 논문의 구성은 다음과 같다. 먼저 2장에서는 이전까지 발표된 유사한 내용의 논문들을 소개하고 제안된 설계와 비교를 한다. 3장에서는 제안하는 MFD의 설계방법을 상세히 소개하며, 4장에서는 제안된 MFD의 크기 및 성능을 요약하여 설명한다. 5장은 본 논문을 결론짓는다.

2. 관련 연구

이전에 발표된 MFD 구조 중 가장 최근 것이며 구조가 유사한 것은 Chien[1]의 멀티-스탠다드 멀티 채널 비디오 복호기로서 고화질급 영상을 처리할 수 있는 MPEG-2/4 와 H.264 MFD이다. 본 논문에서 제안하는 MFD 와는 VC-1이 포함되어 있지 않다는 큰 차이가 있다. VC-1은 H.264와 비슷한 복잡도를 가지는 최신 코덱으로서 VC-1 코덱의 유무는 구현의 복잡도에 있어서 상당한 차이를 가져다 준다. 또한 지원하는 프로파일들은 MPEG-2/4의 경우 심플 프로파일이며 H.264는 베이스라인 프로파일이다. 본 논문에서 제안하는 MFD가 지원하는 프로파일들은 각 코덱의 최상위 프로파일들인 VC-1의 어드밴스드 프로파일과 H.264의 하이 프로파일 이며, 지원 수준은 고화질급 이상이라는 것에서 훨씬 복잡도가 높은 것이라 하겠다. 특히 H.264의 경우 베이스라인과 하이 프로파일 은 CABAD, B-slice, MBAFF 모드 등 크기, 성능 및 메모리 대역폭에서 매우 큰 차이를 보

일 수 있는 상이한 프로파일들이라 할 수 있다. MPEG-4의 심플과 어드밴스드 심플 프로파일도 상당한 차이점을 가진다. 그리고 설계의 개념 자체도 코덱들이 완전히 융합된 최소화를 추구하느냐 아니면 본 논문에서 제안하는 설계처럼 분리 가능한 구조를 가지면서 크기의 소형화를 추구하느냐도 설계 개념의 큰 차이점이 된다.

Hase [2]는 본 논문에서 제안하는 MFD와 유사한 H.264/VC-1/MPEG-4의 부호기 및 복호기를 제안하였는데, 코덱들 간의 하드웨어 자원을 공유하는 방식으로 그 크기를 최적화 하였다. 또한 저전력 설계에 주력하였으며, 지원하는 비디오 포맷은 표준화질급이라는 점에서 본 논문에서 제안하는 MFD가 지원하는 고화질급 이상의 비디오와는 계산량에 있어서 매우 큰 차이가 난다. 마지막으로 제안하는 MFD의 코덱 별로 분리 가능한 구조와는 설계 개념에 있어서 상당한 차이점을 가지고 있다.

Liu[3]는 MPEG-2와 H.264의 베이스라인 프로파일에 대해서 저전력을 위한 설계방법을 제안하였다. 본 논문에서 제안하는 MFD는 전력 보다는 성능과 크기 그리고 분리 가능한 구조를 강조한 것, 그리고 지원 비디오 포맷이 고화질급 이상이라는 것 등에서 많은 차이가 있다. H.264만의 구현에 관련된 연구들도 최근에 여러 가지가 소개되어 있어 [4-7] H.264의 높아진 관심을 보여주고 있다. 본 논문에서 제안하는 MFD는 고화질급 이상의 비디오 포맷을 지원하며 분리 가능한 구조를 가짐으로 H.264를 구현한 연구들과 직접적 비교가 용이하지는 않으나 참고할만하다. Bae[7,8]의 연구들은 본 MFD 이전에 개발하였던 H.264 의 구조에 대한 연구들로서 본 논문에서 제안하는 MFD 구현에 기초가 되었다.

3. 분리형 구조의 MFD

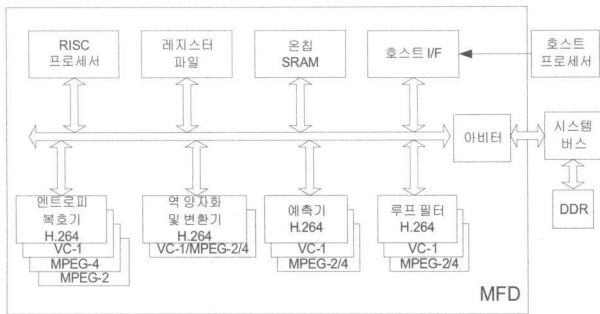
3.1 전체적인 구조와 동작

제안된 MFD의 구조는 (그림 1)과 같다. 펌웨어에 의해 비디오 스트림의 슬라이스 수준 분석까지 담당하는 임베디드 리스크 프로세서, 레지스터 파일, 온칩 SRAM 등의 내부 메모리, 외부 호스트 프로세서와 인터페이스를 하는 호스트 I/F, 외부 버스와 인터페이스를 하는 아비터(arbiter) 부분, 그리고 매크로 블록 레벨에서의 복호화 과정을 담당하는 하드웨어 로직으로 구성된 모듈들인 엔트로피 복호기, 역 양자화/변환, 예측기 및 루프 필터로 구성되어 있다.

MFD의 동작은 다음과 같다. 먼저 외부의 호스트 프로세서로부터 시작신호가 들어오면 호스트 I/F를 통하여 지정된 레지스터에 값이 저장된다. 리스크 프로세서는 시작신호를 확인하여 비디오 스트림을 읽어 들인 후 외부 메모리에 저장한다. 저장된 데이터는 다시 리스크 프로세서에서 읽어 들여 슬라이스 수준까지의 분석을 진행한다. 그 결과가 레지스터 파일과 온칩 SRAM에 저장되고, 매크로블록 레벨부 터의 복호화는 엔트로피 복호기, 역 양자화/변환, 예측기 및 루프 필터의 순서를 거쳐서 파이프라인 되면서 실행된다.

최종 결과는 외부 메모리에 저장되어 출력된다.

제안된 MFD에서 상당한 크기를 차지하는 부분들로서 공유화 가능한 부분들은 임베디드 리스크 프로세서, 캐쉬, 레지스터 파일 및 온칩 SRAM 들이다. 그 외의 호스트 I/F 및 아비터 부분은 공통이므로 공유가 가능하다. 이 부분들을 공유화 하도록 설계하고 나머지 각 코덱에 특화된 하드웨어 로직 부분들은 반드시 필요한 부분들을 제외하고는 서로 분리되도록 설계하였다. 본 장에서는 각 모듈 별로 합리적인 MFD 설계 방법을 제안한다.



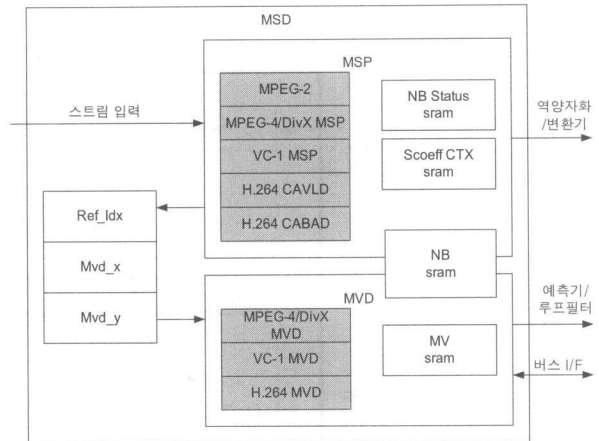
(그림 1) MFD 구조도

3.2 펌웨어

MFD 펌웨어는 내부 코덱 및 외부 인터페이스 관련 초기화를 담당하며, 스트림 입력을 받아 슬라이스 수준까지는 직접 복호화를 하고, 매크로 블록 이하는 하드웨어에 제어를 넘기는 등 전체적인 복호화 과정에 대한 제어를 한다. MFD의 펌웨어는 시스템의 성능이 가능한 한도 내에서 동시에 임의의 개수의 스트림을 처리할 수 있으며, 각각의 스트림 별로 코덱을 결정하게 된다. 예를 들면, MPEG-4 스트림 과 H.264 스트림을 동시에 복호화 할 수도 있고, H.264 스트림 두 개를 동시에 복호화 할 수도 있다. 동시에 두 개의 스트림을 복호화 하려면, 호스트에서 두 개의 스트림에 대한 스트림 입력 포트에 각각 입력을 주어야 하며, MFD 펌웨어는 단위 시간에 각 코덱 별로 한 프레임씩 지속적으로 복호화한다. 디코딩 과정 중 하드웨어에의 작업이 진행되는 동안, 펌웨어는 호스트와의 통신을 통해 필요한 정보들을 상호 교환함으로써, 호스트에서 올바르게 MFD를 제어할 수 있는 방법을 제공한다.

3.3 매크로 블록 구문 복호기

대부분의 코덱은 시퀀스, 픽처, 슬라이스 수준 구문이 복호화 되면 매크로 블록 수준에서 복호화를 시작 할 수 있다. 매크로 블록 복호화는 매크로 블록 수준에서 필요한 구문을 압축된 스트림으로부터 분석하고, 분석된 정보를 약간 가공한 후 역 양자화/변환, 예측, 루프 필터 과정을 거쳐서 하나의 매크로 블록 픽셀을 구하게 된다. MSD(매크로 블록 구문 복호기)는 복호화 과정 중 매크로 블록 수준 분석을 하고 잔류 데이터와 움직임 벡터를 생성해 주는 모듈이다. (그림 2)가 MSD의 구조를 보여준다. MSP가 구문 분석을 하고

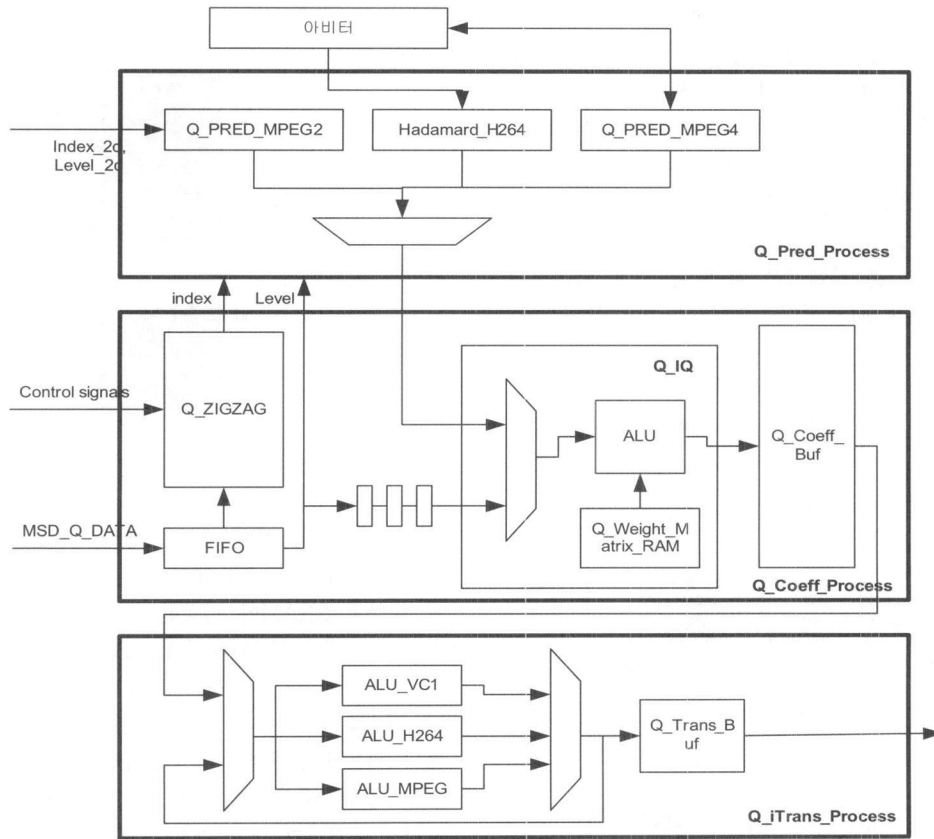


(그림 2) 매크로 블록 구문 복호기 구조

MVD가 움직임 벡터 생성을 한다. MSP로 스트림이 입력되면, 각 코덱 별로 하드웨어 로직이 구문분석을 실행하며 이때 내부의 SRAM에 저장된 테이블을 사용한다. 구문을 분석하는 방법은 H.264의 CABAD만 산술 복호화를 사용하고 MPEG-2, VC-1, MPEG-4 모두 가변 길이 복호화를 사용한다. H.264의 지수 곱셈(Golomb) 코딩도 약간의 차이는 있으나 하드웨어 구현은 더 용이하다. 계수를 구하는 방법은 모두 다 런-레벨 복호화나 그것이 약간 수정된 방법이 사용된다. 그 결과가 역 양자화/변환기로 전송된다. 또한 움직임 벡터 생성을 위한 일부 정보들이 Ref_idx, Mvd_x와 Mvd_y 등의 FIFO를 통하여 MVD에 전달되어 움직임 벡터가 생성되고 이는 예측기와 루프 필터로 전달된다. MSP는 MV SRAM을 계산 중에 활용하며 MVD와 NB SRAM을 공유한다.

움직임 벡터를 구하는 방법은 MPEG-2는 이전 움직임 벡터 예측기를 이용해서 간단히 구해진다. 그리고 MPEG-2를 제외한 나머지인 MPEG-4계열과 H.264 및 VC-1은 매크로 블록당 움직임 벡터의 개수, 순차/비월 주사, P/B 픽처, 다이렉트 모드에 따라 움직임 벡터가 구해진다. 그 각각의 코덱들도 움직임 벡터 구하는 방법이 서로 다르다. 비월 주사 관련 방법을 예를 들면, MPEG-4 계열은 비월 주사 프레임만 존재한다. VC-1은 비월 주사 프레임과 필드에 따라 다르게 구한다. H.264는 MBAFF(매크로 블록 적응적 필드 프레임)이라는 모드가 있어 두 매크로 블록을 한 세트로서 움직임 벡터를 구하는 것을 요구하기도 한다.

앞서 언급되었듯이 분리 가능한 구조로 가져가기 위해서 MSD도 각 코덱은 별도의 논리회로로 구현되었으며, 메모리와 일부 논리회로만 공유 되었다. CABAC을 제외하고는 모두 유사한 가변 길이 복호기 계열이지만 제어부의 유한상태기는 크기가 작고 테이블은 모두 다르기 때문에 논리회로를 같게 설계한다면 너무 복잡해서 크기를 줄이는 효과는 극히 제한되고 오류 발생 가능성만 높아지므로 현재의 구조를 선택했다. MVD도 마찬가지로 복잡도가 증가하는 것에 비해서 크기를 줄이는 효과는 미미하다. 따라서 논리회로는 별도로 하고 메모리와 곱셈기를 공유해서 만들었다.



(그림 3) 역 양자화/변환기 구조

MSD는 논리회로 266K 게이트와 메모리 4.76KB로 구현되었다. 이것은 전체의 14% 정도의 크기이며, 공유된 테이블의 크기가 모듈 크기의 50%에 달하여 이를 공유하는 것만으로도 큰 크기감소 효과가 있었다. 나머지 논리회로들의 공유는 매우 복잡하며 크기 감소효과도 크지 않으며 분리 가능한 구조를 위해서도 바람직하지 않았다.

3.4 역양자화/변환기

역양자화/변환기는 MPEG-2/MPEG-4/VC-1에 대해 DC/AC 예측 및 역 양자화와 변환을 수행하고, H.264의 역 양자화와 변환을 수행한다. (그림 3)은 제안된 MFD의 역 양자화/변환기 구조이다. 각 코덱 간의 유사성은 <표 1>과 같다.

<표 1> 코덱 간의 유사성

역 스캔	MPEG-2 스캔-테이블은 MPEG-4 스캔-테이블의 일부로 공유가능
DC/AC 예측	MPEG-4/VC-1의 DC/AC 예측 과정은 상당 부분 유사
역 양자화	파이프라인 레지스터 공유
역 변환	MPEG-2/MPEG-4의 경우 같은 역변환 매트릭스를 사용하므로, 공유 가능함. 또한, 역변환 수행시 사용되는 변환 매트릭스 공유.

외부 메모리와 매크로 블록 구문 복호기에서 전송된 정보가 예측부, 계수부 및 변환부를 순차적으로 거쳐서 각 코덱 별로 처리된 후 예측기로 전달된다. 각 부분별 상세 설명은 다음과 같다.

역 양자화/변환기의 예측부는 DC/AC 예측을 수행하는 모듈이다. <표 1>에서 기술한 바와 같이 MPEG-4/VC-1의 DC/AC 예측과정은 상당히 유사하기 때문에 하나의 모듈로 구현하고, MPEG-2의 경우는 DC 예측만 수행하고, 연산도 다른 코덱과 상이하기 때문에 별도로 구현하였다. 또한, H.264의 경우는 DC 계수에 대한 하다마드(Hadamard) 변환을 수행한다. 이렇게 함으로써, 분리 가능한 구조로 구현하면서 자원 공유를 통하여 크기의 소형화를 추구하였다.

역 양자화/변환기의 계수부는 역 zigzag 스캔 과정과 역 양자화를 수행하는 모듈이다. MPEG-2의 역 스캔의 경우 스캔-테이블이 MPEG-4에 사용되는 스캔-테이블의 일부이기 때문에, MPEG-4와 공유 가능하다. 그러나, VC-1의 경우는 다른 코덱과는 상이한 스캔-테이블을 사용하기 때문에, 별도의 모듈로 구현하여 분리 가능한 구조로 구현하였다. 역양자화 부분은 코덱 간의 일부 연산이 공유되기도 하고, 파이프라인 레지스터를 공유함으로써 제어부의 소형화가 가능하기 때문에, 하나의 모듈로 구현하였다.

역 양자화/변환기의 변환부는 역변환을 수행하는 모듈이다. MPEG-2/MPEG-4의 역변환 매트릭스는 같기 때문에, 같은 ALU를 사용하도록 구현하였다. 그러나, VC-1/H.264의

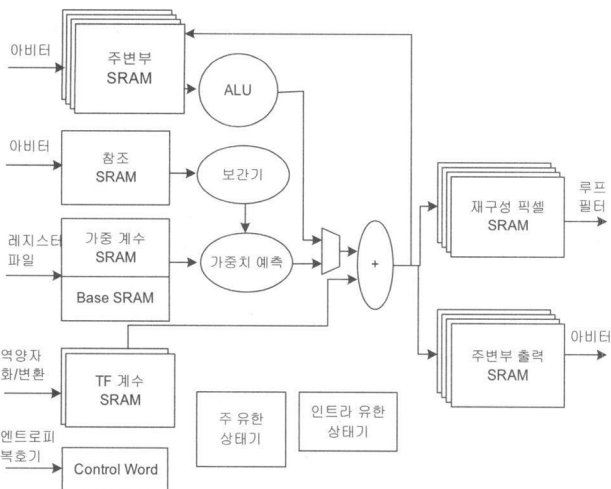
경우는 역변환 매트릭스 가 다르기 때문에, 별도의 ALU를 구현하였다. 다만, 역변환 매트릭스가 상이하더라도, 연산을 수행하기 위한 제어부 및 변환 메모리가 공유될 수 있기 때문에, 이를 공유함으로써 상당한 게이트 수의 감소 효과를 얻을 수 있었다.

역 양자화/변환기의 계수부와 변환부는 이중 버퍼를 이용하여 인터페이스 하도록 설계하였다. 이렇게 함으로써, DC/AC 예측 및 역양자화 과정과 역변환 및 예측 모듈로의 전송 과정을 구분할 수 있어 요구하는 성능을 충족하게 되었다. 이렇게 해서 얻어진 이 모듈의 크기는 전체의 8.4%인 201.6K gates 정도이다.

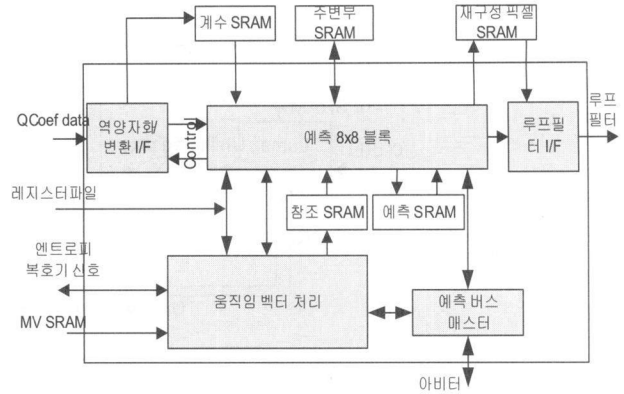
3.5 예측기

예측기는 움직임 벡터를 이용하여 레퍼런스 매크로 블록의 재구성을 수행한다. 인터-매크로 블록에 대해서 동작 재구성을 하는 것은 각 코덱에 공통으로 적용된다. H.264는 인트라-매크로 블록에 대하여 인트라-예측을 하는 기능이 추가되며, VC-1은 인트라-매크로 블록에 대하여 중첩 변환 기능이 추가된다. MPEG-2의 예측기는 MPEG-4의 완전한 부분집합이고, MPEG-4의 예측 기능은 H.264의 그것과 유사하지만 인터-매크로 블록에 대해서 보간 필터가 다르다. 코덱 별로 보간 필터들은 모두 다르고 크기도 크지 않다 (H.264가 8.9K 게이트, VC-1이 39.0K 게이트, MPEG-2/MPEG-4 겸용이 18.0K 게이트). 그래서 예측기는 SRAM 만을 공유하는 분리형 구조로 설계하였다. (그림 4)는 H.264의 예측기의 구조를 보여준다. TF 계수 SRAM과 재구성 픽셀 SRAM은 다른 코덱들과 공유되었으며 주변부 SRAM은 좌, 상, 대각선의 주변 픽셀을 저장하기 위한 것이다.

주변부 SRAM, 참조 SRAM, 가중 계수 SRAM 등으로 입력된 정보는 ALU, 보간기, 가중치 예측기를 거친 후, 역양자화 및 변환을 거쳐 TF 계수 SRAM에 저장된 계수들과 더해져 재구성된 픽셀을 만들어 낸다. 그 결과는 루프 필터로 전달되는데, 내부적 계산에 다시 쓰이는 일부 정보는 주



(그림 4) H.264 예측기 구조

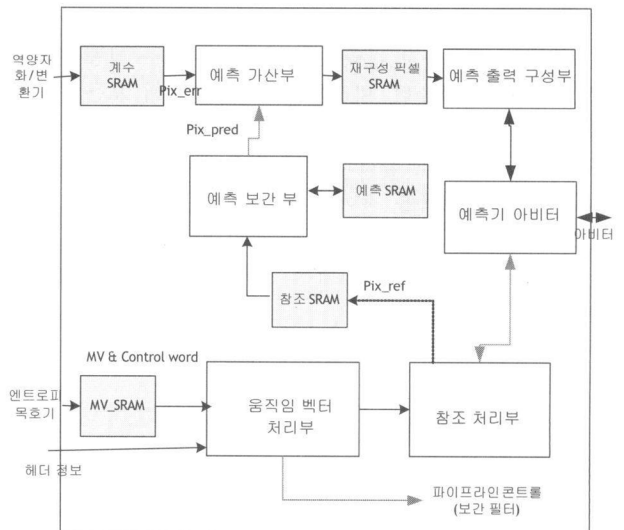


(그림 5) VC-1 예측기 구조

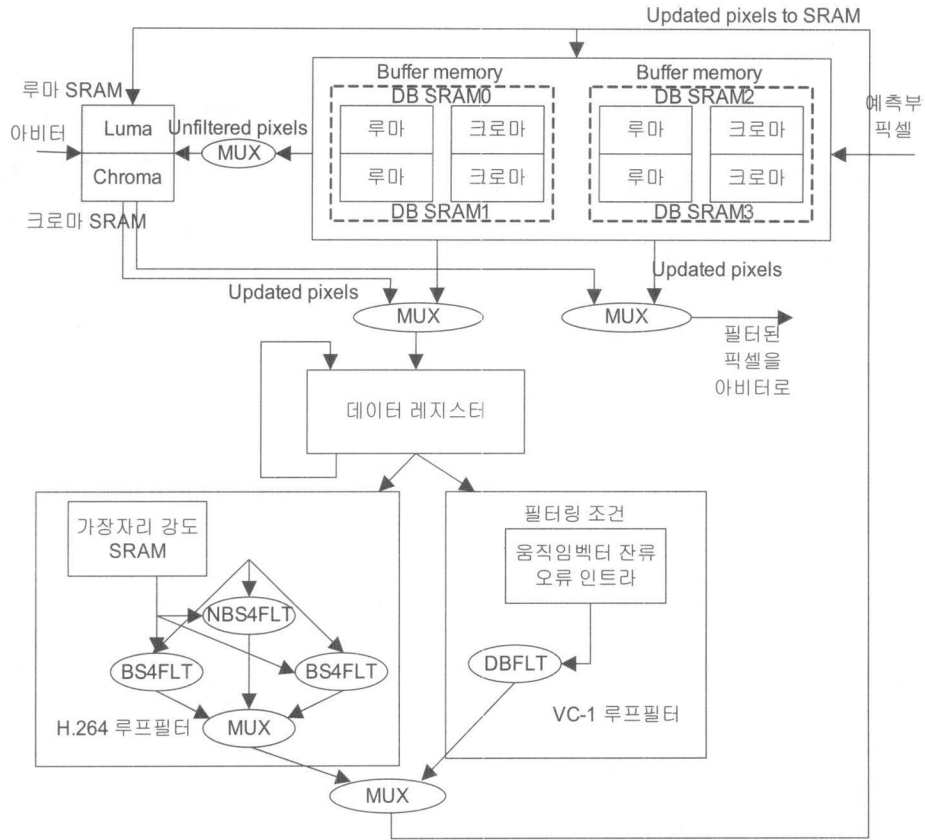
변부 SRAM에 저장되었다가 다시 ALU로 입력된다. 본 H.264의 예측기 설계에 관한 상세한 설명은 [9]에 잘 나타나 있다.

(그림 5)는 VC-1의 예측기의 구조를 나타낸 것이다. 움직임 벡터 처리부는 움직임 벡터를 MV SRAM으로부터 읽어 들인다. 이 움직임 벡터 값에 근거하여 예측 버스 매스터를 통해 참조 픽셀을 읽어 들이도록 지시한다. 모든 인터-블록 참조 픽셀들은 읽혀 와서 참조 SRAM에 저장된다. 이 모듈은 관련된 인트라 블록과 강도 보상 정보를 예측 8x8 블록으로 전달한다. 여기서는 역양자화/변환부에서 들어온 정보를 참조 정보와 결합하여 재구성된 정보를 만들어 낸다. 필요한 주변 정보들이 주변부 SRAM을 통하여 입력된다. 계산된 재구성 픽셀은 루프 필터 I/F를 통하여 루프 필터 부로 전달된다. 주변부 SRAM, 참조 SRAM, 재구성 픽셀 SRAM과 예측 SRAM 등은 모두 다른 코덱들의 예측부와 공유가 가능한 구조로 설계되었다.

(그림 6)은 MPEG-2 및 MPEG-4 겸용 예측기의 구조를 나타낸 것이다. 움직임 벡터 처리부는 엔트로피 복호기의 정보와 헤더 정보를 받아들여 참조 픽처로부터 읽어와 할 참조 블록의 주소와 크기 정보를 가공한 후, 참조 처리부로



(그림 6) MPEG-2/MPEG-4 예측기 구조



(그림 7) 루프 필터 구조

전달한다. 참조 처리부는 이 정보를 이용해 참조 픽셀들을 읽어 들여 참조 SRAM에 저장한다. 이 저장된 정보는 예측 보간부에서 읽어 들여 예측 SRAM에 저장된 정보와 함께 보간 연산을 수행하고, 예측된 픽셀 값은 예측 가산부로 전해져 역양자화/변환기에서 전달되어온 계수 오차 값과 더해진다. 그 결과 최종 결과값을 재구성 픽셀 SRAM에 저장한다. 이 값은 아비터를 통하여 메모리로 보내진다. 참조 SRAM, 재구성 픽셀 SRAM과 예측 SRAM 등은 모두 다른 코덱들의 예측부와 공유가 가능한 구조로 설계되었다.

그림에는 표시하지 않았지만 픽셀DMA라는 기능이 예측기에 추가되어 있는데, 이것은 예측기로부터의 모든 메모리 요청신호를 관리하며 참조 픽처 정보를 전용 메모리에 저장하는 방식으로 참조 블록과 주변 블록 사이의 집약성을 활용하여 버스 대역폭을 줄이는 캐쉬 역할을 한다. 이렇게 함으로써 일반적인 경우 데이터 전송량의 30% 가량을 줄일 수 있다. 예측기 모듈의 크기는 전체의 12.1%인 290.4K 게이트이다.

3.6 루프 필터

디블록킹 필터를 사용하는 코덱은 H.264, VC-1, MPEG-4 이다. 이 중 H.264와 VC-1의 경우는 인루프 필터이고 MPEG-4는 후처리 필터이다. 이는 H.264나 VC-1의 필터를 후처리 필터로써 재활용해서 MPEG-4의 루프 필터

를 대체할 수 있다는 말이 된다. H.264와 VC-1은 인루프 필터라는 점 외에는 필터링 방식이나 순서 등이 완전히 다르기 때문에, 약간의 크기를 줄이기 위해서 일부를 공유하기 보다는 (그림 7)과 같이 분리 가능한 구조를 위해서 내부 공유하지 않았다. H.264는 가장자리 강도에 따라서 내부의 3개의 필터를 적절히 사용하며, VC-1에서는 하나의 필터가 공통으로 사용된다. H.264와 VC-1의 필터 엔진은 필요한 조건에서 하나만 동작하여 불필요한 전력 소모를 방지한다. 공유 면적을 제외한 각각의 크기는 H.264의 경우 33K이고, VC-1의 경우 73.6K이다. 매크로 블록 단위로 필터링이 진행이 되며 픽셀 값들은 SRAM에 저장된다. (그림 7)에서 DB SRAM 0, 1과 DB SRAM 2, 3으로 표기된 이중 버퍼SRAM은 예측부로부터 필터링 될 데이터를 받으면서 동시에 필터링이 완료된 데이터를 저장한다. 이들은 루마에 대한 동작이 완료된 이후에 바로 크로마에 대한 동작을 진행할 수 있도록 루마와 크로마로 분리되어 있다. 이중 버퍼를 사용할 때 단일 버퍼를 사용할 때보다 계산 시간에서 최소 22%의 성능 향상을 가져오는 것이 실험으로 밝혀졌다. 필터링에 필요한 주위의 픽셀 값은 SRAM에 저장되며 다음 매크로 블록의 필터링을 위해서 DB SRAM으로부터 픽셀 값을 넘겨 받고, 일부는 외부 메모리로부터 값을 읽어온다. 루프 필터 모듈의 크기는 전체의 5.5%인 132 K gates이다.

4. 설계 결과

본 논문에서 제안된 MFD는 65nm 공정에서 225MHz로 동작하도록 설계되었다. 64-비트 소닉스 버스를 통하여 32-비트 DDR2 메모리와 연결되며 버스는 200MHz, 외부 메모리는 333MHz로 동작한다. 이때 20%정도의 대역폭은 버스를 공유하는 다른 블록들에게 제공된다.

본 MFD를 디지털 TV SoC에 결합하는 과정에서 메모리 지연 시간을 줄이고 대역폭을 충분히 확보하도록 많은 노력이 필요했다. 성능에 영향을 줄 수 있는 지연 요소는 내부 버스 지연과 외부 버스 지연 및 메모리 접근시간이다. 이 중에서 가장 큰 비중을 차지하는 메모리 접근시간을 줄이기 위해서는 블록 단위의 주소 방식을 사용하였으며, 매크로 블록 단위로 인접한 픽셀들이 메모리의 동일한 행(row) 주소에 저장되도록 하여 성능을 높였다. 또한 내부 버스와 외부 버스에서 발생할 수 있는 지연 시간을 최소화 하고자 요청신호와 데이터 전송을 서로 다른 페이즈로 분리하여 중첩되어 처리 가능하도록 하였다. 외부 버스의 특성을 고려하여 각 모듈이 내부 버스를 통해 한번에 많은 양의 데이터를 요청하더라도 여러 개의 요청신호로 분할 되도록 설계하였다.

실제적 비디오 복호기의 크기들은 구현 예에 따라 다르겠지만 대체적으로 가장 복잡한 프로파일과 레벨들의 고품질 응용 구현의 경우, H.264와 VC-1 등의 구현은 각각 2백만 게이트 가량의 크기로 구현되고, MPEG-2 와 MPEG-4 계열들은 백만 게이트 내외로 구현된다. 제안된 MFD 전체의 크기는 2.4M gates이다. 명령어-캐쉬 8KB, 데이터-캐쉬

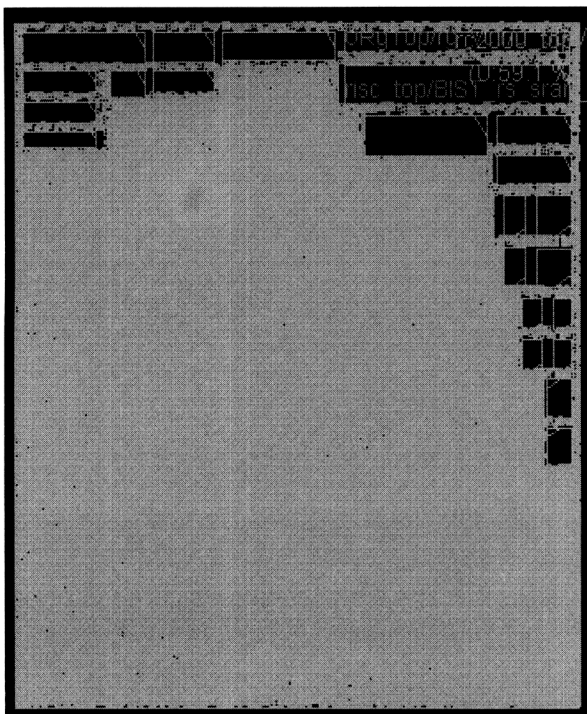
4KB를 갖는 32-비트 임베디드 리스크 프로세서, 온칩 메모리, 호스트I/F, 아비터 등은 전체의 60.1%인 1.44M gates이다. 엔트로피 복호기는 전체의 13.9%인 333.6K 게이트이다. 역 양자화/변환기는 전체의 8.4%인 201.6K 게이트이다. 예측기는 전체의 12.1%인 290.4K gates 이다. 루프 필터는 전체의 5.5%인 132K gates 이다. 이로써 제안된 MFD는 분리형 구조를 활용하며 코덱들을 통합하여 설계하였지만 통합되지 않은 설계에 비해 60% 정도의 크기가 축소됨을 알 수 있다. (그림 8)은 65nm 공정에서 포스트 레이아웃 이후의 MFD의 영상이다. 그 크기는 가로 1,654 μ m, 세로 1,939 μ m이다. 현재 MFD의 설계 및 검증이 완료되었으며 SoC를 위한 과정을 진행 중에 있다.

5. 결론

본 논문에서는 고성능 고품질급 이상을 처리할 수 있으며 MPEG-2/MPEG-4/H.264/VC-1 비디오 표준을 지원하는 MFD의 설계 방법을 제안하였다. 제안된 구조는 기본적으로 리스크 프로세서나 온칩 메모리 등 공유효과가 큰 자원들을 공유하여 60% 이상의 크기를 절약하였다. 그리고 하드웨어로직으로 구성된 그 외의 모듈들은 코덱들의 추가나 제거가 용이하도록 분리 가능한 구조를 사용하여, 설계와 검증의 시간을 단축하고 IP의 안정성을 향상시켰다. 향후 연구는 새로이 등장하고 있는 코덱들을 계속 추가하는 것과 MPEG-4/H.264 부호기를 결합한 MFC의 설계에 관한 연구가 될 것이다.

참 고 문 헌

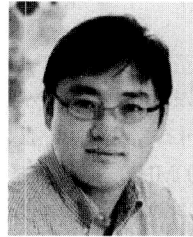
- [1] Chih-Da Chien, et al., "A 252kgates/71mW Multi-Standard Multi-Channel Video Decoder for High Definition Video Applications," ISSCC Dig. Tech. Papers, pp.282-283, 2007.
- [2] M. Hase, et al., "Development of Low-power and Real-time VC-1/H.264/MPEG-4 Video Processing Hardware," Design Automation Conference, pp.637-643, 2007.
- [3] T. M. Liu, et al., "A 125 μ W, Fully Scalable MPEG-2 and H.264/AVC Video Decoder for Mobile Applications," JSSC, pp.161-169, 2007.
- [4] T. W. Chen, et al., "Architecture Design of H.264/AVC Decoder with Hybrid Task Pipelining for High Definition Videos," IEEE Proc. ISCAS, Vol. 3, pp.2931-2934, 2005.
- [5] C. C. Lin, et al., "A 160k Gate 4.5kB SRAM H.264 Video Decoder for HDTV Applications," ISSCC Dig. Tech.



(그림 8) MFD 배치 및 배선 후 영상

Papers, pp.406-407, 2006.

- [6] Y. Hu, et al., "A High Definition H.264/AVC Hardware Video Decoder Core for Multimedia SoC's," Proc. ISCE, pp.385-389, 2004.
- [7] H. Y. Kang, et al., "MPEG4 AVC/H.264 Decoder with Scalable Bus Architecture and Dual Memory Controller," Proc. ISCAS, Vol. 2, pp.145-148, 2004.
- [8] J. Bae, N. Park and S. Lee, "Register Array Structure for Effective Edge Filtering Operation of Deblocking Filter," LNCS, Vol. 4096, pp.805-813, 2006.
- [9] J. Bae, N. Park and S. Lee, "Quarter-pel Interpolation Architecture in H.264/AVC Decoder," IPC 2007, pp.224-227, 2007.
- [10] Draft ITU-T Recommendation and Final Draft International Standard of Joint Video Specification, ITU-T Rec. H.264 | ISO/IEC 14496-10 AVC, 2003.
- [11] SMPTE STANDARD VC-1 Compressed Video Bitstream Format and Decoding Process, Approved 2006.
- [12] International Standard, Information technology - Coding of audio-visual objects - Part2: Visual, ISO/IEC 14496-2 Third Edition, 2004.



배종우

e-mail : jwbae@mju.ac.kr

1988년 서울대학교 제어계측공학과(학사)

1989년 미국 Univ. of Southern California
컴퓨터 공학과(공학석사)

1996년 미국 Univ. of Southern California
컴퓨터 공학과(공학박사)

1996년~1999년 미국 산호세 Actel 주식회사 스텝 아키텍처 엔지니어

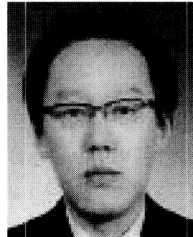
1999년~2002년 미국 산호세 Avanti 주식회사 선임 소프트웨어 엔지니어

2002년~2002년 미국 산호세 Pulsent 사 아키텍처 엔지니어

2003년~2007년 삼성전자 시스템 LSI D-TV 개발팀 수석연구원

2007년~현 재 명지대학교 정보공학과 조교수

관심분야: VLSI/SoC 설계, 영상처리 및 압축, 디지털 TV, 프로세서 등



조진수

e-mail : jscho@kyungwon.ac.kr

1994년 인하대학교 전자공학과(학사)

1998년 미국 Columbia 대학교 전기공학과
(공학석사)

2003년 미국 Georgia Institute of
Technology 전기/컴퓨터공학과
(공학박사)

2004년~2006년 삼성전자 시스템 LSI D-TV 개발팀 책임연구원

2006년~현 재 경원대학교 소프트웨어공학부 전임강사

관심분야: 영상처리, 성능개선 및 압축, 의료용 영상처리 등