



H.264/AVC Baseline

디코더 설계 기술

>> Design Methodology



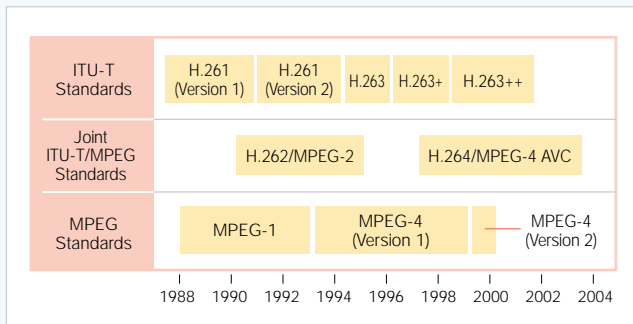
H.264/AVC Baseline 디코더 설계 기술

김승철 연구원_기반기술연구소 SOC설계연구부 멀티미디어SOC설계팀(skimc@etri.re.kr)

조한진 책임연구원_기반기술연구소 SOC설계연구부 멀티미디어SOC설계팀(hjcho@etri.re.kr)

1.Introduction

1.1. 동영상 압축 기술의 변화



<그림 1> 동영상 압축 기술 변화

디지털 신호 처리 기술, 각종 저장 매체의 출현, 네트워크 기술의 진보와 함께 기존의 음성 정보에 국한되었던 미디어 서비스는 영상과 데이터가 포함되는 멀티미디어 서비스로 진화해왔다. 그 중에서도 특히 방대한 양의

동영상 데이터를 저장하고 전송하기 위한 여러 가지 압축 기술이 연구되었으며, 특히 1980년대 후반 디지털 동영상 정보의 부호화 및 저장기술 표준규격을 제정해야 한다는 요구가 제기되면서 기술발전이 가속화하기 시작했다.

H.263+와 MPEG-4 표준이 개발된 뒤 무선통신이 급격히 확산되면서 종전 압축방법에 비해 더욱 향상된 압축효율을 제공하고, 다양한 통신환경을 수용할 수 있는 동영상 압축기술 규격의 필요성이 대두했다. 이에 ITU는 H.26L로 명명한 차세대 부호화 방식의 '기술제안요청서'를 발표했으며, 각급 기업체·연구소·학계의 활발한 연구가 진행됐다. 이후 ISO/IEC의 MPEG 그룹이 H.26L 프로젝트에 참가하였고, 마침내 ITU-T와 ISO/IEC에서 각각 H.264, MPEG-4 Part 10의 이름으로 최종 규격이 승인되었다.[1]

1.2. Introduction to H.264

ITU-T와 ISO가 공동 제정한 차세대 동영상 압축 표준 H.264는 다양한 네트워크 환경에 쉽게 부응할 수 있는 유연성과 동영상의 부호화 효율성 측면에서 MPEG-2, MPEG-4(Part 2) 등 기존 기술표준들에 비해 많은 진

보가 있었다. 비록 H.264가 기존의 표준들에서와 같이 하이브리드 부호화라는 유사한 방식을 이용하고 있으나, 향상된 엔트로피 부호화, 가변 블록 크기의 블록 변환, 향상된 움직임 예측 및 보상, 인루프 디블로킹 필터(In-Loop Deblocking Filter) 등의 기술적 우위를 지닌다.[1]

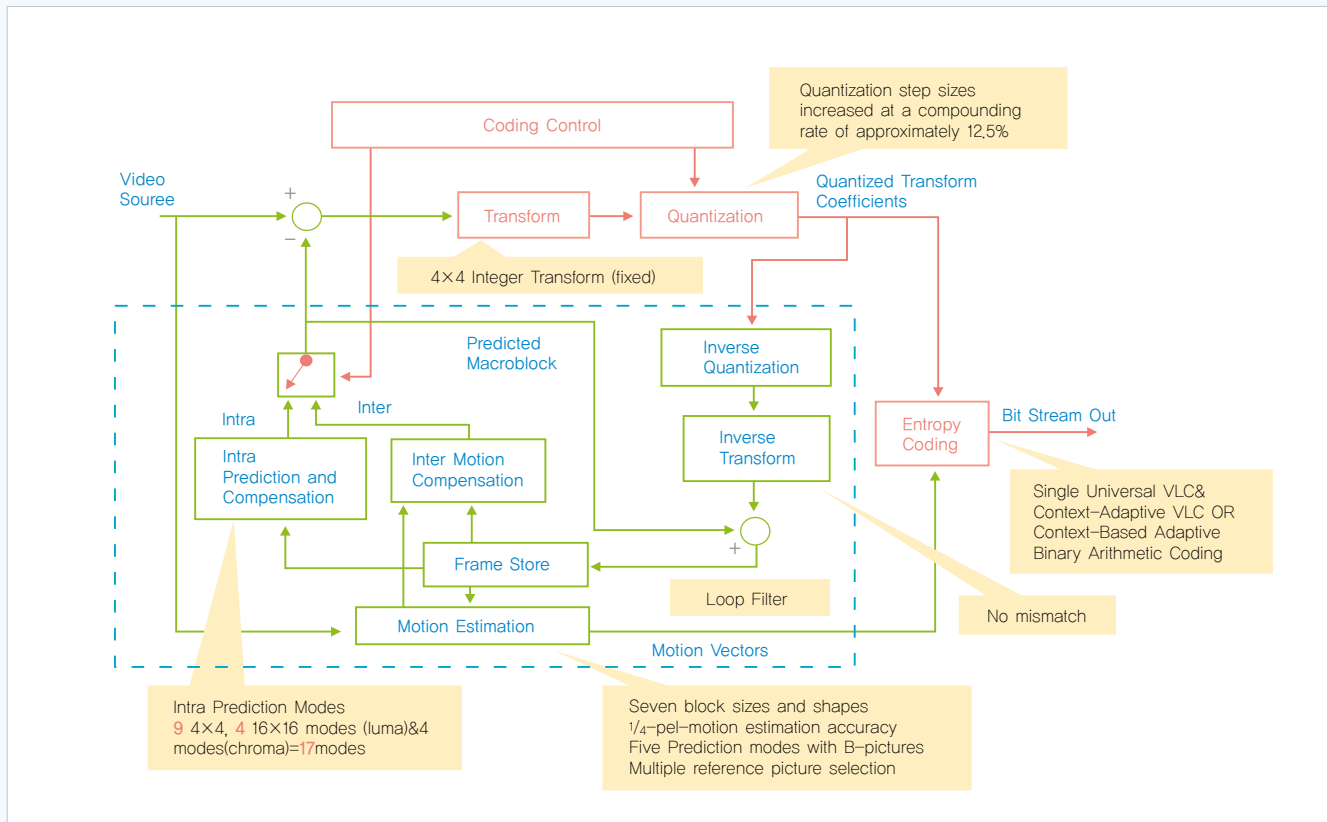
H.264가 MPEG-2, MPEG-4(Part 2) 등 기존 동영상 압축 표준에 비해 높은 압축성과 유연성의 장점을 가진 반면, 부호기 및 복호기의 복잡도 역시 훨씬 증가한다는 단점이 있다. 부호기 측면에서는 기존 표준보다 더욱 많아진 파라미터 및 부호화 모드를 결정해야 하며, 복호기 측면에서 디블로킹 필터나 1/4화소 단위의 움직임 보상 등으로 인해 계산량이 매우 증가했다. 따라서 H.264가 상용화되기 위해서는 기술원리를 정확히 이해하고, 각 알고리즘 요소들을 효율적으로 구현하는 적용기술이 매우 중요하다.[1]

그림2에 H.264 압축 알고리즘의 개요를 보였다.

2. Design Issues

H.264 압축 방식은 압축률과 화질에서의 이득을 위해 보다 복잡한 알고리즘 요소들을 도입한 규격이다. 특히, 예측의 정확도를 높여 압축률과 화질의 개선을 함께 얻기 위해 화면간 움직임 예측과 화면내 예측 과정 등에 도입된 여러 가지 알고리즘 요소들에서 그 복잡도의 변화를 실감할 수 있다. 기존의 알고리즘들과 달리 1개 이상의 참조 화면들로부터 참조 데이터를 가져올 수 있도록 한 규정을 준수하기 위해 N (N >= 1)개의 프레임 영상 데이터를 저장할 수 있는 프레임 메모리 공간을 확보해야 하고 이는 바로 설계 비용을 증가시키는 주요 원인이기도 하다.

가변 크기(최소 4x4, 최대 16x16) 블록 단위의 예측이 가능하도록 한 규정으로 인하여 하나의 매크로블록(MB, Macroblock) 당 최대 16개의 움직임 벡터가 존재하게 되고, MB 예측을 위한 프레임 메모리 액세스



〈그림 2〉 H.264 압축 코딩 알고리즘 개요[2]



회수가 최대 48회까지 늘어나게 되어 디코더 시스템 구조 설계에 커다란 부담을 주게 된다. 가변 크기 블록 단위의 움직임 예측과 다수 참조 화면의 도입은 복호 과정에서 프레임 메모리에 대한 액세스 회수와 총 액세스 데이터의 양을 증가하게 만들었고, 1/4 픽셀 단위의 움직임 예측 방식의 도입으로 인하여 보간 필터 처리에 소요되는 계산량이 현저하게 증가하였다. Deblocking 필터를 반드시 포함하도록 한 요구 사항은 H.264 복호 시스템을 설계하는 데 있어 큰 부담이 될 수 밖에 없다.

SOC 설계의 관점에서 볼 때 전송 트랜잭션 수와 데이터 양의 증가는 버스 상에서의 트래픽 제어의 어려움을 유발하며, 그와 함께 목표 성능을 만족하기 위한 최적의 HW/SW 분할 및 각각의 설계 사양 제시에 곤란을 가져오게 된다.

특히, 가변 크기 블록 변환 및 예측/보상 알고리즘의 구현에 있어, 데이터 전송을 위한 트랜잭션의 수 자체가 가변적일 뿐 아니라, 각 트랜잭션마다의 전송 데이터 양 또한 가변적이므로 버스 트래픽 제어 및 전체 시스템의 동작 제어 스케줄링에 어려움을 더하게 된다.

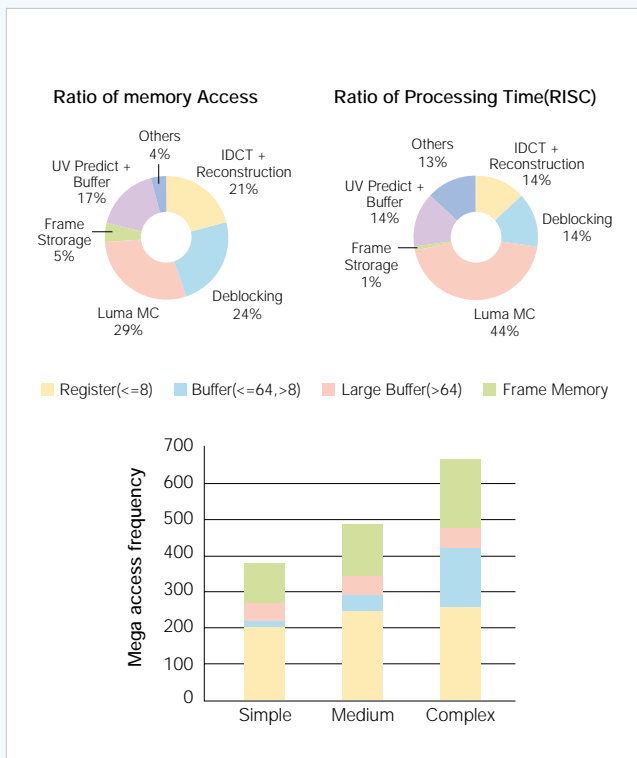
H.264가 기존의 동영상 압축 규격들과 차별화될 수 있는 또 하나의 특징이 바로 화면내(인트라) 예측 방식이다. 인트라 예측 방식은, 16x16, 4x4 크기 등 2가지 타입의 블록 단위 예측이 가능하고, 특히 4x4 블록 단위의 경우 9가지의 예측 모드를 제공하여 그 정확도를 향상시키도록 규정하고 있다.

그러나 예측에 사용되는 인접 MB 데이터를 디블로킹 필터 연산을 거치기 이전의 복원 데이터로 규정함으로써 디블로킹 이후의 결과를 참조하는 움직임 보상 처리 및 디블로킹 필터 처리 등과 데이터 흐름면에서 상충되는 면이 있어 디코더 구조 설계상의 어려움이 뒤따른다. 이러한 기술적 어려움들을 극복하면서 보다 빠르고 효율적으로 구현하기 위한 방법으로 임베디드 SOC 설계 방식이 적합하다. SOC 타입의 설계 방법은 데이터 전송과 알고리즘 고유의 특성에 적합한 플랫폼을 기반으로 하여 각 IP들의 설계 사양을 결정하고 성능을 예측하며, 시스템 동작 제어 스케줄링과 상위 어플리케이션과의 정합 문제 등에서 유연성을 확보할 수 있는 장점이 있다.

H.264 디코더 SOC를 설계하는데 있어서, 가장 먼저 해야 할 일은 그림 3과 같은 분석 결과를 토대로 하여 설계 비용을 추산하고 HW/SW 분할 요소 추정, 메모리 파라미터 추출, 버스 구조 등 아키텍처 프로토타입을 구성하는 것이다. 이를 기반으로 하여 성능을 예측하고, 각 요소들의 설계 비용과 성능을 비교하면서 최적의 구조를 갖는 최종 아키텍처를 찾으려 한다.

그림 3에서 보인 바에 따르면 디코더 시스템 자원의 절반 이상이 움직임 보상 처리에 소요되고, 메모리 액세스의 3/4을 움직임 보상과 디블로킹 필터 알고리즘 수행에 할애되고 있다. 또한 SOC 설계시에 고려해야 할 주요 파라미터인 데이터 전송에 있어서도 프레임 메모리와 레지스터 데이터 액세스가 상당 부분을 차지하고 있어 버스 시스템의 데이터 전송 역량과 효율이 매우 중요함을 알 수 있다.

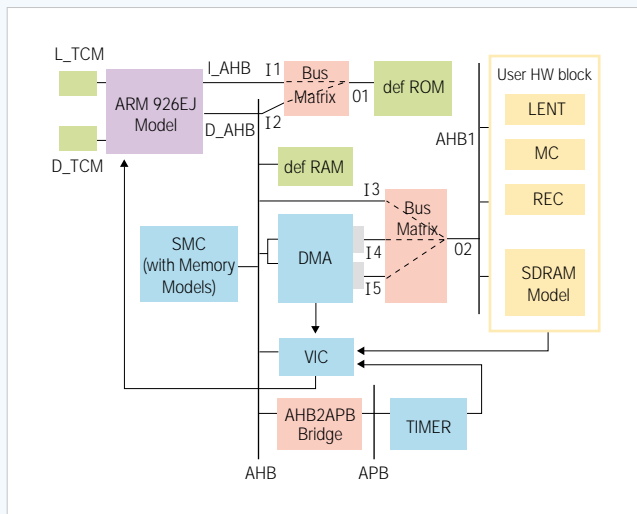
각 알고리즘 요소들을 분석하여 볼 때, 움직임 보상과 디블로킹 필터 처리를 제외한 다른 요소들에 대하여는, 버스를 점유하게 되는 메모리 액세스와 회로 설계 면적 증가 사이의 trade-off가 완전히 혹은 부분적으로 가능하다. 움직임 보상이란 움직임 벡터가 가리키는 참조 데이터에 대해 보간 필터 알고리즘을 적용하여 예측 데이터를 구하고, 이를 역변환/역양자화를 거친 차영상 데이터와 더함으로써 영상을 복원하는 과정을 말하고, 디블로킹 필터 알고리즘은 인접한 매크로블록(MB)과의 경계 혹은 현재 MB 내부에서의 각 블록들간의 경계 부분에서 보이는 시각적인 불연속성을 제거하여 최종 복원 화면의 화질을 개선하기 위한 것이다.



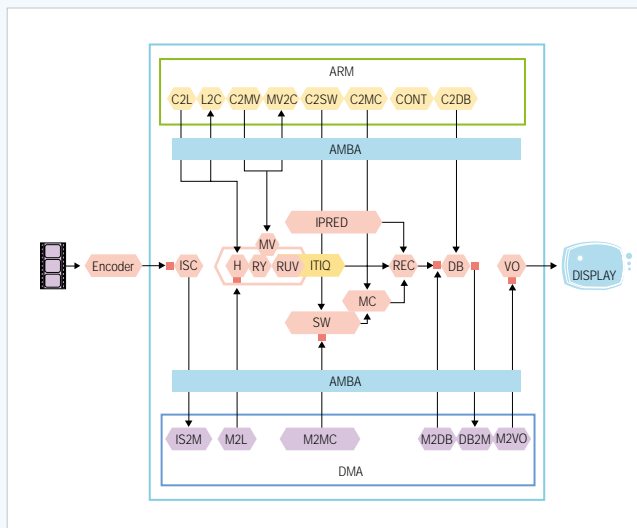
(그림 3) H.264 디코더 프로파일링 결과[3]

3. H.264 Baseline Decoder design

H.264 디코딩 알고리즘 분석을 토대로 하여 그림 4와 같은 형태의 기본 플랫폼을 구성할 수 있다. 플랫폼을 구현함에 있어서, AHB 버스 기반 위에 ARM프로세서와 DMA를 마스터 IP로 사용하였고, 디코딩 알고리즘과 관련된 기능 요소들을 슬레이브 IP로 하여 적용할 수 있도록 하였다. 그림 3에



〈그림 4〉 기본 플랫폼



〈그림 5〉 HW/SW 분할 및 연결 관계

〈표 1〉 H.264 Baseline 디코더 Hardware IPs

LENT	엔트로피 디코딩	H	헤더 디코딩
		RY	Y 레지듀얼 디코딩
		RUV	UV 레지듀얼 디코딩
MV	움직임 벡터 계산 움직임 보상을 위한 프레임 메모리 액세스 어드레스 계산 보간 필터 동작 정보 계산		
ITIQ	정수 역변환 및 역양자화		
IPRED	인트라 예측		
SW	움직임 보상을 위한 참조 데이터 저장 및 가로/세로 패딩		
MC	움직임 보상을 위한 보간 필터		
REC	예측 (IPRED, MC) 결과와 차영상 데이터 (ITIQ)를 더하여 원영상 복원		
DB	디블로킹 필터		

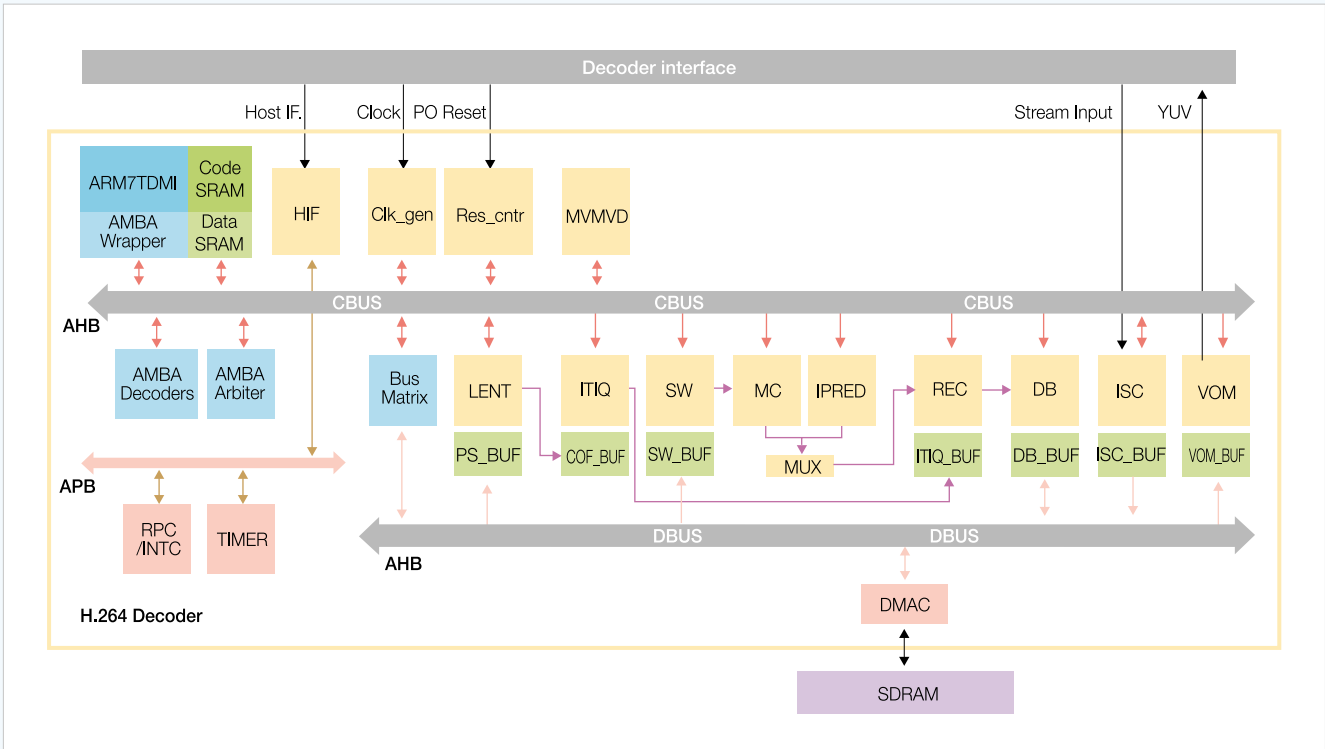
〈표 2〉 H.264 Baseline 디코더 Software IPs

C2L	모듈제어
C2MV	
C2AW	
C2MC	
C2DB	
L2C	모듈 동작 결과 리턴 및 일부 가공
MV2C	
CONT	MB/Frame 단위 파라미터 가공/저장 및 시스템 제어
M2MC	프레임 메모리와 모듈의 버퍼 간의 데이터 전송(DMA) 제어
M2DB	
DB2M	

서 보인 바와 같이 프레임 메모리 데이터 전송량과 각 요소들에 대한 레지스터 데이터 전송량 모두 큰 비중을 차지하고 있으므로 DMA를 이용하는 2개의 분리된 버스 시스템 구조로부터 출발하는 것이 유리하다.

단일 버스에 각 IP들을 연결하는 경우, 프레임 메모리 데이터 전송과 레지스터 데이터 전송이 빈번하게 발생하는 디코딩 알고리즘의 특성으로 인하여 버스 트래픽 제어 효율이 감소할 뿐만 아니라, 각 IP 모듈들의 동작과 데이터 이동을 효율적으로 제어하기 위한 스케줄링이 불가능하게 된다.

디코딩 알고리즘 분석에 따라 그림 5에 포함된 HW/SW 기능 요소들로 분할할 수 있다. SW 기능 요소는 다시 2가지로 세분하여, DMA가 버스 마



〈그림 6〉 Multiple BUS Architecture

스터로서 독립된 데이터 전송을 수행할 수 있도록 제어하는 프레임 메모리 액세스 제어 요소와 그 밖의 모든 프로세서 수행 요소들로 나뉜다.

3.1. Architecture

(그림 6)에 H.264 Baseline 디코더 구조를 보였다. 두 개의 AHB 버스에 2개의 마스터 모듈(ARM7TDMI, DMAC)를 갖는 버스 분리형 플랫폼을 기반으로 하며, ARM 프로세서와 각 모듈들간의 통신이 이루어질 질 제어 버스(CBUS)와 DMAC와 모듈들간의 통신이 이루어질 데이터 버스(DBUS)로 나뉘어진다.

상호간 독립적인 CBUS와 DBUS는 ‘버스 매트릭스’를 매개로 하여 연결되어 프로세서가 DMA의 동작을 제어할 수 있도록 한다. 각 하드웨어 모듈들은 필요에 따라 CBUS와 DBUS 모두 연결될 수 있도록 2개의 AHB 인터페이스 포트를 갖거나 프레임 메모리와 데이터를 주고 받을 필요가 없는 모듈의 경우 1개의 AHB 포트를 갖는다.

버스 분리형 플랫폼은 수 백에서 수 천 클럭 사이클이 소요되는 데이터

전송 시간 동안 프로세서로 하여금 필요한 모듈들의 동작을 준비하거나 결과를 넘겨 받을 수 있도록 함으로써, 단일 버스 기반의 플랫폼 구조에서 발생하는 디코딩 스케줄링 상의 잉여 사이클들을 제거할 수 있을 뿐 아니라, 스케줄링 작업 자체가 매우 용이하여진다.

그러나, 이러한 버스 분리형 플랫폼은 단순히 버스를 분리하는 것만으로는 그 장점을 최대한 활용할 수 없다. 앞서 언급한 바와 같이 움직임 보상을 위한 프레임 메모리 데이터 액세스의 형태가 그 길이와 회수에 있어서 가변적이므로 각 전송 트랜잭션들과 프로세서가 수행해야 할 또 다른 프로세스들을 적절히 혼합하여 스케줄링하는 것이 매우 어렵기 때문이다. 또한 DMA 제어와 데이터 전송이 반복되는 형태의 구현은 그 자체가 상당한 양의 잉여 사이클을 포함할 수 밖에 없다.

따라서, DBUS의 마스터 역할을 수행할 DMA가 프로세서의 개입 없이 1개 이상의 전송 트랜잭션을 지속적으로 수행할 수 있을 때에야 비로소 최상의 효율을 갖는 시스템이 될 수 있다.

3.2. Low Power Consideration

H.264 디코딩 과정에 대한 각 알고리즘 요소들의 활용 빈도수를 측정 한 통계 결과에 따르면, 각 요소들마다 그 수치에서 현저한 차이를 보임을 알 수 있다. 비록 영상의 특성에 따라 그 결과에 차이가 있을 수는 있지만, 통신에서와 같이 대역이 제한되는 응용 분야에서 영상의 차이에 따른 편차는 크지 않게 된다. 이러한 알고리즘 상의 통계적 특성을 시스템 구조와 동작 제어 스케줄링, 각 모듈들의 설계 사양에 적용할 경우 성능의 개선은 물론 저전력 특성의 향상을 꾀할 수 있다.

구조 및 스케줄링의 관점에서 고려되어야 할 특성에는 'Skipped MB', 'No Residual', '정수 움직임 벡터', 'Zero-strength Loop filter' 등이 있다. 위의 특성들을 적용하면 각 MB의 인코딩 특성에 따라 해당 MB를 디코딩하는 과정에서 엔트로피 디코딩, 정수역변환 및 역양자화, 보간 필터, 디블로킹 필터 연산 등을 생략하거나, 전송 및 처리해야 할 데이터의 양을 현저히 줄일 수 있다.

3.3. 2-stage Pipelined Scheduling

그림 4의 구조를 바탕으로 하여 H.264 디코딩 알고리즘에 대한 상위 수준에서의 성능을 분석한 결과, 그림 6의 시스템 구조와 그림 5의 디코딩 프로세스를 수행하기 위한 2단 파이프라인 제어 체계를 도출하였다.

파이프라인 단계가 깊을수록 시스템의 병렬 특성은 커지나, 반대로 이를 제어하기 위한 복잡도, 특히 프로세서가 처리해야 할 데이터의 양과 종류가 늘어나고 하드웨어 모듈들의 인터페이스 설계 비용이 증가함에 따라 병렬화의 효과가 감소된다.

전체 모듈은 '엔트로피 디코딩, 예측 및 보상'의 1 단계와 '디블로킹 필터'의 2단 파이프라인의 체계로 운용된다. '디블로킹 필터' 단계의 수행을 위해 필터 연산 정보(필터링 정도를 나타내는 지수 파라미터)들은 1 단계에서의 '디코딩, 예측' 과정에서 계산되도록 함으로써, 디블로킹 모듈(DB)의 인터페이스와 이를 제어하기 위한 소프트웨어의 복잡도를 함께 줄일 수 있다.

각각의 시스템 동작 주파수를 나타내었다. 성능 검증에 사용된 비트스트림들은 JM8.6 reference C로부터 대역 제한 조건 없이 고화질로 생성하였고, HW/SW 혼합 시뮬레이션과 FPGA 에뮬레이션 등의 방식으로 검증되었다.

〈표 3〉 H.264 Baseline 디코더 설계 결과 요약

	QCIF 30 fps	QVGA 30 fps	CIF 30 fps	VGA 30 fps
Foreman	6 MHz	15 MHz	18 MHz	54 MHz
Mobile	9 MHz	20 MHz	24 MHz	60 MHz
Coastguard	8 MHz	17 MHz	20 MHz	57 MHz

5. References

- [1] 이승준(주엠큐브웍스), "[테마특강]모바일 멀티미디어 기술의 진화", '전자신문 2004.3.16'
- [2] "Emerging H.264 Standard: Overview and TMS320DM642-Based Solutions for Real-Time Video Applications", 'UB Video Inc.'
- [3] 이용기, 한기훈, 이영렬, "JVT(Joint Video Team) Codec의 복잡도 분석", 2002 컴퓨터/반도체 소사이터티 추계학술대회, 제 25권 제 2호, pp. 299-302, 2002년 11월

4. Performance

설계된 H.264 디코더의 동작 성능은 아래 표와 같다. Foreman, Mobile, Coastguard 등의 테스트용 표준 권고 시퀀스들을 대상으로 하여, 4종류의 화면 해상도 조건에서 30 frame/sec를 실현하기 위한